Received: 22 March 2021 | Revised: 29 May 2021 | Accepted: 21 June 2021

IET Energy Systems Integration

DOI: 10.1049/esi2.12030

ORIGINAL RESEARCH PAPER

Deep multi-agent Reinforcement Learning for cost-efficient distributed load frequency control

Sergio Rozada[1] | Dimitra Apostolopoulou[2] | Eduardo Alonso[1]

[1]Artificial Intelligence Research Centre (CitAI), University of London, London, UK

[2]Department of Electrical and Electronic Engineering, University of London, London, UK

Correspondence

Sergio Rozada, Artificial Intelligence Research Centre (CitAI), City, University of London, Northampton Square, London, UK.
Email: sergiorozada12@gmail.com

Abstract

The rise of microgrid-based architectures is modifying significantly the energy control landscape in distribution systems, making distributed control mechanisms necessary to ensure reliable power system operations. In this article, the use of Reinforcement Learning techniques is proposed to implement load frequency control (LFC) without requiring a central authority. To this end, a detailed model of power system dynamic behaviour is formulated by representing individual generator dynamics, generator rate and network constraints, renewable-based generation, and realistic load realisations. The LFC problem is recast as a Markov Decision Process, and the Multi-Agent Deep Deterministic Policy Gradient algorithm is used to approximate the optimal solution of all LFC layers, that is, primary, secondary and tertiary. The proposed LFC framework operates through centralised learning and distributed implementation. In particular, there is no information interchange between generating units during operation. Thus, no communication infrastructure is necessary and information privacy between them is respected. The proposed framework is validated through numerical results and it is shown that it can be used to implement LFC in a distributed and cost-efficient manner.

1 | INTRODUCTION

Electrical systems are undergoing major changes. There is a large number of deployed distributed generation systems that is slowly substituting large electromechanical generators [1]. In the past, the majority of the load was met by large generation units, such as coal or nuclear plants. Nowadays, every single house can be a prosumer, that is, it can produce and consume energy and deliver excess energy to the network. This is facilitated by new market designs, for example, peer-to-peer markets.

This paradigm shift is shaping our understanding of energy and bringing us a whole new range of opportunities as well as challenges. In this context of decentralisation, coordination amongst generators to balance generation and load [2] is more taxing. Traditionally, a hierarchical control system is used to meet this objective, that is, primary, secondary and tertiary frequency control. Primary control keeps the frequency between some acceptable limits, secondary control restores the frequency to the nominal value, and tertiary control does so in a cost-efficient way. Secondary and tertiary control layers need a central authority to send appropriate control signals to generators to shift their generation to meet the load. However, in this new paradigm where there are numerous generators participating in frequency control, the centralised approach shows important limitations in terms of computation and privacy concerns. In this regard, new distributed schemes are necessary to deal with the aforementioned challenges [3].

Different approaches have attempted to tackle this problem by implementing the traditional hierarchical control in a distributed manner (see e.g. [3–5]). In [6], the authors propose a methodology for primary control to mimic droop control strategies which are by nature decentralised algorithms that act upon each generator. The proposed methodology explicitly represents the modified system dynamics of having electronic inverters instead of large turbines. Moreover, efforts have been made to implement a decentralised secondary control scheme, for example, the centralised averaging proportional integral (PI) control presented in [7] and the distributed averaging PI control in [8]. These algorithms use weighted averages of the frequency as the integral feedback. Despite their theoretical appeal, such

approaches suffer from lack of robustness, and their communication demands make them difficult to implement in real-life scenarios [9]. Recently, several nature-inspired optimisation techniques have been proposed to solve the primary and secondary layers of the load frequency control (LFC) problem. Some of the most relevant ones are the water cycle algorithm (see e.g. [10, 11]), the yellow saddle goatfish (see e.g. [12]) and the butterfly optimisation (see e.g. [13]). However, none of these methods take the economic cost into consideration. Regarding tertiary control, it is usually common to solve a primal-dual algorithm that converges to the solution of the dual problem (see e.g. [4], [14–17]), where the communication between nodes enables joint global actions. Nevertheless, as with other approaches, communication is intense between nodes and the system may become very complex. Multi-Agent Reinforcement Learning (MARL) is a promising alternative to implement LFC in a decentralised way (see e.g. [18, 19]). The main drawback of these methods is their computational complexity, which grows exponentially with the number of agents. However, the rise of Deep Learning has opened the door to new techniques and algorithms that address these scalability issues in the LFC problem (see e.g. [20, 21]).

In MARL, various software agents learn optimal policies by negotiating, cooperating and/or competing [22]. MARL has already been used in several power system applications, namely autonomous voltage control (see e.g. [23]), home energy management frameworks (see e.g. [24]) and power system resilience (see e.g. [25]). A review on Reinforcement Learning (RL) for decision-making and control in power systems is given in [26]. In terms of LFC, an initial introduction of RL techniques for such purposes was performed in [27]. Next, more recent further work has been performed in this field. The authors in [28] propose an LFC framework with stability guarantees. The focus of this work is on primary frequency control, and through simulations it is shown that the proposed algorithm outperforms the optimal linear droop control. In [29], a MARL framework is proposed that develops controllers that only use local area state information to cooperatively minimise the frequency deviations and unscheduled tie-line power flows for all the areas. In [30, 31], MARL techniques for multi-area power systems' frequency control are developed; however, in these methods, individual generators are not specifically modelled. In [32], the authors include the economics' element by proposing a multi-objective secondary control framework that takes into account the frequency deviation as well as the frequency mileage payments.

The proposed framework advances research in the field of MARL in LFC by using a realistic representation of the components that comprise the overall system and by achieving both restoration of frequency and cost minimisation. In this regard, we formulate the Balancing Authority (BA) area dynamic behaviour, the individual generator dynamics and its generation rate constraints, a simplified network representation, and wind-based generation output. Next, we recast the LFC problem as a Markov Decision Process (MDP), as is standard in RL problems. We define the states, which are the frequency deviation, the control action of each generator, and their action space. We model the dynamic behaviour of the generators and the network to determine the probability state transition function of the MDP. We design the reward function of the agents so that frequency deviation and total cost are minimised. The design of the reward function is critical, since it determines the behaviour that each agent will learn. In order to determine the reward function, we make use of the frequency deviation as well as the optimality conditions of the economic dispatch problem to incorporate the cost component in the proposed framework. We use this setup to estimate the action-value function of each state-action pair with the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm. MADDPG is an actor-critic algorithm; this means that the architecture of each agent is split into two in which first, the actor directly estimates an action, and then, the critic assesses the suitability of such action by estimating the action-value function of the state-action pair. In MADDPG, the critics use central information to teach each actor the dynamics of the environment as well as the behaviour of the rest of the agents. In operation, actors only use local information since they have learnt how other actors behave during the training phase. Each actor and critic is modelled with a Long Short-Term Memory (LSTM) Network so that previous history is stored and acted upon. The proposed LFC framework operates through centralised learning and distributed implementation. In particular, there is no information interchange between generating units during operation. Thus, no communication infrastructure is necessary between agents and information privacy between them is respected. We validate the proposed framework through realistic case studies and demonstrate that the proposed framework can implement the LFC in a distributed and cost-efficient manner.

To summarise, the contributions of the article are as follows: (i) reformulation of the LFC problem as an MDP; (ii) use of a detailed model taking into account the network, renewable-based generation, generator dynamics and generator rate constraints; (iii) design of the reward function of the agents so that frequency deviation and total cost are minimised; (iv) development of the proposed framework to manage the optimal LFC in a fully distributed manner with the use of local information only; and (v) validation of its robustness against uncertainty introduced from renewable-based generation. This problem was initially introduced in [20] and is extended in this article to implement tertiary control or economic dispatch, that is, the generation units modify their output to meet the change in load in a cost-efficient way to include a detailed power system model by explicitly incorporating the network, wind generation, and a more realistic design of the synchronous generator with its dynamics and generator rate constraints.

The remainder of the article is organised as follows: In Section 2, we describe the power system model that we adopt to develop our analysis framework. In Section 3, we formalise the frequency control problem as an MARL problem. In Section 4, MADDPG is used to implement primary, secondary and tertiary control in a multi-agent problem. In Section 5, we present numerical studies to demonstrate that the proposed methodology is a valid alternative to solve LFC in a distributed and cost-efficient manner. In Section 6, we summarise the results and make some concluding remarks.

2 | PRELIMINARIES

In this section, we present the secondary and tertiary control models that we utilise to develop our framework. More specifically, we introduce dynamic models for synchronous generators, the automatic generation control (AGC) system, the network, and the economic dispatch.

The frequency of the system indicates whether supply and demand are properly balanced. When the generated power exceeds the load, the system frequency increases. Similarly, the frequency decreases if generation is not sufficient to meet the load. Thus, controlling the frequency is a standard approach to balancing demand and supply [33]. Frequency control is structured as a hierarchy of three layers: primary, secondary and tertiary control. In primary control, generation and demand are rapidly balanced since the synchronous generators are either speeding up or slowing down due to the load generation imbalance. This is achieved by a decentralised proportional control mechanism called droop control [34]. Then, a secondary control layer implements an integral control that compensates for the steady-state error derived from droop control. AGC [35] implements the secondary control layer, collecting information from all generation units in a centralised way. Finally, the tertiary control layer is related to the economic aspect of power system operations. This layer establishes the load share between the sources so that the operational costs are minimised [36]. Tertiary control is implemented through the economic dispatch, which calculates the optimal operating point in an offline process. Next, we present two models, that is, Model I and Model II, for the description of the power system dynamics. These two models will be used to formulate the frequency control problem of a power system with n generators denoted by $\mathcal{G} = \{G_1, \ldots, G_n\}$.

2.1 | Model I: Balancing authority area dynamics

It is common in power systems' operations to model the dynamic behaviour of the entire BA area instead of each individual generator. In this regard, we define by $\Delta\omega$ the deviation of the centre of inertia speed from the synchronous speed; the total mechanical power produced by $P_{SV} = \sum_{i\in\mathcal{G}} P_{SV_i}$, with P_{SV_i} the mechanical power of generator i; and the total secondary command by $Z_G = \sum_{i\in\mathcal{G}} z_i$, with z_i the participation of generator i to AGC. Then the BA area dynamics are as follows:

$$M\frac{d\Delta\omega}{dt} = P_{SV} - P_G - D\Delta\omega, \tag{1}$$

$$T_{SV}\frac{dP_{SV}}{dt} = -P_{SV} + Z_G - \frac{1}{R_D}\Delta\omega, \tag{2}$$

where $M = \frac{2H}{\omega_s}$, with H denoting the system inertia constant and ω_s the synchronous speed; $T_{SV} = \frac{\sum_{i\in\mathcal{G}} T_{SV_i}}{n}$, where T_{SV_i} is the time constant of the mechanical power dynamics of generator i; $D = \sum_{i\in\mathcal{G}} D_i$, with D_i representing the machine i damping coefficient; $\frac{1}{R_D} = \sum_{i\in\mathcal{G}} \frac{1}{R_{D_i}}$, with R_{D_i} representing the governor droop of generator i. We neglect the network effects and set $P_G = P_L(1 + \rho)$, where P_L is the system load and ρ denotes the sensitivity of the losses with respect to the system load. The normalised participation factor of bus load changes ΔP_{L_i} with respect to the total system load change ΔP_L is denoted by σ_i, the output of generator i, P_i, and then ρ, which denotes the sensitivity of the losses with respect to the system load is

$$\rho = \sum_{i\in\mathcal{G}} \sigma_i \frac{\partial P_{\text{losses}}}{\partial P_i}. \tag{3}$$

2.2 | Model II: Synchronous generator dynamics

In Model II, the individual generators' dynamics are represented. For the ith synchronous generator, the three states are the rotor electrical angular position δ_i, the deviation of the rotor electrical angular velocity from the synchronous speed $\Delta\omega_i$, and the mechanical power P_{SV_i}. We denote by z_i the participation of each generator i in AGC. The evolution of the three states of the generator i is determined by the following:

$$\frac{d\delta_i}{dt} = \Delta\omega_i, \tag{4}$$

$$M_i\frac{d\Delta\omega_i}{dt} = P_{SV_i} - P_i - D_i\Delta\omega_i, \tag{5}$$

$$T_{SV_i}\frac{dP_{SV_i}}{dt} = -P_{SV_i} + z_i - \frac{1}{R_{D_i}}\Delta\omega_i, \tag{6}$$

where the inertia constant is H_i, the synchronous speed is ω_s and $M_i = \frac{2H_i}{\omega_s}$,; the machine damping coefficient is D_i, the governor droop is R_{D_i},; and the parameter z_i is an input provided by the AGC. The definitions of the machine parameters may be found in [34]. The output of generator i P_i is determined by Equation (11).

2.3 | Network

Let us consider a power system with N nodes and P_{L_i} represents the real power load at bus i. Further, let Q_i and Q_{L_i} denote the reactive power supplied by the synchronous generator and demanded by the load at bus i, respectively. Then, we model the network using the standard non-linear power flow formulation (see e.g. [34]); thus, for the ith bus, we have that

$$P_i - P_{L_i} = V_i \sum_{k=1}^{N} V_k (G_{ik}\cos\theta_{ik} + B_{ik}\sin\theta_{ik}), \tag{7}$$

$$Q_i - Q_{L_i} = V_i \sum_{k=1}^{N} V_k (G_{ik}\sin\theta_{ik} - B_{ik}\cos\theta_{ik}), \tag{8}$$

where $G_{ik} + jB_{ik}$ is the (i, k) entry of the network admittance matrix and $\theta_{ik} = \theta_i - \theta_k$.

We assume that (i) bus voltage magnitudes are $|V_i| = 1$ p.u for $i = 1, ..., N$, (ii) lines are lossless and characterised by their susceptances $B_{ik} = B_{ki} > 0$ for $i, k = 1, \ldots, N$ with $i \neq k$, (iii) reactive power flows do not affect bus voltage phase angles and frequencies and (iv) the coherency between the internal and terminal voltage phase angles of each generator so that these angles tend to *swing together*, that is, $\delta_i = \theta_i$. As a result, we neglect Equation (8) and simplify Equation (7) to be the following:

$$P_i - P_{L_i} = \sum_{\substack{k=1 \\ i\neq k}}^{N} B_{ik}(\delta_i - \delta_k). \tag{9}$$

If bus i does not contain a generator then $P_i = 0$.

In order to increase the accuracy of Equation (9), we can slightly modify it by incorporating an approximation of the losses. We define the normalised participation factor of bus load changes ΔP_{L_i} with respect to the total system load change ΔP_L by σ_i and then ρ_i, which denotes the sensitivity of the losses with respect to the system load at bus i and is

$$\rho_i = \sigma_i \frac{\partial P_{\text{losses}}}{\partial P_i}. \tag{10}$$

Then Equation (9) becomes the following:

$$P_i - (1 + \rho_i)P_{L_i} = \sum_{\substack{k=1 \\ i\neq k}}^{N} B_{ik}(\delta_i - \delta_k). \tag{11}$$

2.4 | Economic dispatch

The economic dispatch process is formulated as an optimisation problem, where the objective function that needs to be minimised is the sum of the individual costs of all generating units, $c_i(P_i)$, for $i \in \mathcal{G}$; this is typically a quadratic function that computes the production cost of each generation unit. Here, the constraint is that the system has to keep the generation and load balanced; if the generation and load are balanced then frequency is also nominal. The economic dispatch problem may be formulated as follows:

$$\begin{aligned} &\underset{P_i}{\text{minimize}} \sum_{i\in\mathcal{G}} c_i(P_i) \\ &\text{subject to} \sum_{i\in\mathcal{G}} P_i = (1 + \rho)P_L. \end{aligned} \tag{12}$$

2.5 | Wind generation

The increasing penetration of renewable-based resources in the system introduces a source of uncertainty in power system operations and thus in LFC problems. In this regard, we investigate the effect of wind generation units in the proposed framework. The relationship between the wind speed and the generated power can be efficiently modelled as a linear dynamical system [37]. More specifically, P_W denotes the real wind generation power output, Δv the variation of the wind speed, α_{W_1} and α_{W_2} are parameters that depend on the wind turbine characteristics, W_t is a Wiener process and β_{W_1} and β_{W_2} are coefficients that represent prior knowledge of the wind speed probability distribution. Then, the dynamics of the wind generation power output are formulated as follows:

$$\frac{d\Delta P_W}{dt} = \alpha_{W_1}\Delta P_W + \alpha_{W_2}\Delta v, \tag{13}$$

$$d\Delta v = \beta_{W_1}\Delta v dt + \beta_{W_2} dW_t. \tag{14}$$

3 | MULTI-AGENT REINFORCEMENT LEARNING FOR LOADFREQUENCY CONTROL

In this section, we formulate the LFC problem as an MARL problem. RL is an area of Machine Learning strongly related with the notion of software agents [38]. RL studies how autonomous agents interact with the environment to maximise their long-term performance. We use MARL to train a collection of agents on how to implement the LFC problem in a distributed way. In this article, an agent physically represents the controller of a generation unit. As such, by using the MARL scheme, which allows for a fully distributed control architecture, LFC can be achieved with no communication infrastructure between the agents. The controller's physical circuit of each generator does not have to be connected with any of the other controllers, thus allowing a physical "distribution" of the LFC system. A diagram of the entire architecture and how the RL-based AGC design fits in the power system dynamics is shown in Figure 1.

RL problems are mathematically formalised through an MDP [39] that is defined as the tuple:

$$MDP = \langle S, A, P, R \rangle, \tag{15}$$

where each term is as follows:

- *S or state space*: all possible states where the agent can be in the environment. There are two continuous states in

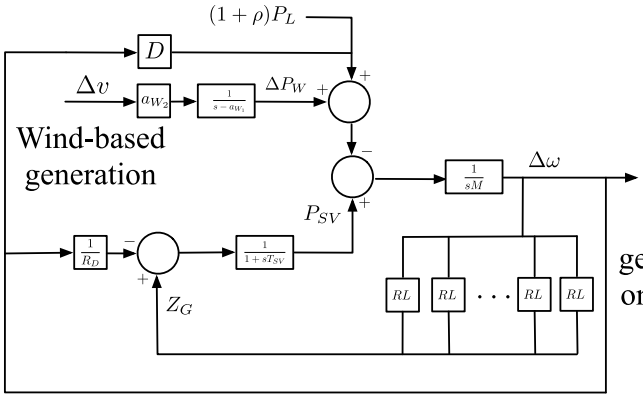


FIGURE 1 Diagram of the proposed load frequency control scheme

LFC: the deviation from synchronous speed, which is quantified by $\Delta\omega_i$ for each generator i or by $\Delta\omega$ in the case of the BA area model, and z_i, the current control action of each generator i. These states provide to the agent information about the difference between demand and supply and how much they are contributing to the total generation.

- *A or action space*: all possible actions that an agent can take in each state. Our agent generators can increase or decrease the control action z_i in order to modify the state of the environment.
- *P or probability state transition function*: it defines the dynamics of the environment, modelling the transition between states. For the BA area model or Model I described in Section 2.1, the transition equations derived from Equations (1) and (2) are as follows:

$$M\frac{d\Delta\omega^{\text{new}}}{dt} = P_{SV}^{\text{old}} - (1+\rho)P_L - D\Delta\omega^{\text{old}}, \tag{16}$$

$$T_{SV}\frac{dP_{SV}^{\text{new}}}{dt} = -P_{SV}^{\text{old}} + Z_G^{\text{new}} - \frac{1}{R_D}\Delta\omega^{\text{old}}, \tag{17}$$

$$Z_G^{\text{new}} = \sum_{i\in\mathcal{G}} z_i^{\text{new}}, \tag{18}$$

$$z_i^{\text{new}} = z_i^{\text{old}} + \Delta z_i, \tag{19}$$

$$\Delta\omega^{\text{new}} = \Delta\omega^{\text{old}} + \frac{d\Delta\omega^{\text{new}}}{dt}\Delta t, \tag{20}$$

$$P_{SV}^{\text{new}} = P_{SV}^{\text{old}} + \frac{dP_{SV}^{\text{new}}}{dt}\Delta t. \tag{21}$$

For the detailed modelling of Model II given in Section 2.2, the transition equations based on Equations (4–6) and (11) are as follows:

$$\frac{d\delta_i^{\text{new}}}{dt} = \Delta\omega_i^{\text{old}}, \tag{22}$$

$$M_i\frac{d\Delta\omega_i^{\text{new}}}{dt} = P_{SV_i}^{\text{old}} - P_i - D_i\Delta\omega_i^{\text{old}}, \tag{23}$$

$$T_{SV_i}\frac{dP_{SV_i}^{\text{new}}}{dt} = -P_{SV_i}^{\text{old}} + z_i^{\text{new}} - \frac{1}{R_{D_i}}\Delta\omega_i^{\text{old}}, \tag{24}$$

$$z_i^{\text{new}} = z_i^{\text{old}} + \Delta z_i, \tag{25}$$

$$\Delta\omega_i^{\text{new}} = \Delta\omega_i^{\text{old}} + \frac{d\Delta\omega_i^{\text{new}}}{dt}\Delta t, \tag{26}$$

$$\delta_i^{\text{new}} = \delta_i^{\text{old}} + \frac{d\delta_i^{\text{new}}}{dt}\Delta t, \tag{27}$$

$$P_{SV_i}^{\text{new}} = P_{SV_i}^{\text{old}} + \frac{dP_{SV_i}^{\text{new}}}{dt}\Delta t, \tag{28}$$

$$P_i - (1+\rho_i)P_{L_i} = \sum_{\substack{k=1 \\ i\neq k}}^{N} B_{ik}(\delta_i^{\text{new}} - \delta_k^{\text{new}}), \tag{29}$$

where Δz_i is the increase or decrease in power generation by each unit i in $\mathcal{G}$ estimated by each agent. MADDPG is used to estimate Δz_i, as described in Section 4.

- *R or reward function*: it defines a numerical signal or reward expressing the value of being in a state and performing an action. The reward function considers two different dimensions in our case: frequency deviation and operational costs. The specific rewards are defined in Section 4.

MARL attempts to learn an optimal policy π: $S \mapsto A$ that maximises the cumulative reward or return. However, the reward is instantaneous and does not address the global nature of the task, that is, one bad action can lead to an extremely good position from which the agent can obtain a good reward. Thus, action-value functions Q^{π} are used in RL to express the

expected long-term reward achievable from being in a state, taking an action and following a policy π:

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[R_t | s_t, a_t] = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t, a_t\right], \quad (30)$$

where $\mathbb{E}[\cdot]$ is the expectation operator, γ is the discount factor, which expresses the fidelity in long-term predictions of Q^{π}, the cumulative reward achievable in the long run R_t, and the reward r_t at time t. Most RL algorithms use value functions, such as Q-learning [40]. To support the learning process, the action-value function associates a value Q^{π} to each state-action pair. However, when the number of states and actions is very large, it becomes computationally expensive to estimate them efficiently. Recent work has merged the field of RL with Deep Learning, giving birth to a powerful algorithm called Deep Q-learning [41]. This algorithm uses deep neural networks as parametric function approximators to estimate the action-value function of each state-action pair.

The spectrum of existing algorithms to solve MARL problems is wide. Most of them use game-theoretic approaches to augment Q-learning, that is, Nash Q-learning or minimax Q-learning [42]. In our case, state and action spaces are continuous and the interaction of various agents is required. This limits the range of algorithms available in the literature. MADDPG addresses both the constraints at the same time [43].

4 | MULTI-AGENT DEEP DETERMINISTIC POLICY GRADIENT

In this section, we present a multi-agent actor-critic algorithm that takes into account the design of the reward function and the fact that state and action spaces are continuous and.

MADDPG is an actor-critic algorithm. This means that the architecture of each agent, or generation unit, is split into two. First, the actor directly estimates an action and then, the critic assesses the value of the action by estimating the action-value function Q^{π} of the state-action pair. The Q^{π} estimated by the critic is used by both the critic and the actor to learn how to behave in the environment. In MADDPG, the critics use central information to teach each actor the dynamics of the environment as well as the behaviour of the rest of the agents. In operation, actors only use local information because they have learnt how other actors will behave. As such, no communication infrastructure between agents is necessary.

We describe here the actor-critic algorithm for the BA model or Model I. This is the same for Model II presented in Section 2.2; the only difference is that instead of the deviation from the centre of inertia, in Model II, the input to the actor and the critic is the deviation of the rotor speed from the synchronous speed of each generator $\Delta\omega_i$. For the BA area model given in Section 2.1, we have each actor i that estimates Δz_i, given the state of the environment $\Delta\omega$ and its current z_i. Each critic assesses each state-action pair defined by the environment and the actions of all the actors. The critic estimates each state-action value that is used during the actor's training, as can be seen in Figure 2. We denote by Δz_{-i} (Δz_{-j}), the action predicted by all other actors besides i (j), and z_{-i} (z_{-j}) is the control action of all other actors besides i (j).

Deep Recurrent Neural Networks, specifically LSTMs [44], are used to model each actor and critic. LSTMs implement memory so that previous history is stored and acted upon [45]. In MDPs, the Markov assumption dictates that the current state comprises all the information needed to choose an action. However, in the frequency control problem the dynamics are quite complex and the Markov assumption may not hold. Thus, LSTMs help to correct the violation of the Markov assumption.

The actor network, see Figure 3, has as inputs $\Delta\omega$ and z_i and computes Δz_i. The critic network, depicted in Figure 4, has as inputs the frequency state of the network $\Delta\omega$, the secondary control action z_i, the change in the action predicted by the actor associated to that critic Δz_i, the secondary control action z_{-i}, and the change in the action predicted by all other actors Δz_{-i}. The critic network then computes the $Q^{\pi}(\cdot)$ value of the state-action pair estimated by the actor associated with that critic. As seen in the respective figures, both networks consist of 100-neuron LSTM that implements memory and three more 1000, 100 and 50 fully connected hidden layers. Generation rate constraints can be easily introduced in this neural network

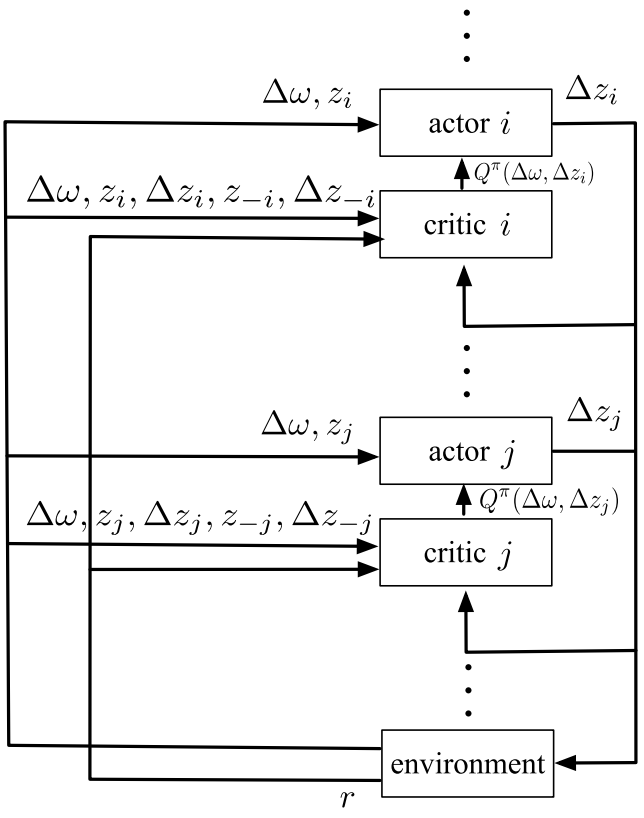


FIGURE 2 MADDPG schema in a frequency control scenario

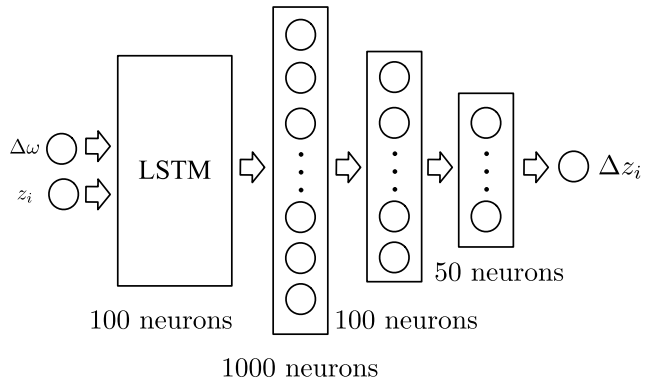


FIGURE 3 Architecture of the MADDPG actor

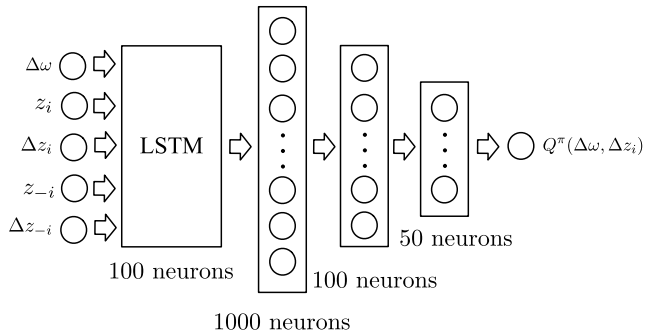


FIGURE 4 Architecture of the MADDPG critic

based approach. More specifically, the output Δz_i of each actor can be bounded by applying a non-linear function (e.g. sigmoid function, hyperbolic tangent etc.) at the output of the network; so the agent has to learn that it cannot generate at an unrealistic rate.

The proposed method includes offline centralised learning towards the global objective of frequency restoration in a cost-efficient way and online distributed implementation by only using local information. As such, no communication infrastructure is necessary between the agents as the implementation of the proposed method is based only on local information. In particular, during the training phase the critics collect the information of all the generation units. Then, in the operation phase each ith actor observes the frequency state of the network $\Delta\omega$ and its own secondary control signal z_i, both of which are local variables. The action taken by the ith actor is the change in its own secondary control action Δz_i. None of the actors need the secondary control action $z_j, j \neq i, j \in \mathcal{G}$ of the rest of the actors to operate. Contrary to traditional ACG, where all secondary control signals z_i, $\forall i \in \mathcal{G}$ are centralised, the proposed framework operates in a fully distributed manner. In operation, there is no communication between agents/actors. The implications of this point are, on the one hand, that we do not have to consider synchronisation issues, communication burden or information leaks. On the other hand, privacy is completely guaranteed since each generation unit does not share any type of information with the rest of the generation units during operation.

The design of the reward function is critical, since it determines the behaviour that agents will learn. Designing a reward function is much of an art as it is strongly problem-dependent. One principle is to effectively reflect the control goal, which in our case is to restore the system frequency and achieve this with minimum cost after net load disturbances occur. As such, usually quadratic, exponential, absolute value and other sophisticated reward functions are used (see e.g. [20, 26]). Ideally, the reward function of this problem incorporates two different components: (i) the frequency state of the environment to solve the primary and secondary problem and (ii) the operational cost associated with the system to solve the tertiary control problem. Incorporating the frequency component in the reward function is straightforward since we set a higher reward for smaller frequency deviations. Next, we need to determine how the reward function can be defined in order to take into account the cost component. In this regard, we study the case where the cost functions of generators are of the form $c_i(P_i) = a_i P_i^2 + \beta_i P_i + \gamma_i$, for $i \in \mathcal{G}$ [4]. Cost minimisation is part of the tertiary control in the hierarchical control setting, the formulation of which may be found in (12). For quadratic cost functions under no generation limits we can find the optimal solution in an analytical way [33]. The Lagrangian may be written as

$$\mathcal{L}(P_i, \lambda) = \sum_{i \in \mathcal{G}} c_i(P_i) + \lambda \left((1+\rho) P_L - \sum_{i \in \mathcal{G}} P_i \right),$$

where λ is the dual variable of the power balance constraint. The conditions necessary for a minimum are

$$\frac{\partial \mathcal{L}}{\partial P_i} = 0 \Rightarrow \frac{dc_i}{dP_i} - \lambda = 0 \Rightarrow 2a_i P_i + \beta_i = \lambda, \forall i \in \mathcal{G}. \quad (31)$$

The solution to the problem above defines the base point operation of tertiary control. We now define with the aid of participation factors how a generator would participate in a load change so that the new load is served in a cost-efficient way. We start from a given base point λ_0 as found from (31). Assume that the change in load is ΔP_L; the system incremental cost moves from λ^0 to $\lambda^0 + \Delta\lambda$. For a small change in power output on unit i, ΔP_i, we have

$$\Delta\lambda \approx \frac{d^2 c_i}{dP_i^2} \Delta P_i \Rightarrow \Delta P_i = \frac{\Delta\lambda}{\frac{d^2 c_i}{dP_i^2}}, \forall i \in \mathcal{G}. \quad (32)$$

Thus, we require that each generator i changes its output so that the following holds:

$$\Delta\lambda = \frac{d^2 c_i}{dP_i^2} \Delta P_i = \frac{d^2 c_j}{dP_j^2} \Delta P_j, \forall i, j \in \mathcal{G}, \quad (33)$$

that is, for each generator a change in the action Δz_i, where for $i \in \mathcal{G}$ we require that

$$\left|\Delta z_i \frac{d^2 c_i}{dP_i^2} - \Delta z_j \frac{d^2 c_j}{dP_j^2}\right| = 0, \forall i, j \in \mathcal{G}. \tag{34}$$

Now we use these two conditions, that is, frequency deviation and cost information, to determine the reward functions for each modelling approach.

4.1 | Reward function: Model I

We construct two conditions that will be used in the formulation of the reward function. The first condition is as follows:

$$C1 : |\Delta\omega| < \epsilon_1,$$

where ϵ_1 is some selected tolerance; this condition ensures that the reward function r will reward actions that help in frequency restoration. The second condition is as follows:

$$C2 : \frac{\sum_{i\in\mathcal{G}}\sum_{j\in\mathcal{G}, j>i}\left|z_i \frac{d^2 c_i}{dP_i^2} - z_j \frac{d^2 c_j}{dP_j^2}\right|}{(n-1)!} < \epsilon_2,$$

where ϵ_2 is some selected tolerance; this condition ensures that r will reward actions that follow the cost-efficient path.

When only the primary and secondary control problems need to be solved, the reward function may be formulated using $C1$ as

$$r = \begin{cases} d, & \text{if } C1, \\ 0, & \text{otherwise,} \end{cases}, \tag{35}$$

where d is a constant. On the other hand, by taking these two conditions into account we may formulate a general form of the reward function to solve all the levels of control, from primary to tertiary as

$$r = \begin{cases} d_1, & \text{if } C1 \wedge C2, \\ d_2, & \text{if } C1 \vee C2, , \\ 0, & \text{otherwise,} \end{cases} \tag{36}$$

where $\wedge$ is the logical *and*; $\vee$ is the logical *or*; and d_1, d_2 are constants with $d_2 < d_1$. This reward function facilitates frequency restoration in a cost-efficient way, since the critic values actions higher if the frequency of the system is close to nominal and the cost is small.

4.2 | Reward function: Model II

In order to ensure frequency restoration, that is, secondary control, we require that $|\Delta\omega_i| < \epsilon$ for all $i \in \mathcal{G}$. To this end, we formulate the reward function as follows:

$$r = \begin{cases} d'_1, & \exists i : |\Delta\omega_i| < \epsilon \\ d'_2, & \exists i, j : j \neq i, |\Delta\omega_i| \wedge |\Delta\omega_j| < \epsilon, \\ d'_3, & \exists i, j, j' : j \neq j' \neq i, |\Delta\omega_i| \wedge |\Delta\omega_j| \wedge |\Delta\omega_{j'}| < \epsilon, \\ \vdots & \\ d'_n, & |\Delta\omega_1| \wedge |\Delta\omega_2| \wedge \cdots \wedge |\Delta\omega_n| < \epsilon, \\ 0, & \text{otherwise} \end{cases}, \tag{37}$$

where $\wedge$ is the logical *and* sign and $d'_1, d'_2, \ldots, d'_n$ are constants with $d'_1 < d'_2 < d'_3 < \cdots < d'_n$. This formulation ensures

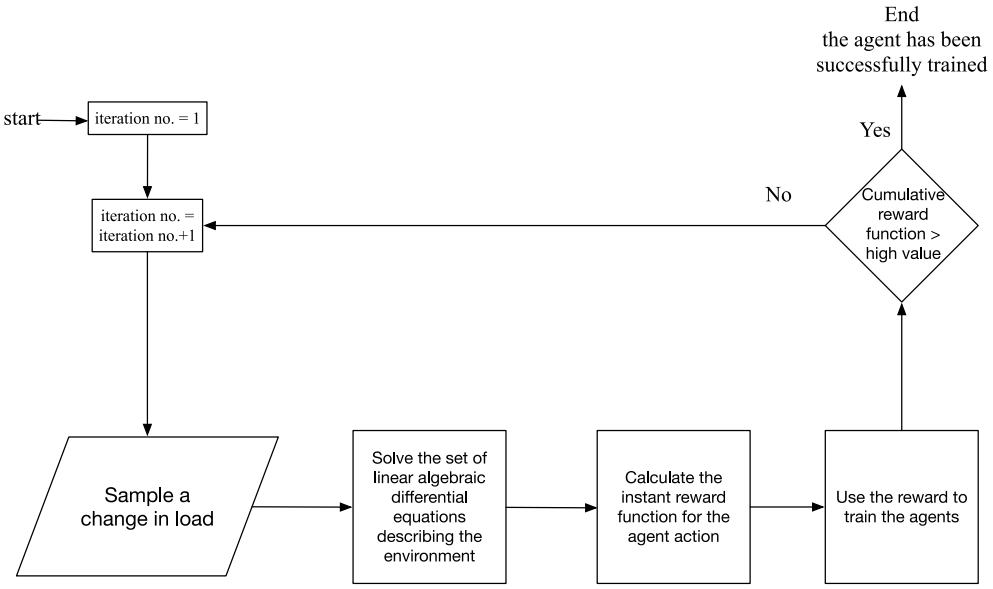


FIGURE 5 Diagram of the training process for developing the proposed load frequency control scheme

that the reward is higher when the frequency deviation is smaller than a specified tolerance. In this work, we have not performed tertiary control for Model I. The reward function integrates the training loop, informing each agent whether its actions are desirable or not. After each time step, the agents receive a reward based on the state of the environment and the actions taken. Then, these rewards are used to optimise the weights of the networks of the actors and the critics. A flow diagram of the training process involved in determining the RL-based control for each agent is shown in Figure 5.

TABLE 1 Eight-generator and one-load power system data

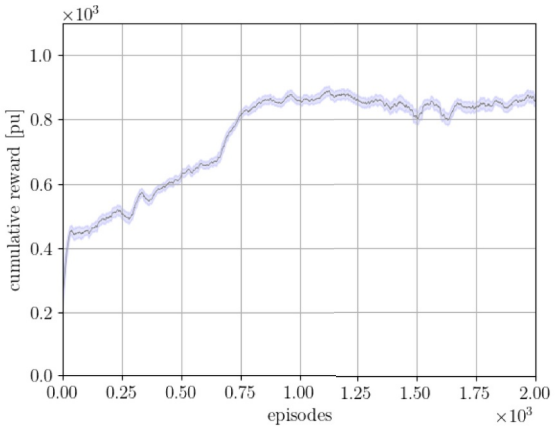
Nominal frequency	$f^{\text{nom}} = 50$ Hz
Initial operating point	$P_i = 0.375$ pu, $i = 1, ..., 8$
Inertia parameter	$M = 0.1$ pu
Droop	$R_D = 0.1$ pu
Load damping	$D = 0.0160$ pu
Generator time constant	$T_{\text{SV}} = 30$ s

5 | NUMERICAL RESULTS

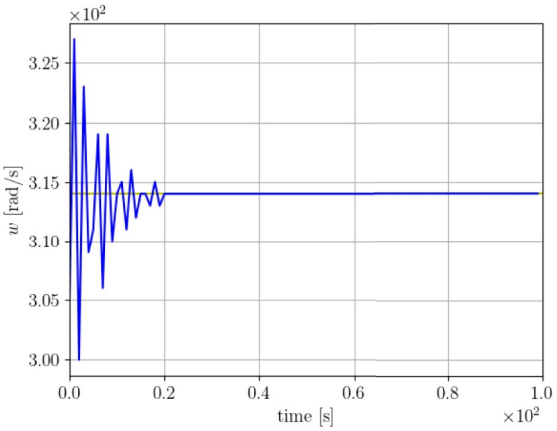
We validate the MARL methodology using three test systems. We formulate the reward function and present the results of the primary and secondary control problems for Model I and for the detailed modelling of Model II, taking into account the network effects. We also demonstrate the flexibility of the proposed methodology to incorporate generation rate constraints as well as its robustness against the uncertainty introduced due to wind generation. Then, we formulate the reward function and present the results of all levels of control for one single BA area using Model I. We demonstrate that the generators are able to restore the system frequency back to nominal and operate at a point close to optimal when a change in load occurs in a distributed way. We compare the results with a standard distributed optimal load frequency controller [14].

5.1 | Secondary control: Model I

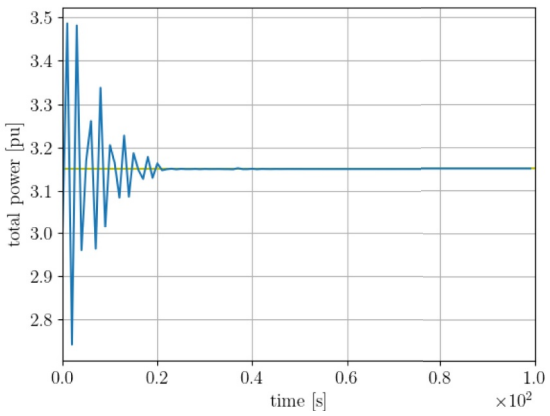
The test case used to validate the secondary control using Model I comprises a group of eight generating units or agents



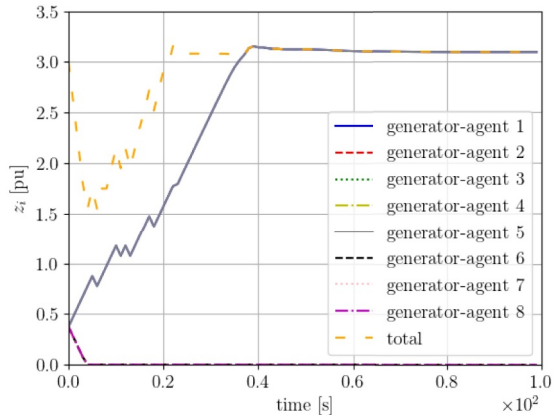
(a) Smoothed cumulative reward per episode with 95% confidence levels.



(b) Centre of inertia speed.



(c) Power response of the eight agents.



(d) Secondary control action of the eight agents

FIGURE 6 Secondary control Model I: Change in load by 0.15 pu

that interact with a load. The parameters of the environment can be found in Table 1. In each training episode, the load varies around a nominal set point randomly. The modification is indicated by $P_L \pm \Delta P_L = 3 \pm \beta$ pu, where β follows a uniform distribution. The reward function has been derived following (35) and is defined by the following:

$$r = \begin{cases} 10, & \text{if } C1 \\ 0, & \text{otherwise} \end{cases}.$$

During operation, only the actors interact with the environment. They can only observe the local information about the frequency of the system and the control action that they are executing. Following training, they know how to act according to the state of the environment in order to keep the load and generation balanced. The validation of training is tested by changing the load by 0.15 pu and observing how the generators modify their output.

TABLE 2 Two-generator and one-load power system data

Nominal frequency	$f^{nom} = 50$ Hz
Initial operating point	$P_1 = 1.5$ pu, $P_2 = 1.5$ pu
Inertia parameter	$M = 0.1$ pu
Droop	$R_D = 0.1$ pu
Load damping	$D = 0.0160$ pu
Generator time constant	$T_{SV} = 30$ s

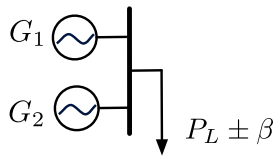


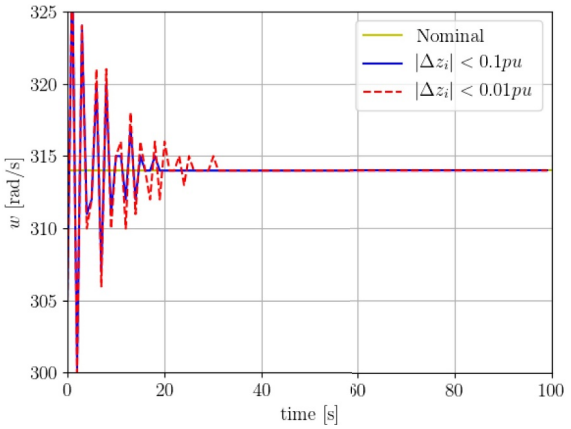
FIGURE 7 One-line diagram of a two-generator and one-load power system

In Figure 6a, the cumulative reward obtained by the agents is depicted. The agents can obtain 1000 at maximum per episode, that is, the maximum reward per step is 10 and the number of steps per episode is 100. The agents learn how to obtain higher rewards as the number of episodes increases, since if that was not the case, the cumulative reward function would oscillate around small values near zero.

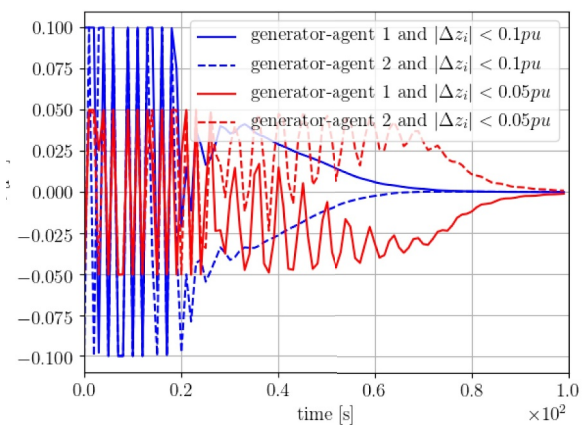
In Figure 6b,c, the centre of inertia speed and power response of an eight-generator system when a single load increases by 0.15 pu is depicted. However, as can be seen in Figure 6d, the solution may be unrealistic given that the operational cost component is neglected in this test case. As such, one agent learns to balance the entire system while the others have zero output. Further analysis on secondary control using Model I may be found in [20].

In order to highlight the ability of the proposed framework to implement generation rate constraints, we have used Model I to conduct numerical analyses of the response of a two-agent system, whose data can be seen in Table 2 and is depicted in Figure 7, when the generation rate of each unit is bounded by different values. We modify the load by 0.15 pu and limit the output of each actor Δz_i to 0.1, 0.05 and 0.01 pu, respectively, by using a hyperbolic tangent function. As depicted in Figure 8a, although the system manages to meet the new load, the generation rate constraints affect the elapsed time until the new steady state is reached. This can be also inferred by observing Figure 8b, where the actual values of Δz_i are shown. When the generation rate is more constrained, that is, the generators are allowed to modify their output in smaller increments, the system spends more time balancing the generation and demand, as expected.

Next, we validate that the proposed framework is robust against the uncertainty introduced by wind generation, as described in Section 2. We model a wind generator as a stochastic process with parameters $\alpha_{W_1} = -0.002$, $\alpha_{W_2} = 0.01$, $\beta_{W_1} = -0.5$, and $\beta_{W_2} = -0.4$. We train the two-agent system of Table 3 to balance the load under such conditions. In Figures 9a,b, it can be seen that the load is met and the frequency is close to nominal under the scenario that the wind generation



(a) Centre of inertia speed.



(b) Secondary control action of the two agents.

FIGURE 8 Secondary control Model I: change in load by 0.15 pu, with Δz_i bounded at different levels

evolves randomly. More specifically, minor variations appear in the frequency response as the agents adapt to rebalance the load.

5.2 | Secondary control: Model II

Analogously, we have designed a test case to validate the performance of the proposed solution using the detailed Model II. The dynamic behaviour of two generators that are part of a BA area as well as the network is explicitly taken into account. The configuration of the system that has two loads, that is, P_{L_1} and P_{L_2}, can be found in Figure 10. The parameters of the environment can be found in Table 3. In each training episode, each load varies around a nominal set point randomly. The modification of each load is indicated by $P_{L_i} \pm \Delta P_{L_i} = 1.5 \pm \beta$ pu, where β follows a uniform distribution.

The reward function has been derived following Equation (37). We set $\epsilon = 0.05$ pu, $d_1' = 100$, and $d_2' = 200$. The reward function is formulated as follows:

$$r = \begin{cases} 100, & \exists i : |\Delta\omega_i| < 0.05 \\ 200, & |\Delta\omega_1| \wedge |\Delta\omega_2| < 0.05 \\ 0, & \text{otherwise} \end{cases}.$$

Figure 11a shows the cumulative reward obtained by the agents during training. Again, we notice that the agents are learning and have discovered how to obtain higher rewards. In this case, the agents learn how to jointly balance generation and demand.

Following the same schema, we change both loads by 0.15 pu and observe how the frequency and the output of each generator change. The rotor electrical angular velocity of each generator is restored, as can be seen in Figure 11b. The generation output of the two generators is depicted in Figure 11c,d; Figure 11c is a zoomed-in version of Figure 11d. In Figure 11c, where the timescale is up to 100 s, we notice that the total power of the generators meets the new load, thus restoring frequency. However, the secondary control system sends signals to the generators to modify their output, as seen in Figure 11d,e. The system frequency is nominal since, even if the output of the two generators changes, the summation of the output remains constant and equal to the new load.

We have demonstrated that the proposed framework can be applied to solve primary and secondary control problems, with the detailed modelling of Model II. This is achieved in a distributed way, that is, without centralising any kind of information the agents learn how to balance the system. Here, the agents learn that keeping $\Delta\omega_i$ close to 0 for all the generators is associated with high rewards.

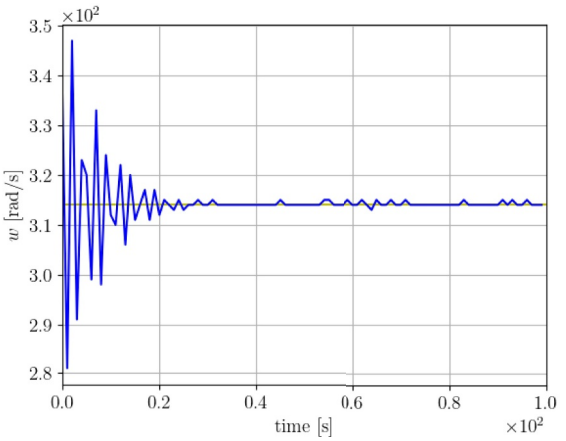
TABLE 3 Two-generator and two-load power system data

Nominal frequency	$f^{\text{nom}} = 50$ Hz
Initial operating point	$P_1 = 1.5$ pu, $P_2 = 1.5$ pu
Inertia parameter	$M_1 = 0.1$ pu, $M_2 = 0.15$ pu
Droop	$R_{D_1} = 0.1$ pu, $R_{D_2} = 0.08$ pu
Load damping	$D_1 = 0.0160$ pu, $D_2 = 0.0180$ pu
Generator time constant	$T_{SV_1}, T_{SV_2} = 30$ s

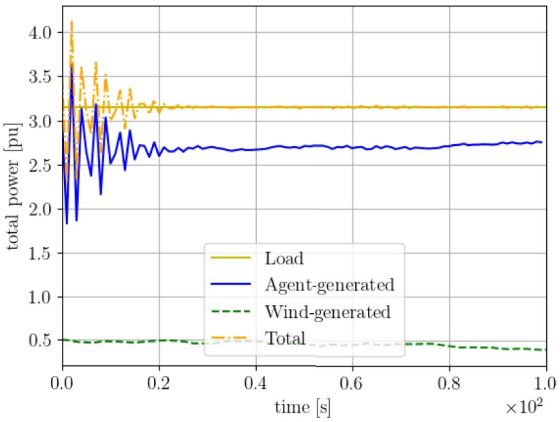
5.3 | Tertiary control: Model I

The test case designed to check the performance of all levels of LFC in a single BA area comprises two generation units or agents that interact with a load whose configuration during training can be found in Figure 7. The parameters of the environment are specified in Table 2, with cost functions for generator 1 $c_1 = 2P_1^2$ [£/pu] and generator 2 $c_2 = P_2^2$ [£/pu]. In each episode, or training simulation, the load varies randomly around a nominal set point. The load varies as $P_L \pm \Delta P_L = 3 \pm \beta$ pu, where β follows a uniform distribution.

The reward function has been derived following (36). We set $\epsilon_1 = 0.05$ pu, $\epsilon_2 = 0.2$ pu, $d_1 = 200$, and $d_2 = 100$. Thus, we have two conditions as follows:



(a) Centre of inertia speed.



(b) Secondary control action of the two agents.

FIGURE 9 Secondary control Model I: change in load by 0.15 pu with wind generation

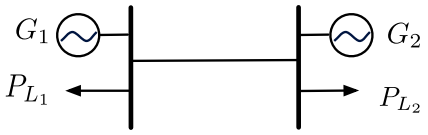


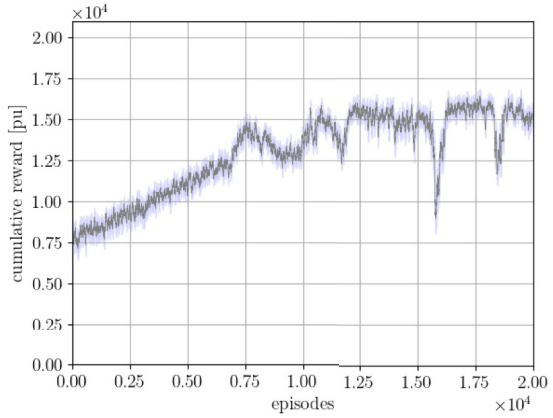
FIGURE 10 One-line diagram of a two-generator and two-load power system

$$C1 : |\Delta\omega| < 0.05,$$

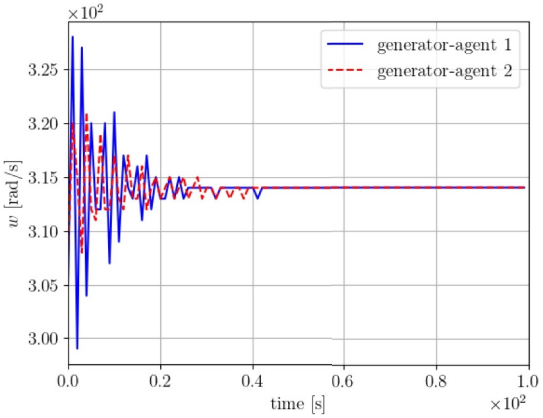
and

$$C2 : |2z_1 - z_2| < 0.2.$$

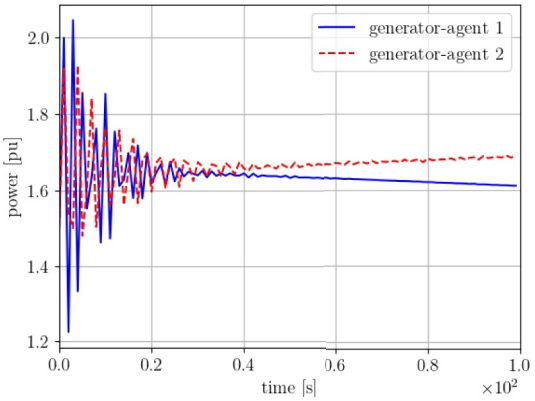
Taking these two conditions into account, we may formulate the reward function as



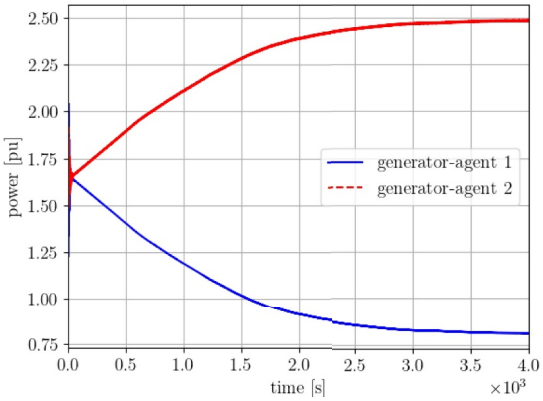
(a) Smoothed cumulative reward per episode with 95% confidence levels.



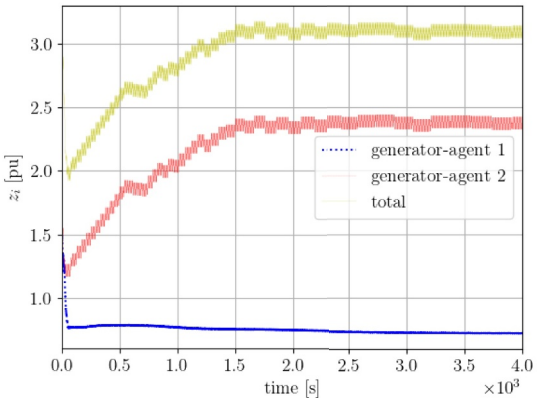
(b) Rotor electrical angular velocity of the two generators.



(c) Generators' output in the first 100 s.

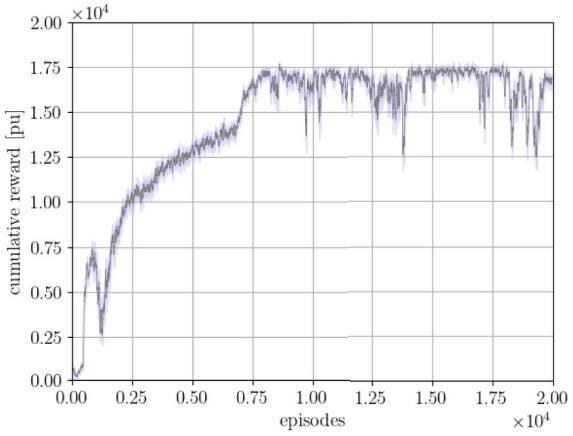


(d) Generators' output.

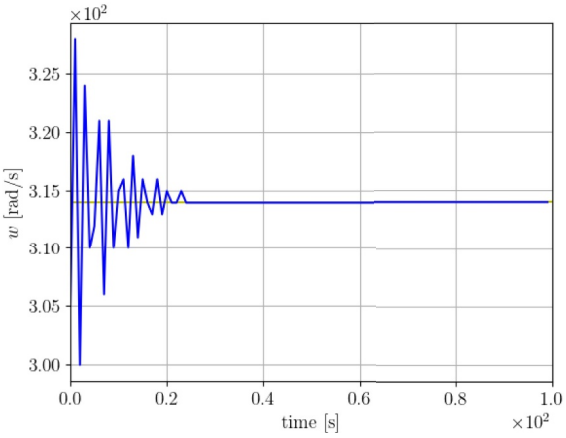


(e) Secondary control action.

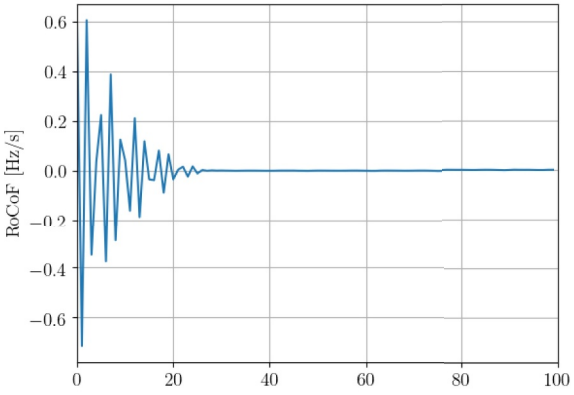
FIGURE 11 Secondary control Model II: change in load by 0.15 pu



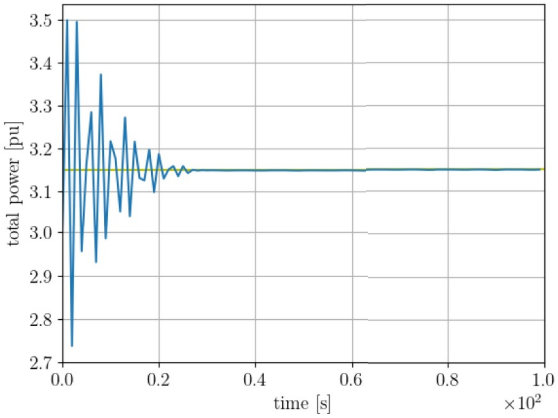
(a) Smoothed cumulative reward per episode with 95% confidence levels.



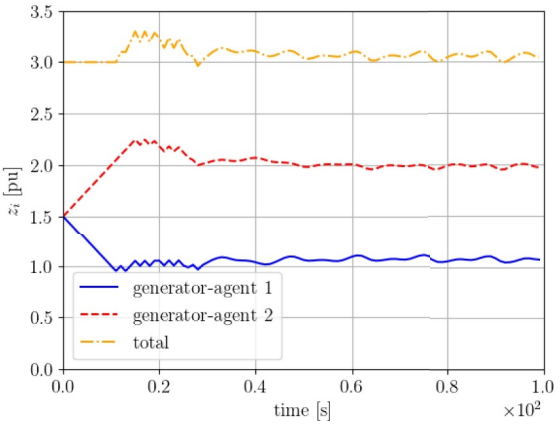
(b) Centre of inertia speed.



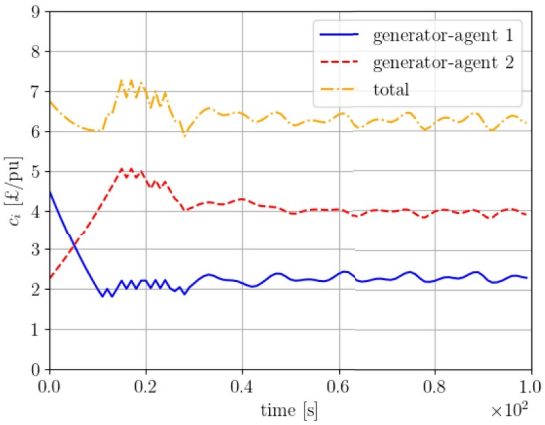
(c) RoCoF.



(d) Total power.



(e) Secondary control action.

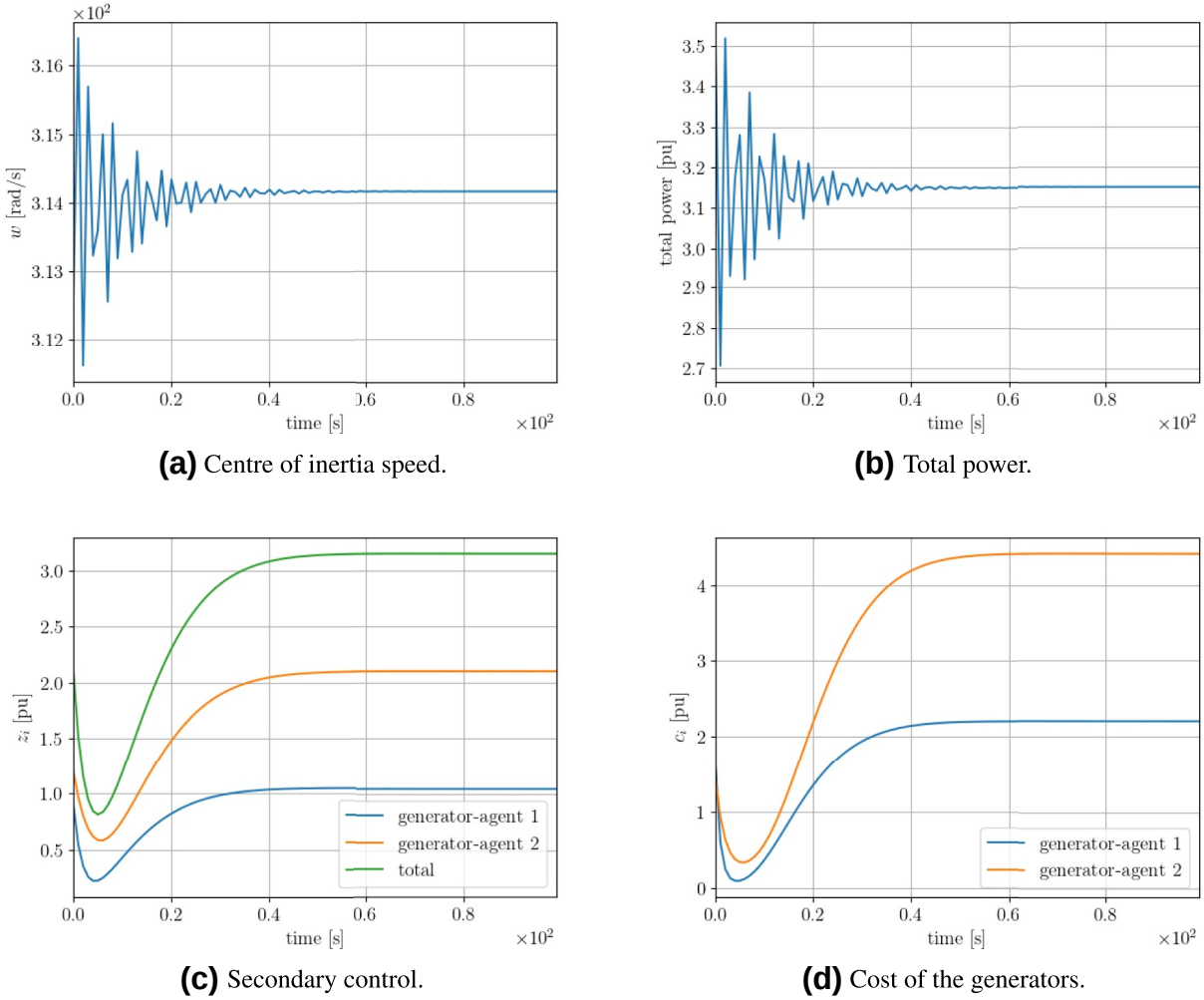


(f) Cost of the generators.

FIGURE 12 Tertiary control Model I: change in load by 0.15 pu

$$r = \begin{cases} 200, & \text{if } C1 \wedge C2 \\ 100, & \text{if } C1 \vee C2 \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

The reward function is used only during the training period. In the operation phase, the actors interact with the environment without experiencing any reward. Agents only



(a) Centre of inertia speed.

(b) Total power.

(c) Secondary control.

(d) Cost of the generators.

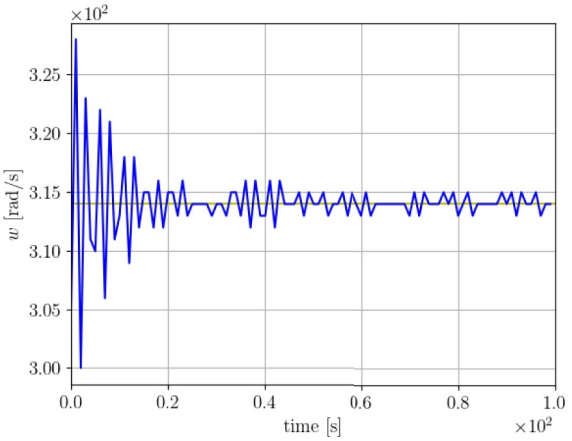
FIGURE 13 Change in load by 0.15 pu using the benchmark algorithm

observe the frequency of the system and their own control action z_i. They have learnt during training how to behave according to the evolution of the environment to balance supply and demand while minimising operational costs. For the operation phase, we change the load by 0.15 pu and then observe how the agents restore the system frequency.

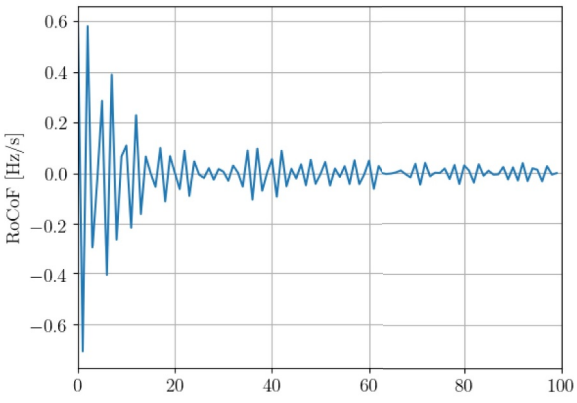
We can observe in Figure 12a the cumulative reward obtained by the agents. The agents can obtain 20,000 at maximum per episode, that is, the maximum reward per step is 200 and the number of steps per episode is 100. The agents learn how to obtain higher rewards as the number of episodes increases. If that were not the case, the cumulative reward function would oscillate around small values near zero.

In Figure 12b,d, we see how the agents restore the frequency to the nominal set point, thus balancing supply and demand. In Figure 12c, the rate of change in frequency (RoCoF) that measures the dynamic performance of the system is depicted. The maximum, minimum and mean RoCoF values are 0.607, −0.712 and 0.002 Hz/s, respectively, thus being within the admissible limits of 1 Hz/s recommended by ENTSO-E [46]. Actors learn how to balance generation and demand without exchanging information. The agents have learnt that keeping $\Delta\omega$ close to 0 is the key to obtaining high rewards. Thus, the agents are able to perform primary and secondary control in a totally distributed manner.

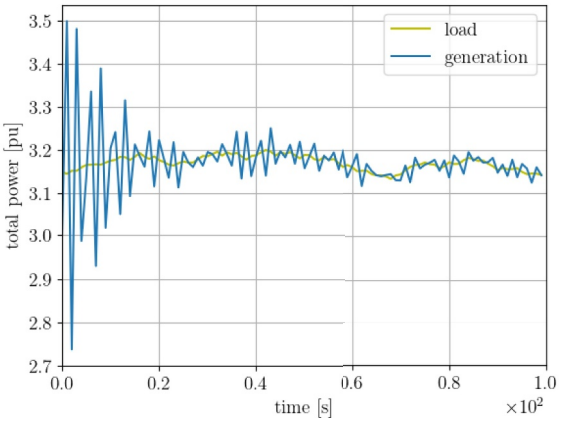
In order to test the optimality of the solution provided by the proposed approach in terms of cost, we need to calculate the optimal point when the load in the system is $P_L = 3 + \Delta P_L = 3.15$ pu for the cost functions given in this case study. By solving the economic dispatch problem as given in (12), we have $P_1 = 1.05$ pu and $P_2 = 2.10$ pu. In Figure 12e,f, the behaviour of each generator output and its associated cost are depicted. It can be observed that the agents operate near the optimal solution, namely that generator 2 generates twice as much as generator 1. As seen in Figure 12e, the control action of agent 1 stabilises around a set point that is approximately half of the control action of agent 2. This does not coincide with the optimal solution (slightly above half the production, i.e. 60%), but through the training process the agents learn how to keep load and supply balanced in a fully distributed cost-efficient way. The performance of the agents is determined by those actions they learn during training that lead to high rewards. Thus, the reward function is the main tool for showing each agent what the optimal action is. The reward



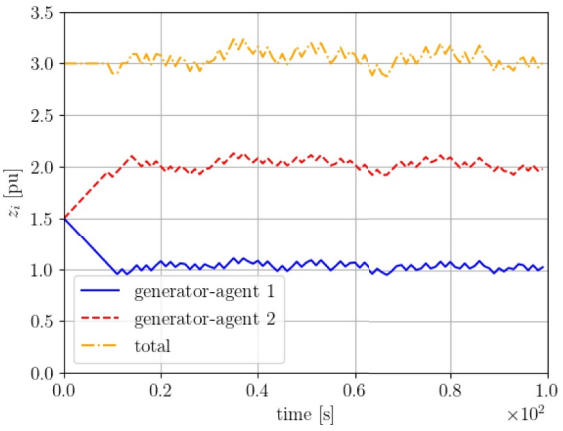
(a) Centre of inertia speed.



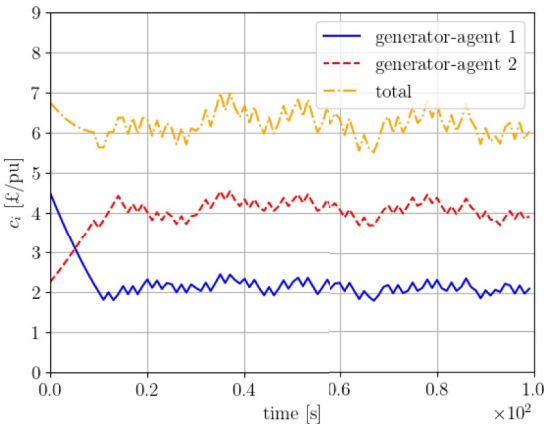
(b) RoCoF.



(c) Total power.



(d) Secondary control.



(e) Cost of the generators.

FIGURE 14 Tertiary control Model I: change in load by 0.15 pu followed by continuous changes in the load

function defined in (38) builds a reward combining two different dimensions: cost and frequency. This means that the reward function can show various maxima depending on the combination of both the reward dimensions. The agents learn by trial and error a behavioural heuristic to obtain high rewards, but they can converge to a local optimum that may be different from the global one. Indeed, a modification of the reward function (38) could improve the results.

We compare the proposed framework with [14], neglecting the network effects. In [14], a distributed LFC algorithm that restores system frequency in a cost-effective way is presented. This is achieved by exchanging some information between the

generators during the operation phase. The algorithm is based on a partial primal-dual gradient scheme to solve the optimal LFC problem, which is the standard in the literature. We refer to this algorithm as the benchmark algorithm. In Figure 13a,b, it can be seen that the benchmark algorithm manages to balance generation and demand, although it converges slightly slower than the proposed approach. In Figure 13c,d, the secondary control action and the cost of each agent are shown. The response of the benchmark algorithm is smoother than the proposed approach and the generation cost is minimised. However, this solution still needs to share dual information across units. On the other hand, although there are no optimality guarantees in the proposed framework, the results show that a sub-optimal solution is reached, and it is fully distributed, that is, no information is shared between the agents and they only use local information.

We also run a numerical experiment implementing a more realistic scenario, where an initial load increase of 0.15 pu is followed by a continuous change in the load sampled from a uniform distribution defined in the [−0.1, 0.1] interval. We can observe in Figure 14a,c that the agents manage to keep generation and demand balanced, although the load is continuously changing. The dynamic behaviour of the system is depicted in Figure 14b, where it can be seen that the RoCoF does not go beyond the admissible 1 Hz/s bound (maximum, minimum and mean RoCoF are 0.595, −0.708 and 0.002 Hz/s, respectively). Interestingly, it is shown in Figure 14d,e that the agents keep generating in a close-to-optimal ratio despite the continuous change in the load that increases the difficulty of the task.

In the numerical studies, we have shown that LFC may be performed efficiently in a distributed manner. More specifically, we demonstrated that instead of solving the economic dispatch to obtain the optimal operating point, the MARL framework can be used to infer the production costs while balancing demand and supply. The benefits of the proposed approach is that the agents can act in real time in a distributed way, restore the system frequency to the nominal value by satisfying the LFC performance criteria, and achieve a near optimal cost when doing so. Once trained, they do not need to centralise information at all. Dynamics that only use local information are embedded in the agents. We focussed on small-scale systems to demonstrate the performance of the proposed framework. In this way, we can provide insights into and physical interpretations of the presented results. In particular, we perform simulation studies for systems up to eight generators that participate in LFC. As such, the overall system that contains the generators could be up to some decades (see e.g. [29]). The conditions under which the system operates are realistic, since uncertainty in the net load is represented. For future work, we plan on implementing the proposed framework in large-scale systems (hundreds or thousands of nodes). Furthermore, modifications of the current framework will be investigated to further improve scalability. Some preliminary results for improving scalability concerns of such methods are given in [47, 48]. In [47], the authors propose a MARL framework with observation embedding, which is used to reduce computation through dimensionality reduction and parameters sharing. In [48], the authors exploit other agents' policies in a MARL framework in the training phase to reduce computational burden.

6 | CONCLUDING REMARKS

In this article, we proposed a MARL alternative to implement LFC in a distributed and cost-efficient way. To this end, we have expressed the LFC problem in an MARL setup and designed the reward functions based on insights on the economic dispatch problem. We have used MADDPG to implement this solution. Through numerical examples, we have shown that the proposed framework performs LFC in a satisfactory way. In particular, we demonstrate that all levels of control are achieved using Model I, that is, frequency is restored to the nominal value in a cost-efficient way and that secondary control is performed under the detailed modelling of Model II. Moreover, we have shown that the proposed methodology can cope with generation rate constraints and uncertain sources efficiently.

There are natural extensions of the work presented here. For instance, different elements of the MARL paradigm can be enhanced, that is, the reward function, the LSTM architecture and the introduction of domain knowledge could be further analysed to come up with agents that are able to improve their performance. More specifically, other architectures such as gated recurrent units could be used instead of an LSTM. An exhaustive search for the appropriate architecture, parameters and hyperparameters is necessary. Another obvious extension consists of adding the tertiary control layer to the network model. In the future, we also plan on studying the applicability and scalability of these techniques in more complex scenarios. In addition, we will investigate the performance of MADDPG when dealing with different types of generation resources and a large-scale power system.

ORCID

Sergio Rozada https://orcid.org/0000-0003-1042-7502

REFERENCES

1. Singh, A., Surjan, B.S.: Microgrid: A review. Int. J. Renew. Energy Technol. 3(2), 185–198 (2014)
2. Wang, X., et al.: A review of power electronics based microgrids. J. Power Electron. 12(1), 181–192 (2012)
3. Cady, S.T., et al.: A distributed frequency regulation architecture for islanded inertialess ac microgrids. IEEE Trans. Contr. Syst. Technol. 25(6), 1961–1977 (2017)
4. Apostolopoulou, D., Sauer, P.W., Domínguez-García, A.D.: Distributed optimal load frequency control and balancing authority area coordination. In: 2015 North American Power Symposium (NAPS), pp. 1–5. (2015)
5. Apostolopoulou, D., Sauer, P.W., Domínguez-García, A.D.: Balancing authority area coordination with limited exchange of information. In: 2015 IEEE Power & Energy Society General Meeting, pp. 1–5. IEEE (2015)
6. Guerrero, J.M., et al.: Hierarchical control of droop-controlled ac and dc microgrids—a general approach toward standardization. IEEE Trans. Ind. Electron. 58(1), 158–172 (2011)
7. Shafiee, Q., Guerrero, J.M., Vasquez, J.C.: Distributed secondary control for islanded microgrids – a novel approach. IEEE Trans. Power Electron. 29(2), 1018–1031 (2014)

8. Simpson-Porco, J.W., et al.: Secondary frequency and voltage control of islanded microgrids via distributed averaging. IEEE Trans. Ind. Electron. 62(11), 7025–7038 (2015)
9. Dörfler, F., Simpson-Porco, J.W., Bullo, F.: Breaking the hierarchy: distributed control and economic optimality in microgrids. IEEE Trans. Contr. Netw. Syst. 3(3), 241–253 (2016)
10. Latif, A., et al.: Comparative performance evaluation of WCA-optimised non-integer controller employed with WPG–DSPG–PHEV based isolated two-area interconnected microgrid system. IET Renew. Power Gener. 13(5), 725–736 (2019)
11. El-Hameed, M.A., El-Fergany, A.A.: Water cycle algorithm-based load frequency controller for interconnected power systems comprising non-linearity. IET Gener. Transm. Distrib. 10(15), 3950–3961 (2016)
12. Latif, A., et al.: Illustration of demand response supported co-ordinated system performance evaluation of YSGA optimized dual stage PIFOD-(1+ PI) controller employed with wind-tidal-biodiesel based independent two-area interconnected microgrid system. IET Renew. Power Gener. 14(6), 1074–1086 (2020)
13. Latif, A., et al.: Optimum synthesis of a BOA optimized novel dual-stage PI-(1+ ID) controller for frequency response of a microgrid. Energies. 13(13), 3446 (2020)
14. Li, N., Zhao, C., Chen, L.: Connecting automatic generation control and economic dispatch from an optimization view. IEEE Trans. Contr. Netw. Syst. 3(3), 254–264 (2016)
15. Yang, T., et al.: Minimum-time consensus-based approach for power system applications. IEEE Trans. Ind. Electron. 63(2), 1318–1328 (2015)
16. Trip, S., et al.: Passivity-based design of sliding modes for optimal load frequency control. IEEE Trans. Contr. Syst. Technol. 27(5), 1893–1906 (2018)
17. Moayedi, S., Davoudi, A.: Distributed tertiary control of dc microgrid clusters. IEEE Trans. Power Electron. 31(2), 1717–1733 (2016)
18. Daneshfar, F., Bevrani, H.: Load–frequency control: a GA-based multi-agent reinforcement learning. IET Gener. Transm. Distrib. 4(1), 13–26 (2010)
19. Eftekharnejad, S., Feliachi, A.: Stability enhancement through reinforcement learning: load frequency control case study. In: 2007 iREP Symposium-Bulk Power System Dynamics and Control-VII. Revitalizing Operational Reliability, pp. 1–8. IEEE (2007)
20. Rozada, S., Apostolopoulou, D., Alonso, E.: Load frequency control: a deep multi-agent reinforcement learning approach. In: 2020 IEEE Power & Energy Society General Meeting, pp. 1–5. IEEE (2020)
21. Yan, Z., Xu, Y.: Data-driven load frequency control for stochastic power systems: a deep reinforcement learning method with continuous action search. IEEE Trans. Power Syst. 34(2), 1653–1656 (2018)
22. Sutton, R.S., Barto, A.G.: Reinforcement learning: an introduction. IEEE Trans. Neural Netw. 16, 285–286 (1988)
23. Wang, S., et al.: A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning. IEEE Trans. Power Syst. 35(6), 4644–4654 (2020)
24. Xu, X., et al.: A multi-agent reinforcement learning-based data-driven method for home energy management. IEEE Trans. Smart Grid. 11(4), 3201–3211 (2020)
25. Kamruzzaman, M., et al.: A deep reinforcement learning-based multi-agent framework to enhance power system resilience using shunt resources. IEEE Trans. Power Syst. 1 (2021)
26. Chen, X., et al.: Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision (2021)
27. Imthias Ahamed, T., Nagendra Rao, P., Sastry, P.: A reinforcement learning approach to automatic generation control. Electr. Power Syst. Res. 63(1), 9–26 (2002)
28. Cui, W., Zhang, B.: Reinforcement learning for optimal frequency control: a Lyapunov approach (2021)
29. Yan, Z., Xu, Y.: A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system. IEEE Trans. Power Syst. 35(6), 4599–4608 (2020)
30. Abouheaf, M.: Load frequency regulation for multi-area power system using integral reinforcement learning. IET Gener. Transm. Distrib. 13(12), 4311–4323 (2019)
31. Yin, L., et al.: Artificial emotional reinforcement learning for automatic generation control of large-scale interconnected power grids. IET Gener. Transm. Distrib. 11(9), 2305–2313 (2017)
32. Li, J., Yu, T.: Deep reinforcement learning based multi-objective integrated automatic generation control for multiple continuous power disturbances. IEEE Access. 8, 156 839–156 850 (2020)
33. Wood, A., Wollenberg, B.: Power Generation, Operation and Control. Wiley, New York (1996)
34. Sauer, P.W., Pai, M.A.: Power System Dynamics and Stability. Prentice Hall, Upper Saddle River, NJ (1998)
35. Glavitsch, H., Stoffel, J.: Automatic generation control. Int. J. Electr. Power Energy Syst. 2(1), 21–28 (1980)
36. Kirschen, D., Strbac, G.: Fundamentals of Power System Economics. Wiley (2004)
37. Apostolopoulou, D., Domínguez-García, A.D., Sauer, P.W.: An assessment of the impact of uncertainty on automatic generation control systems. IEEE Trans. Power Syst. 31(4), 2657–2665 (2016)
38. Nwana, H.S.: Software agents: an overview. Knowl. Eng. Rev. 11(3), 205–244 (1996)
39. Littman, M.L.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 157–163. Morgan Kaufmann (1994)
40. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. 8(3), 279–292 (1992)
41. Mnih, V., et al.: Playing ATARI with deep reinforcement learning. In NIPS Deep Learning Workshop (2013)
42. Bosoniu, L., Babuŝka, R., Schutter, B.D.: A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst. Man. Cybernet. C. 38(2), 156–172 (2008)
43. Lowe, R., et al.: Multi-agent actor-critic for mixed cooperative-competitive environments. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, Vol. 30, pp. 6379–6390. Curran Associates, Inc. (2017)
44. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997)
45. Lample, G., Chaplot, D.S.: Playing FPS games with deep reinforcement learning (2017)
46. Frequency Measurement Requirements and Usage, Final Version 7. RG-CE System Protection & Dynamics Sub Group, ENTSO-E, Brussels, Belgium (2018)
47. Zhang, J., et al.: Scalable deep multi-agent reinforcement learning via observation embedding and parameter noise. IEEE Access. 7, 54615–54622 (2019)
48. Mao, H., et al.: Modelling the dynamic joint policy of teammates with attention multi-agent DDPG. arXiv preprint arXiv:1811.07029 (2018)

How to cite this article: Rozada, S., Apostolopoulou, D., Alonso, E.: Deep multi-agent Reinforcement Learning for cost-efficient distributed load frequency control. IET Energy Syst. Integr. 3(3), 327–343 (2021). https://doi.org/10.1049/esi2.12030