Adaptive neural control of PEMFC system based on data-driven and reinforcement learning approaches

Christophe Lin-Kwong-Chon, Cédric Damour, Michel Benne, Jean-Jacques Amangoua Kadjo, Brigitte Grondin-Pérez

Adaptive neural control of PEMFC system based on data-driven and reinforcement learning approaches

Christophe Lin-Kwong-Chon*, Cédric Damour, Michel Benne, Jean-Jacques Amangoua Kadjo, Brigitte Grondin-Pérez

LE2P-EnergyLab, EA 4079, University of Réunion Island, 15 Av. René Cassin, BP 7151, Saint-Denis 97715, Réunion, France

Abstract

Proton exchange membrane fuel cell systems are being increasingly put forward as hydrogen energy carrier converters. Recent advancements in reliability strategies have been stimulated through maintaining a healthy operating condition of the system and covering plant faults. However, it is observed that occurence or even the mitigation of these faults cause multilateral effects that can potentially destabilize the normal operation of the system. In the active fault tolerant control strategy, two modules are designed to fault management. The diagnostic module identifies the apparent fault and identifies the corrective commands, then the re-design module adapts the controller to dynamic system changes. In order to improve the generic characteristics of the re-design module, this paper presents a data-driven neural controller capable to automatically adapt to system health states. The developed approach comes from the machine learning class and combines adaptive dynamic programming, deep echo-state neural network models and fuzzy logic learning. The proposed controller is evaluated under occurence of channels flooding and membrane drying faults, but also actuators and water purging disturbances. Simulation and experimental results show the effectiveness of the proposed data-driven approach without prior neural model training, while guaranteeing the stability and learning convergence of the adaptive controller.

Keywords: Proton exchange membrane fuel cell, Active fault tolerance, Adaptive dynamic programming, Reinforcement learning, Deep echo state network

Contents

1. Introduction

To fulfil climate commitments and limit the rise of global temperatures by 2050, carbon-free hydrogen could help to reduce fossil-fuel reliance. Proton Exchange Membrane Fuel Cell (PEMFC) system is one electrochemical converter of the hydrogen energy carrier. This system has high potential due to its multiple advantages, such as low greenhouse gases emission, high efficiency, low operating temperature and rapid start. However, the PEMFC system has several technological and societal barriers. Indeed, there is a general apprehension about risks of hydrogen, the source of decarbonized hydrogen can be challenging, the cost of production and operation remains high, and the durability and reliability of these systems need to be improved. In fact, for reliability, the system may be subject to the occurrence of operating faults, which are symptomatic of performance degradations.

A plant fault is characterised by an undesired deviation of at least one physical quantity representative of the system from normal or acceptable operating conditions [1]. If the fault cumulates factors, such as a severe degree, a lack of identification or a failure to correct in time, it can cause irreversible damages to the PEMFC components in which case a degradation will occur, or even premature failure. The most common works around faults include

*Corresponding author.
Email address: christophe.lin-kwong-chon@univ-reunion.fr (Christophe Lin-Kwong-Chon)

reactant starvation, cathodic flooding and membrane drying. The main cause of the flooding fault is a low gas-flow rate in the channels [2] but could have impacts on the gas supply, voltage decrease or instability and cathodic differential pressure increase. Another level, partial flooding does not significantly affect the PEMFC performance, but will impact current density distribution, leading to degrade PEMFC durability [3]. A significant reduction of gas supply leads among more than 15 causes to the reactant starvation fault, such as low gas distribution, impurity gases or flooding [4]. This fault has the characteristic of provoking severe degradations, such as corrossion of carbon supports, reverse cell phenomenon and irregular distribution of current density. Membrane dehydration occurs when the production or presence of water at the cathode becomes insufficient in proportion to electro-osmosis drag from the anode. High operating temperature [5] or dry gas feeding also plays an essential role in the performance loss, as well as the gas flow velocity convecting vapor water [6]. The consequences of such a fault are voltage loss, cathodic pressure drop or even physical rupture of the membrane. Properly implemented, a fault management can keep optimum operating level, provide a fault tolerance and minimize downtime or temporary performance loss.

A Fault Tolerant strategy (FTC) is one of these closed-loop strategies that can tolerate a certain number of faulty operations while guaranteeing the system's expected performance and stability [7]. There are actually two main approaches [8]. The Passive FTC employs robust, predefined controllers to compensate a known *a priori* list of potential malfunctions without modifying the controller parameters. And, the Active FTC approach uses the reconfiguration of the controller according to the system state. In AFTC strategy, two main parts are designed : the diagnosis module identifies the occurring fault, localisation and magnitude, and these informations are used by the second one. This second module objective is to satisfy the closed-loop system's requirements despite the plant's faulty behaviour [1]. Commonly used, the diagnosis or a stand-alone module named decision, can provide fault recovery or mitigation command. Several plant fault tolerant strategies have already been developed, but have some limitations. An AFTC strategy is experimentally validated in [9] to flooding fault mitigation via oxygen stoichiometry. It is designed with a neural-based diagnosis, a self-tuning model-based Proportional-Integral-Derivative (PID) controller and a pre-defined decision mechanism. In [10], PTFC stoichiometry tracking is achieved based on the previous design. This approach's main limitation is the requirement to offline training of the neural models used for diagnosis and controller re-design. A simulated PFTC is proposed in [11] with electrodes flooding and membrane drying faults. Depending on the diagnosis output, a pre-definied control law from a bank of controllers is implemented in the closed-loop to maintain the voltage setpoint and zero pressure difference at the membrane. This process is fast and robust in a real-time application. However, the bank of controllers takes into account only the faults situations that were initialy considered. Others FTC are developed in [12, 13, 14, 15] for actuators or sensors faults. The developed controllers are mainly static approaches, such as linearization feedback, feedforward law or sliding mode. Regarding both modules, diagnostic methods are strongly studied in literature [16, 17, 18, 19, 20, 21, 22, 23], however, studies on decision [24] and controller re-design modules [25, 26] are lacking on the PEMFC system, which is not the case with other systems. In [27] an adaptive radial neural event-trigger control is developed for electromagnetic active suspension system. An other adaptive neural event-trigger is investigated in [28] for nonstrict feedback nonlinear systems with presence of nonaffine nonlinear faults.

The adaptive controller automatically compensates for all dynamic variations in the system. It considers any degradation of regulation over time, whether it is related to a change in operating point, a fault or even an external disturbance [29]. However, it requires system expertise and a large amount of data [30]. For that purpose, data-driven modelling through machine learning and, specially, its sub-discipline Reinforcement Learning (RL), offer powerful tools for on-line identification and control of complex time-varying systems [31, 32] without prior knowledge. Reinforcement learning denotes an optimal solving class-method in an unknown and uncertain environment based on the action-reward principle. In [33], Bellman extended Hamilton and Jacobi's works, establishing an optimal sequential decision-making algorithm, commonly referred to as dynamic programming [34]. Recently, Artificial Neural Networks (NN) have been highlighted into Adaptive Dynamic Programming (ADP) for their estimation capacities [35, 36]. In [37] ADP is combined with a Deep NN for self-learning the policy to drive the surge speed and yaw dynamics of a unmanned surface vehicle, simulation and real-world experimental results validate the proposed approach. An other ADP with four Deep Differential NN is developed in [38] for micro-grid control. A rejection process is designed to mitigate the inaccurate actions with the low probability distribution of the neural models. Simulation results show higher economically control performance with the proposed approach compared to 25 conventional control algorithms. A full-scale engine validation have been conducted in [39] with a model-free RL approach and implementation of an auxiliary tracking trajectory. In [40] an event-triggered ADP method allows to reduce computationnal effort for unknow nonaffine continuous-time systems with input constraints.

The main objective of this paper is to implement a neural controller-based exclusively on data-driven, which provides generic and adaptive properties regarding system health states to integrate within an AFTC strategy. This study focuses on the reconfigurable controller module of the fault strategy. The proposed adaptive neural controller includes three main features : an ADP structure, an deep Echo State Network (deepESN) neural model and

a Fuzzy-Logic (FL) approach. All these elements are carefully designed to satisfy the temporal context of fault dynamics and real-time application. The dynamic programming structure provides the ability to system anticipation control based on future states' prediction [41]. Compared to other recurrent neural networks, deepESN model is fast learning, resulting in low training cost [42]. Its training is performed only on the output layer and the model does not suffer from local minima if a least squares learning method is implemented [36]. Fuzzy logic is applied to update online the learning rate in order to improve the learning effectiveness and speed.

The main contributions of this paper are summarized as follows:

1. Channels flooding and membrane drying faults are considered throughout the design of the controller.
2. Compared to traditional PID and HDP controllers, results demonstrate the higher precision of the proposed method in a multiple-input multiple-output application, especially in faulty situations.
3. Experimental results effectively illustrate the performance of the proposed adaptive control strategy on real-time implementation.
4. The fuzzy logic method, coupled with the deepESN models allow quick and efficient weight adjustment in the online application.
5. The stability of deepADP is guaranteed by the Lyapunov method by proving the bounded domain of validity.

The rest of the paper is organized as follows. Section 2 describes the adaptive deepADP controller features: the ADP control structure, the deepESN model, optimization, and finally, control stability and learning convergence. In section 3, the simulation model and scenarios are presented, then the results are analyzed. Section 4 gives the experimental set-up, tests scenarios and on-line results. Finally, a conclusion draws the various advances obtained. As well, perspectives are given in order to open up new control proposals.

2. Adaptive neural controller

2.1. Dynamic programming approach

Consider a non-linear constant delay system with the following dynamics [43] as follows:

$$x_{k+1} = f\left(x(k), x(k-\theta_x), u(k), u(k-\theta_u)\right),\ 0 \geq k \geq \theta_{x,u} \tag{1}$$

where $x(k) = \left[x_1(k), x_2(k), ..., x_n(k)\right]^N \in \Re^n$ is a state vector, $\theta_{x,u}$ denotes unknown pure delays on system state and application of the command. The function representing the dynamics of the system $f\left(x_k, x_{\theta_x}, u_k, u_{\theta_u}\right)$ is defined on a group set $\Omega_x \subseteq \Re^n$ with $f(0,0,0,0) = 0$ from an initial state $x(0) = 0$. Let $\bar{x}_i$ denotes the sequence of state $x(k), x(k-\theta_x)$ and $\bar{u}_i$ denotes the sequence of state $u(k), u(k-\theta_u)$. Delays are not considered as a time-variable, therefore as justified in [44] the system can be considered as delay-free. In any case, a time recurrent model defined online and in real time such as the deepESN model helps to estimate these unknown parameters.

A cumulative cost function is defined as follows:

$$J(k) = \sum_{j=0}^{N-1} U^j(\bar{x}_j(k), \bar{u}_j(k)) \tag{2}$$

The objective is to minimize this cost function and determine an optimal command $u^*(k)$ as follows:

$$\begin{aligned} J^*(k) &= \min_{u(k)} \left\{J(\bar{x}(k), \bar{u}(k))\right\} \\ u^*(k) &= \underset{u(k)}{\operatorname{argmin}} \left\{J(\bar{x}(k), \bar{u}(k))\right\} \end{aligned} \tag{3}$$

Considering Bellman's principle of optimality and an initial state condition $x(k)$, if a controller is optimal on a given space between k and N, then it is optimal on each step $k+1 > k$. However, the principle of optimality cannot be solved without the knowledge of all temporal states at step k. Therefore a solution consists of an optimal local resolution between each timestep with predicted values. In addition, in the case of systems with respect of these conditions: constant delay, $\partial f/\partial u \neq 0,\ 0 \leq k \leq N_\theta$ and assuming that the matrix of partial derivatives $\partial f_\alpha/\partial u^n,\ \alpha = 1, ..., N$ has a maximum rank, then the delayed system could be reduced to a nondelayed problem [44]. Hence, the cost function takes this form:

$$\begin{aligned} J(k) &= \sum_{j=0}^{N-1} U^j(x_j(k), u_j(k)) + \gamma^{j-1} \sum_{j=N}^{N} \hat{U}^j(\hat{x}_j(k), \hat{u}_j(k)) \\ J^*(k) &= \min_{u(k)} \left\{U(x(k), u(k)) + \gamma \hat{U}(\hat{x}(k+1), \hat{u}(k+1))\right\} \end{aligned} \tag{4}$$

where $0 < \gamma \leq 1$ is the discount factor that modulates the amplitude of the predicted values in the cost function. The optimal command can then be formulated as follows:

$$u^*(k) = \underset{u(k)}{\operatorname{argmin}} \left\{U(x(k), u(k)) + \gamma \hat{U}(\hat{x}(k+1), \hat{u}(k+1))\right\} \tag{5}$$

To determinate this solution the prediction of future values is carried out by deepESN-type neural models. Specifically, an actor deepESN provides the optimal command, a predict deepESN provides the system response to this new command, and finally, a critic deepESN evaluates the evolution of the new state.

2.2. Global controller structure

The overall controller diagram is presented in Fig.1. The input signal of the predicted model $d(k)$ refers to all

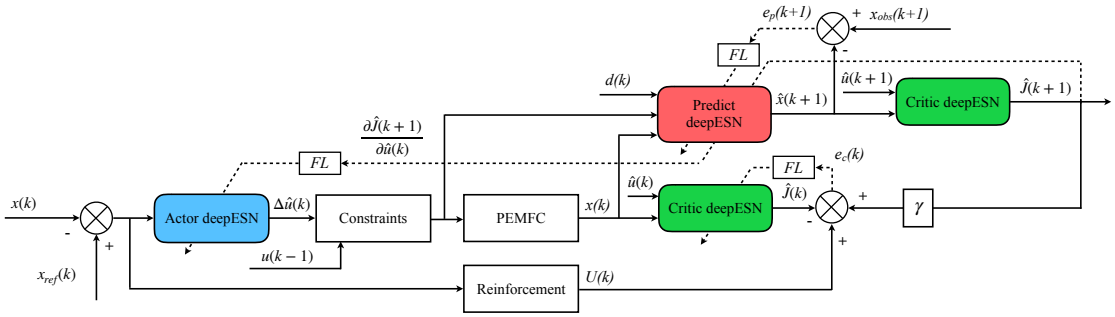


Figure 1. Adaptive control structure by dynamic programming

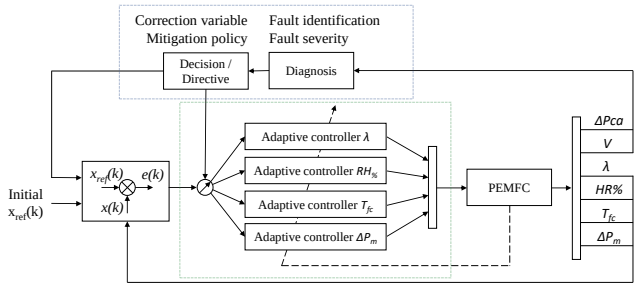


Figure 2. Position of adaptive controllers in the AFTC strategy

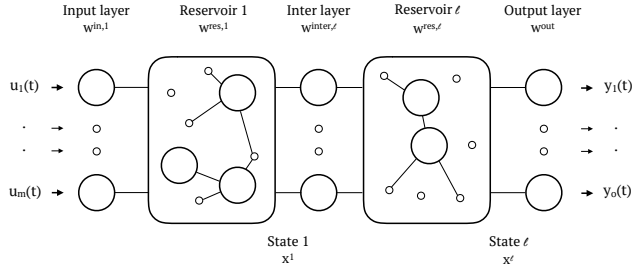


Figure 3. DeepESN-type neural network with l reservoirs

measurable perturbations allowing the model to predict better. Safety constraints are considered according to the operating ranges of the actuators.

In the AFTC design strategy, the diagnostic output is used only by the decision module, determining the variable to be modified, and the corrected setpoint to be applied. Illustrated in Fig.2, the different adaptive control loops modify their parameters according to the system's dynamic changes. The diagnosis and decision modules are not the main focus of this study and are directly simulated by voluntary setting fault conditions and predefined corrections.

2.3. Reservoir computing deepESN neural model

2.3.1. Deep ESN model

The classic ESN model is built on the basis of 3 basic layers: a fixed sparse reservoir of recurrent neurons as a hidden layer, a fixed random input layer, and finally, an output layer initialized to zero which is adjusted all along the adaptive control process. When the neural model is composed of several pairs of input layer-reservoirs, the structure then becomes a deep-ESN [45, 46]. However, learning cannot be considered as deep, since it only influences the output layer. Each reservoir has the same number of neurons and the output of each reservoir acts as input for the next layer. A unit biais is added as an input in order to shift the input-layer activation functions.

Illustrated in Fig.3, the output of the first reservoir is given as follows :

$$\begin{aligned} \tilde{x}^1(k) =& f\left(W^{in}u(k) + W^{res,1}x^1(k-1) + W^{back}y(k-1)\right) \\ x^1(k) =& (1-\alpha^1)x^1(k-1) + \alpha^1\tilde{x}^1(k) \end{aligned} \tag{6}$$

The other reservoirs output are then given as follows :

$$\begin{aligned} \tilde{x}^l(k) =& f\left(W^{inter,l}x^{l-1}(k) + W^{res,l}x^l(k-1)\right) \\ x^l(k) =& (1-\alpha^l)x^l(k-1) + \alpha^l\tilde{x}^l(k) \end{aligned} \tag{7}$$

where $x(n) \in \Re^n$ denotes the state of the neurons in the reservoir, $\tilde{x}(n) \in \Re^n$ denotes its update, $W^{in} \in \Re^{n\times m}$ denotes the input weights matrix, $W^{inter} \in \Re^{n\times n}$ denotes the inter-layer connections, $W^{res} \in \Re^{n\times n}$ denotes the reservoir weights matrix, l denotes the deep reservoir layer and $0 < \alpha \leq 1$ denotes the leaking rate which is a memory term. Then, the overall output is calculated by:

$$y(k) = W^{out}\left(x^1(k), ..., x^l(k)\right) \tag{8}$$

where $W^{out} \in \Re^{l\times nl}$ denotes the output weight matrix.

One major limitation of deepESN model is their performance dependence on their hyper-parameters (HP). Moreover, the number of HP can be quite large. Furthermore, in the deepADP proposed method severals neural models are used. Thus, an offline optimization is prioritized throught assuming that estimated HPs will be valid online. Indeed, incorrect on-line change in the HP can have severe consequences on the control closed-loop, such as control deviation. For this purpose, a pre-study was achieved to validate optimal values. It is accomplished with a grid search

Table 1
Estimated values for HP of all deepESN models

HP	Name	Estimated value
$nDeep$	Number of reservoir	100
$nRes$	Number of neurons per res.	900 or [30x30]
Co	Connectivity	0.1
α	Leaking rate	0.5
$\rho(W^{res})$	Spectral radius	Verified for each res.
η	Learning rate	Adjusted

method under prediction application $\hat{y}(t+1)$ of PEMFC system variables, such as voltage, stoichiometry and pressure difference at the membrane. The inputs and outputs of the deepESN model are normalized between 0 and 1. The least-square-based training algorithm is performed for the output weights layer estimation, and a train-test method is implemented to evaluate the model precision. Hyperparameters do not all have the same influence on the modeling error. The learning rate contributes to a 90% reduction in modeling error on a mean absolute percentage error criterion. Then followed by the number of neurons, the leaking rate and the spectral radius. The spectral radius is an index of the echoïc property of the reservoir, *i.e.* its ability to reflect information until it vanishes. It is an important HP for the deepESN design, therefore, it will be verified in the following way [46] for multi-reservoirs ESN models:

$$max\ \rho\left(\left(1-\alpha^{l}\right)I_n+\alpha^{l}W^{res,l}\right)=max\ \rho^{l}<1 \tag{9}$$

The learning rate is the main coefficient for modeling process and a specific decisional tool is be implemented for its estimation. In the result, the estimated HP values are given in Table 1.

2.3.2. Online learning phase

All deepESN models are initialized at the beginning of the control process and are online trained from that time on.

Reinforcement signal. The reinforcement signal used is a quadratic error criterion defined as follows:

$$U(k)=\frac{1}{2}\left(x_{ref}(k)-x(k)\right)^2 \tag{10}$$

Predict deepESN model. The predict model has the function of anticipating the evolution of the command calculated by the actor model on the PEMFC system. The inputs and outputs of the predict model are given in Table 2.

The modeling error used in training of predict model is given as follows:

$$e_{predict}(k)=x(k)-\hat{x}(k-1) \tag{11}$$

Table 2
Inputs and outputs of Actor, Predict and Critic deepESN models

deepESN	Input	Output
Predict (k)	1	$\hat{x}(k+1)$
	$\hat{u}(k)$	
	$x(k)$	
	$d(k+1)$ (known)	
Critic (k)	1	$\hat{J}(k)$
	$x(k)$	
	$\hat{u}(k)=\hat{u}(k-1)+\Delta\hat{u}(k)$	
Critic $(k+1)$	1	$\hat{J}(k+1)$
	$\hat{x}(k+1)+e_{predict}(k)$	
	$\hat{u}(k+1)=\hat{u}(k)+\Delta\hat{u}(k+1)$	
Actor (k)	1	$\Delta\hat{u}(k)$
	$x_{ref}(k)-x(k)$	
Actor $(k+1)$	1	$\Delta\hat{u}(k+1)$
	$x_{ref}(k)-\left(\hat{x}(k+1)+e_{predict}(k)\right)$	

where x denotes the real PEMFC state and $\hat{x}$ denotes output deepESN model. The actual value is not available at time k, therefore the previous time is considered. The training of the predicted model will therefore be delayed.

Critic deepESN model. As the critic model can estimate the present cost function and predict the future cost function of the reinforcement signal, it is used twice. The actor and critic models that are used for prediction $k+1$ include as input a state compensation $e_{predict}(k)$ from predict model. This modification is introduced to compensate the training delay of the predict model in the controller algorithm. The inputs and outputs of the critic model are given in Table 2.

Forward-in-time approach is adopted for the adjustment of the critic model and is defined as follows:

$$\begin{aligned}&\hat{J}(x(k))-U(k)-\gamma\hat{J}(x(k+1))=0\\&e_{critic}(k)=U(k)+\gamma\hat{J}(x(k+1))-\hat{J}(x(k+1))\end{aligned} \tag{12}$$

Actor deepESN model. The actor model has the role of providing the evolution of the command for the control process. This model is used twice, as the critical model. The inputs and outputs of the actor model are given in Table 2.

The command at time k is determined as follows:

$$u(k)=u(k-1)-\zeta\frac{\partial J(k)}{\partial u(k)} \tag{13}$$

where $0\leq\zeta\leq1$ is the control step length. During this study, the HP ζ and γ are set to a value of 0.3. The corresponding modeling error takes this form:

$$e_a(k)=-\zeta\frac{\partial J(k)}{\partial u(k)}-\Delta\hat{u}(k+1) \tag{14}$$

The jacobian information is obtained by the following means:

$$\frac{\partial J(k)}{\partial u(k)}=\frac{\partial U(k)}{\partial u(k)}+\gamma\frac{\partial\hat{J}(k+1)}{\partial u(k)} \tag{15}$$

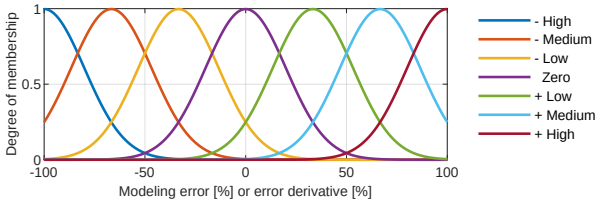


Figure 4. Membership functions related to the modeling error and its variation for the fuzzification process

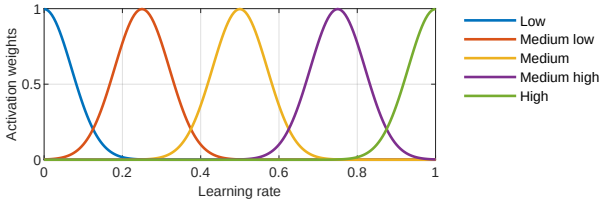


Figure 5. Membership functions related to the learning rate for defuzzification process

The future Jacobian information is obtained on the chain derivation rule as follows:

$$\frac{\partial \hat{J}(k+1)}{\partial u(k)} = \frac{\partial \hat{J}(k+1)}{\partial \hat{x}(k+1)} \frac{\partial \hat{x}(k+1)}{\partial u(k)} \tag{16}$$

And it is solved by the derivative of the activation function:

$$\begin{aligned} \frac{\partial J(k+1)}{\partial x(k+1)} &= \frac{\partial \hat{y}_{critic}(k+1)}{\partial \hat{x}(k+1)} \\ \frac{\partial J(k+1)}{\partial x(k+1)} &= \frac{\partial}{\partial x(k+1)} \left[W_{critic}^{out} \left(x^1(k), ..., x^l(k) \right) \right] \end{aligned} \tag{17}$$

$$\frac{\partial x(k+1)}{\partial u(k)} = \frac{\partial \hat{y}_{predict}(k+1)}{\partial u(k)} \tag{18}$$

2.4. Fuzzy logic for learning rate

The learning rate quantifies the gain in neuronal weight adjustment. At each timestep k the weight adjustment is obtained according to the modeling error subject to a learning rate. An important learning rate brings fast and robust adjustment. Conversely, a low learning rate ensures no overshooting but brings slow convergence. A fuzzy logic method is a decision-making tool that aims to formalize gradual processes based on real rules combinations. Fuzzy logic entries are modeling errors $e_{A,P,M}$ and errors derivative $\Delta e_{A,P,M}$. The only output of the fuzzy logic method is the learning rate $\eta_{A,P,M}$. Modeling errors and error derivative are constraints between [-100,100] and learning rate is constraint between [0,1]. Errors criteria are Mean Percentage Error to preserve negative and positive values. As seen in Fig.4, the input fuzzy set's operating range is partitioned according to a total of 7 sigmoidal membership functions (- High, - Medium, - Low, Zero, + Low, + Medium, + High).

And as suggested by Fig.5, the operating range of the fuzzy output set is partitioned according to a total of 5 sigmoidal membership functions (Low D1, Medium low D2, Medium D3, Medium high D4, High D5).

Fuzzy rules are built and proposed based on of the expertise, they are given in Table 3. This table refers to a discrete time space filled in by user expertise. In order to obtain all background values, i.e. all values in the quasi-continuous framework, a compromise is made between the continuous and discontinuous approaches. The fuzzy space offers the possibility of formalizing the union between the expertise and the background framework.

Table 3
Action region of fuzzy rules and their overlaps

If e_m / **Then** η \ **And** Δe_m	- H	- M	- L	Zero	+ L	+ M	+ H
- H	D1		D3		D5		D4
- M		D2					D3
- L	D2						D2
Zero	D1	D2	D3	D3	D3	D2	D1
+ L	D2						D2
+ M	D3					D2	
+ H	D4		D5		D3		D1

To compute the final learning rate value, the combined output fuzzy set is defuzzified using the Mamdani method. Then, a fuzzy surface is obtained by running the entire operating area of the inputs. The resulting matrix, as shown in Fig.6, is directly used for on-line training of deepESN models.

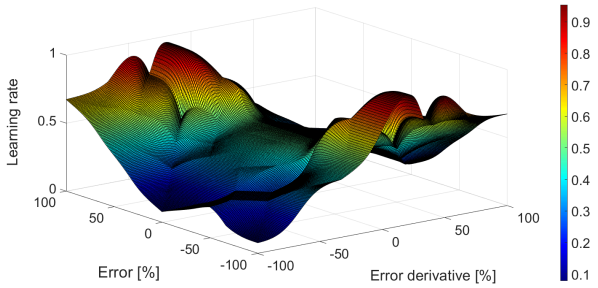


Figure 6. Fuzzy surface related to the learning rate

An example application is given in Fig.7 with a single learning target of 0.49V. The variable learning rate obtained by the fuzzy logic method makes it possible to reach this target quickly and stably over time.

2.5. Control stability and neural learning convergence

Two levels of functionality are implemented to ensure the correct operation of the controller:

- The higher level, ensures the deepADP validity domain through Lyapunov stability theorem. In case the regulation would be divergent, the system is shut down to avoid irreversible degradation of the PEMFC membranes.

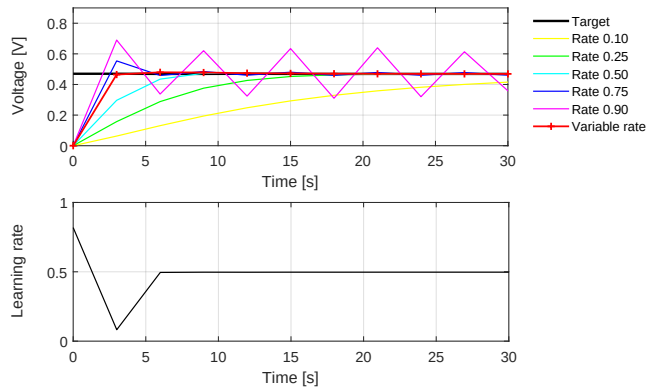


Figure 7. Application of the fuzzy logic method on learning rate

- The lower level, works internally and ensures a fast learning of each neural networks using bounded variable learning rate.

2.5.1. Control stability

Stability is the central concern in control and specifically in adaptive neural control. Lyapunov stability analyses are the most commonly adopted approach for this study. Huang et al. [47] recently analyzed the ADP controller's domains of stability according to the reinforcement function. For a control utility in a square-weighted sum form $U(k) = \sum_{j=1}^{N} \frac{1}{2} e_j^T(t) e_j(t)$, lyapunov's candidate function is defined as follows:

$$V(t) = \sum_{k=t+1}^{N} \gamma^{k-t-1} \left(U^T(k) U(k) R[X(k)] \right) \tag{19}$$

where N is an optimal control horizon. The derivative of the candidate function is built from the boundaries theorem around minimum and maximum values. These bounds are defined according to the minimum and maximum value of the matrix R(k). If the resolution of the derivative indicates a negative value, then that demonstrates the existence of an asymptotically stable deepADP control loop in the sense of Lyapunov:

$$\dot{V}_{max}(k) = -\sum_{j=1}^{n} \frac{1}{2} \left\| \frac{x(k+1) - x_{ref}(k+1)}{x_{max}(k+1)} \right\|^2 \tag{20}$$
$$\dot{V}(k) \leq \dot{V}_{max}(k) < 0$$

2.5.2. deepESN learning convergence

For a learning algorithm using the recursive least-squares method, with W^{out^*} the ideal weights of the output matrix, the output of the deepESN model at step k is then given by:

$$y_{ref}(k) = x^T(k) W^{out^*}(k) \tag{21}$$

The error between the ideal weights and the weights is given as follows:

$$\tilde{W}^{out}(k) = W^{out^*}(k) - W^{out}(k) \tag{22}$$

From a boundary analysis performed by Bo & Zhang [36] on ESN models, it has been shown that in this configuration the weights $W^{out}(k)$ trend towards the ideal weights $W^{out^*}(k)$ in infinite time.

$$\lim_{k \to \inf} W^{out}(k) = W^* out(k) \tag{23}$$

Therefore, the adjustment of the output matrix is defined as below under the condition that the learning rate is bounded $0 < \eta \leq 1$:

$$W^{out}(k) = W^{out}(k-1) + \eta \frac{P(k) x(k) e(k)}{1 - x^T(k) P(k) x(k)} \tag{24}$$

where $P(k)$ is the covariance matrix of the estimation error and is defined as follows:

$$P(k) = [P^{-1}(k-1) + x^T(k) x(k)]^{-1}$$
$$P(k-1) - \frac{P(k-1) x(k) x^T(k) P(k-1)}{1 + x^T(k) P(k-1) x(k)} \tag{25}$$

where $P(k) \in \Re^{nxn}$ is initialized on a unit matrix.

The complete process of the deepADP controller is presented in Algorithm 1.

Algorithm 1: deepADP controller

Input: Control errors $e_{control}(k)$, Control references $e_{ref}(k)$
Output: Commands $\hat{u}(k)$, deepESN models
Data: deepESN models configuration
// Initialization
Set HP of actor, predict and critic deepESN models;
Initialization of all neural weigths;
Verification of the echoïc properties of the models;
Loading of the fuzzy surface;
Define user satisfaction survey $e_{ref}(k)$;
Collect PEMFC data and initial system state $x(k)$;
// Control start
while $e_{control}(k) > e_{ref}(k)$ **do**
  Estimation of the Predict learning rate $\eta_{predict}(k)$;
  Adjustment of the Predict model $e_{predict}(k) \rightarrow W^{out}_{predict}(k)$;
  Calculation of the reinforcement signal $U(k)$;
  Estimation of the control cost $\hat{J}(k)$;
  Estimation of the command $\hat{u}(k)$;
  Prediction the system's future state $\hat{x}(k+1)$;
  Verification of stability via the Lyapunov theorem;
  Prediction the future command $\hat{u}(k+1)$;
  Prediction the future control cost $\hat{J}(k+1)$;
  Estimation of the Critic learning rate $\eta_{critic}(k)$;
  Adjustment of the Critic model $e_{critic}(k) \rightarrow W^{out}_{critic}(k)$;
  Estimation of the Actor learning rate $\eta_{actor}(k)$;
  Adjustment of the Actor model $e_{actor}(k) \rightarrow W^{out}_{actor}(k)$;
  Saving deepESN models

3. Simulation results

The PEMFC system requires several auxiliaries to control different operating conditions. The flow of reactive

gases is the most critical variable to be regulated, it denotes the transport of reactants in the distribution channels, in the diffusion layer and finally towards the catalytic layer. The stoichiometry must be maintained for PEMFC operation regardless of changes in the current load. However, the gas flow will also influence the evacuation of water [48]. Membrane humidification is essential for proton mobility and electric performance, in short, for the membrane durability. However, an excess can lead to water plugging of channels reducing the circulation of reactive gases. On the other hand, a humidity deficit is risky for the membrane's integrity, which then becomes more sensitive to high temperature and pressure [49]. On top of that, temperature influences the dew temperature, which can have consequences on the relative humidity. Thus, heating or cooling subsystems are essential auxiliaries to guarantee optimal electrical production continuity without any structural degradation [50]. The pressure within the distribution channels influences the cell's performance, which modifies the kinetics of the reactions [51]. However, the pressure difference between the anode and cathode must be relatively small ($< 30kPa$ [52]) or even zero to maintain the physical integrity of the membrane. While, events such as water accumulation, water purge or changes in operating conditions can lead to pressure disturbances [53]. For these reasons, the variables of stoichiometric control, pressure difference at the membrane, and temperature are implemented in simulation. Humidity will be influenced by the dew temperature

Simulation PEMFC model and experimental system configurations do not involve the same types of components, in regard to cell size and control variables. Nonetheless, these two different configurations allows the validation of the adaptive controller generic property.

3.1. PEMFC simulation model

For simulation, the fuel cell core is modelled based on the work of Pukrushpan [54]. This stack model of 381 cells is later implemented in the study of Wu & Zhou [11] which emulates the phenomenology of the cathode flooding and membrane drying faults. The configuration of the stack model stays the same as the original design. The controlled variables are the oxygen stoichiometry, the pressure difference at the membrane and the fuel cell temperature.

$$[y_1; y_2; y_3] = [\lambda_{O_2}; \Delta P_{mbr}; T_{fc}] \quad (26)$$

The commands variables are the air inlet pressure, the hydrogen inlet pressure and the cooling massflow rate.

$$[u_1; u_2; u_3] = [p_{air,in}; p_{H_2,in}; w_{cooling,in}] \quad (27)$$

3.2. Simulation scenarios

Traditional PID and HDP controllers are used during the simulation study in order to make a performance comparison with the proposed deepADP controller. PID controllers are optimized with the Ziegler-Nichols heuristic method. The action and critic modules in the traditionnal HDP controller are carried out by fully connected feedforward neural network models. All controllers are evaluated according to the Root Mean Square Error criterion, in this way, the error can be quantified according to the controlled variable. The simulation of the PEMFC model is performed with a sampling time of 2 ms. The 3 PID controllers require 0.04 ms for the calculation of a one control loop. In comparison, the proposed controller used in multiple-input multiple-output (3-3) requires a calculation time of 0.5 ms, which corresponds to a ratio of x12 on an i5 desktop computer. Sample time of the control loop is defined at 5 ms for all controllers.

3.2.1. Simulation scenarios

The operating parameters of the PEMFC system for the simulation are detailed in Table 4. The simulation proceeds according to 3 scenarios : normal operating conditions (S1.1 & S1.2), flooding fault (S2) and membrane drying fault (S3).

Table 4
Operating conditions for the healthy mode and faults occurence simulation

Variables	Normal S1	Flooding S2	Drying S3
Current [A]	100-200	200	150
Anode input pressure [kPa]	150	150	150
Cathode input pressure [kPa]	200	200	200
H2 stoichiometry	2.5	2.5	2.5
O2 stoichiometry	3-5	3	2.5-4
Temperature [°C]	60	60	60-70
Ambient temperature [°C]	25	25	25
Anode humidity [%]	80	80	80
Cathode humidity [%]	80	100	50
Cathode injected water [kg/s]	0.005	0.005-0.015	0

Simulations S1.1 & S1.2: Normal operating conditions. S1.1 is a trajectory tracking of the oxygen stoichiometry which is modified with common values between 3 and 5. The load current is fixed to 150A. S1.2 is a regulation of the 3 control variables with current disturbance rejection between 150A, 175A and 200A.

Simulation S2: Flooding fault. The system load current is fixed at 200 A to enable the production of water. The flooding fault is simulated at the cathode by increasing the injected water mass flow at 4 seconds, the injected liquid water is located at the cathodic chamber and reduces the quantity of reagents for a fixed physical volume. The fault effect is a voltage loss and a pressure drop at the cathode.

Simulation S3: Membrane drying fault. The drying fault is simulated by a decrease in the cooling liquid mass flow rate. The temperature then increases by 60 to 70 °C in 15 seconds. The increase in temperature leads to evaporation of the water contained in the stack and the membrane. The stoichiometry is set to a value of 4 to contribute to

Table 5
Control errors in simulation for oxygen stoichiometry, difference pressure at membrane and fuel cell temperature

		RMSE λ_{O_2}	RMSE ΔP_m	RMSE T_{fc}
S1.1	deepADP	0.0477	**1 775.2 Pa**	0.4054 K
	HDP	0.0562	2 618.0 Pa	0.4079 K
	PID	**0.0436**	6258.9 Pa	**0.4053 K**
S1.2	deepADP	0.0236	**1 548.0 Pa**	**0.4056 K**
	HDP	0.0416	4 857.6 Pa	0.9214 K
	PID	**0.0206**	5 220.2 Pa	0.4090 K
S2	deepADP	0.0734	3 093.5 Pa	**0.0227 K**
	HDP	0.0948	**3 089.3 Pa**	0.0477 K
	PID	**0.0622**	12 585 Pa	0.0664 K
S3	deepADP	0.1105	**5 732.4 Pa**	4.4821 K
	HDP	0.2102	5 957.8 Pa	4.4659 K
	PID	**0.0925**	12 176 Pa	**4.4819 K**

the evacuation of the water produced by the electrochemical reaction. The AFTC strategy is simulated here by a decreased of the stoichiometry from 4 to 2.5 to attempt to mitigate the drying fault. The fault's consequences are a voltage loss and the fault mitigation causes a large change in pressure of 0.3 bar.

3.3. Simulation results

For all tests, the stability is verified in the sense of Lyapunov, the derivative of the function (19) remained strictly negative. Moreover, it is correctly limited by its maximum limit of stability $\dot{V}_{max}(t) - \dot{V}(t) \geq 1$. Therefore, the deepADP controller remained asymptotically stable. Also, each of the deepESN reservoirs has an echoïc nature and each target learning converges in a maximum of 10 iterations.

3.3.1. Simulation S1.1 - Stoichiometric setpoint tracking under normal operating conditions

PEMFC response under S1.1 is given in Fig.8. Control errors of all scenarios are given in Table 5. All controllers are satisfactory on the conducted test. For stoichiometry as for temperature, regulation is substantially equivalent for PID, HDP and deepADP controllers. For the control of the pressure difference, the deepADP controller has a smaller deviation than the others controllers, its anticipation capacity allows to strongly reduce the dynamic error. However, PID and HDP controllers reach the setpoint faster. The deepADP controller presented a better accuracy, but has a smoother and slower convergence to the setpoint in return.

3.3.2. Simulation S1.2 - Regulation with disturbance rejection under normal operating conditions

PEMFC response under S1.2 is given in Fig.9. For difference pressure regulation, deepADP controller performs better in the presence of this disturbance than PID and HDP controller. It offers half the deviation reduction than the PID controller. However, a static error is detected on the first steady state, then it is faded out after 5 seconds. The deepADP controller directly converges to the setpoint only after the second disturbance occurrence. Thus, it requires a certain amount of data before converging effectively, this is due to the deepESN models learning capacity.

3.3.3. Simulation S2 - Regulation in presence of flooding fault

PEMFC response under S2 is given in Fig.10. The PID controller allows a fast regulation but with high pressure variations, whereas the HDP and deepADP controller allow a slower regulation but with low pressure variations. Here the deepADP controller is more precise and effective in maintaining the balance of pressures to the membrane. The temperature control is also more efficient with the proposed controller. On the other hand, the PID controller is slightly more efficient for stoichiometry control.

3.3.4. Simulation S3 - Regulation in presence of membrane drying fault

PEMFC response under S3 is given in Fig. 11. The corrective action modifies the anodic input pressure, and consequently, the pressure difference at the membrane. The deepADP controller has higher accuracy than HDP and PID controllers. In this scenario the proposed controller provides a deviation four times smaller than the PID controller. The difference in performance between fixed and adaptive controllers are more significant in a fault correction situation.

A Kiviat diagram is presented in Fig.12 to synthesize the simulation results. Five criteria are presented: controller adaptation, prediction capacity, control performance (accuracy, response time, stability), immediate control ability and finally calculation time. The simulations show the deepADP controller relevance for the control of the PEMFC system, and the comparison with HDP and PID controllers reveal significant improvements, such as precision, prediction and adaptation.

1. First, deepADP and HDP approaches provide the best results in the presence of system faults compared to the PID controller. And precisely, the proposed controller gives the best regulation accuracies in these scenarios.
2. Secondly, on control variables less influenced by faults (stoichiometry and temperature control variables) the proposed controller has an equivalent accuracy with the optimized PID controller. The HDP controller presents larger position errors in these cases.
3. However, one limitation is highlighted with the proposed controller. The learning process requires several disturbances to eliminate static error completely, so that the deepADP controller allows direct convergence towards the setpoint. The HDP controller does not suffer from this limitation.

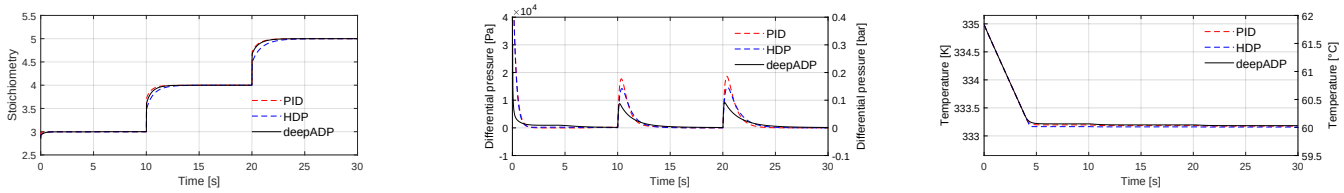


Figure 8. Simulation results S1.1 - PEMFC response under stoichiometric setpoint tracking (left) stoichiometric setpoint tracking, (middle) difference pressure setpoint tracking and (right) temperature setpoint tracking

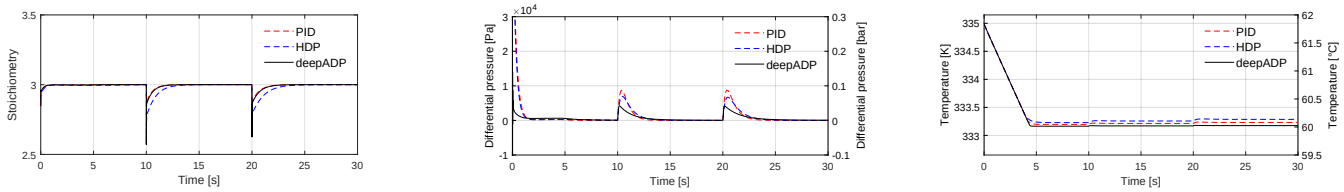


Figure 9. Simulation results S1.2 - PEMFC response in presence of current disturbances (left) stoichiometry setpoint tracking, (middle) difference pressure setpoint tracking and (right) temperature setpoint tracking

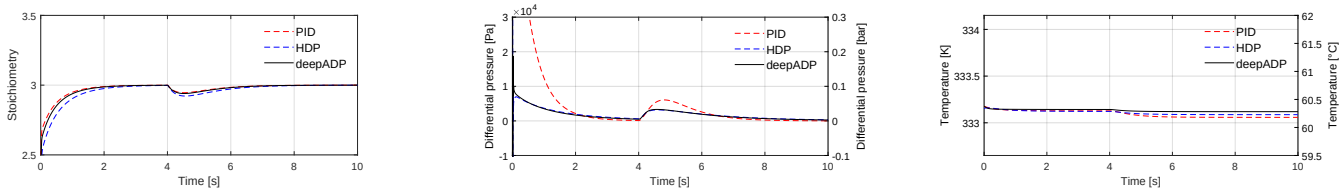


Figure 10. Simulation results S2 - PEMFC response under flooding fault (left) stoichiometry setpoint tracking, (middle) difference pressure setpoint tracking and (right) temperature setpoint tracking

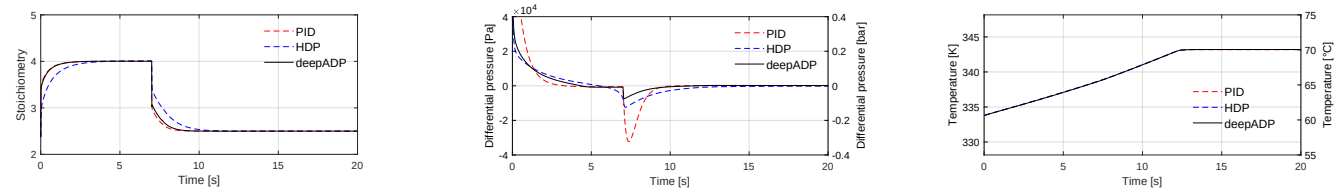


Figure 11. Simulation results S3 - PEMFC response under membrane drying fault (left) stoichiometry setpoint tracking, (middle) difference pressure setpoint tracking and (right) temperature setpoint tracking

Table 6
Main components of the test bench and the PEMFC system

Type	Reference	Details
MEA	1x N117	50 cm^2
Mass flow controller	2x MKS 1179A	Responsive time : 2 sec Accuracy : +/- 1 %
Temperature controller	Love controls 16A & 32DZ	
Back-pressure controller	2x 20812BP Fairchild	p_{max} = 17 bar Accuracy : 1 % at 7 bar
Pressure sensor	2x MSP-300100P4N1 2x TT7800-705	Accuracy : 1% Repetability : 0,1 % of p_{max}
Electronic charge	1x N3300A	

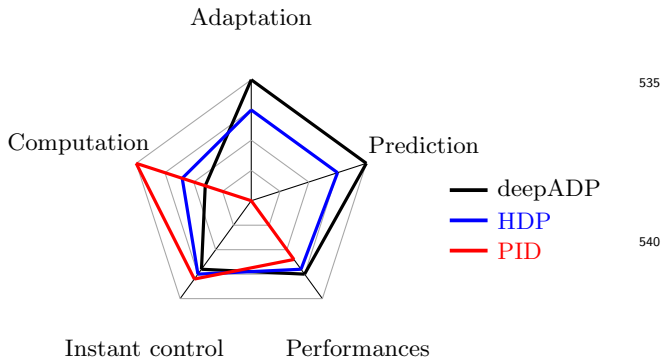


Figure 12. Comparison between deepADP, HDP and PID controllers in simulation with a normalized Kiviat diagram

4. Experimental results

4.1. PEMFC experimental setup

The test bench presented in Fig.13 includes all the auxiliary devices and actuators required for the fuel cell's operation, where a single cell is involved. Table 6 summarizes details of each main components. Due to the constraints of the experimental test bench, the minimum sampling time is 4 seconds. This sample time is very long compared to the simulation and considering the fluidic dynamics and implemented actuators. Nevertheless, the dynamic of the adaptive controller could be analyzed over a long sample time.

Figure 13. Laboratory test bench with its single cell

The test station automatically regulates the flow rate and temperature, and the response time of the massflow rate controller is shorter than the sampling time. Therefore, only the regulation of the pressure difference at the membrane is investigated in the experimental study. However, it is difficult to measure the PEMFC internal pressure. Thus, the experimental study considers the pressure difference at the inlet of the membrane, especially since this is where the risk of tearing is the highest. The controled variable is $y = \Delta P_{mbr,in}$ and the command variable is anode back-pressure set point $u = p_{an,out}$. The cathode back-pressure set point is given to the AFTC strategy. Back-pressure set points are then ensured by back-pressure actuators. This regulation indirectly affects the inlet valve regulated by the mass flow controller, which consequently modifies the inlet pressure and finally the pressure difference at the membrane inlet.

Non-linearities are observed in cathodic fluidic actuators. An illustration is given in Fig.14, at the beginning the pressure difference at the PEMFC inlet is zero. Then, equivalent reference modifications at the anodic and cathodic output back-pressure actuators are done. A nonlinear static error is observed on the back-pressure regulator when pressure rises. Fluctuations are also present at low pressures and the zero pressure difference is not maintained at the end of the test despite a return to the initial operating conditions.

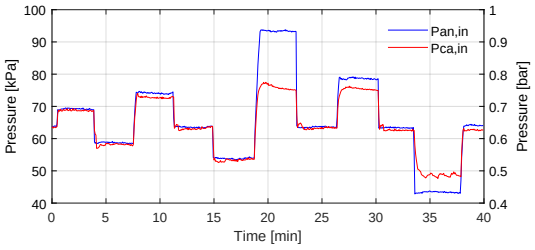


Figure 14. Experimentation - PEMFC response under identical back pressure reference changes at the anode and cathode

4.2. Experimental scenarios

4.2.1. Experimentation scenarios

The operating parameters of the PEMFC system for the experimentation are detailed in Table 7. The experimentation proceeds according to 3 scenarios: normal operating conditions (S1.1 & S1.2), flooding fault (S2) and membrane drying fault (S3).

Experimentation S1.1 & S1.2: Normal operating conditions. S1.1 is a difference pressure regulation with a rejection of a periodic disturbance, which is a repetitive step of $+5\ kPa/-5\ kPa$ on the cathodic back-pressure reference. The objective is to observe the learning process and the controller's potential for improvement when facing the repetition of a same event. S1.2 is a difference pressure regulation with stoichiometry disturbance rejection between

Table 7
Operating conditions in the healthy mode and faults occurence experimentation

Variables	Normal	S1 Flooding	S2 Drying S3
Current [A]	20	30	20
Hydrogen stoichiometry	2	2.5	2
Oxygen stoichiometry	2.5-10	3	10
Fuel cell temperature [°C]	70	55	80
Preheater temperature [°C]	70	55	80
Boiler temperature [°C]	62	70	63
Cathode humidity [%]	70	100+	50
Cathode out pressure ref. [kPa]	50	30	50

2 and 10, which are common values for a single cell under an AFTC strategy. The cathodic back-pressure is fixed.

Experimentation S2: Flooding fault. The flooding fault is caused by water condensation at the fuel cell inlet and its accumulation into the PEMFC. The fault is identified after 15 minutes by a voltage loss and a cathodic pressure drop. These effects reflects the accumulation of water and its evacuation at the cathode.

Experimentation S3: Membrane drying fault. The membrane drying fault is achieved by high oxygen stoichiometry and temperature rise from 60°C to 70°C. The fault is identified after 10 minutes by an increase then a voltage loss.

4.3. Experimentation results

4.3.1. Experimentation results S1.1 - Learning and adaptability of the deepADP controller

The system response is given in Fig.15. With the proposed approach, the regulation is much more efficient from the second occurrence of the disturbance. Moreover, the controller is more reactive after the first occurrence, the controller's response time is reduced by half upon the second occurrence. Compared to simulation, the deepADP controller does not suffer from static error at the first occurrence of a perturbation. This can be explained by the initial balanced state of the control. Two events are enough to improve the controller, and additional occurrences do not improve the control performances. In contrast, the optimized PID controller is equally efficient throughout this scenario.

4.3.2. Experimentation S1.2 - Regulation with stoichiometric disturbance rejection

PEMFC system response under oxygen stoichiomety disturbance is shown in Fig.16. DeepADP and PID controllers operate on the anodic output pressure according to the disturbance. The response times are equivalent for the two controllers. However, in terms of precision the PID controller presents some tremors that appear when the stoichiometry decreases, which is not the case with the proposed controller.

4.3.3. Experimentation S2 - Regulation under flooding fault operating conditions

PEMFC system response in the presence of flooding fault is shown in Fig.17. There are 2 types of disturbances highlighted in this test case, the internal disturbance by the noise of the pressure sensor and the external disturbance by the injection of condensed water. Fluctuations in the regulation are a direct consequence of the water accumulation and evacuation at the cathode. These external disturbances are perfectly taken into account by the deepADP controller, which ensures a zero balance at the inlet of the cell. The robustness of the developed controller to cope with fluctuations and noises is thus demonstrated here.

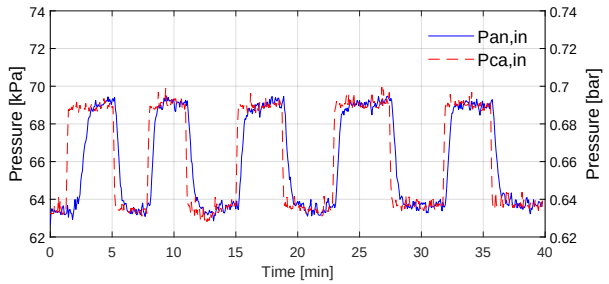
An unexpectedly back-pressure actuator valve partial-opening occurs at 11 minutes. This phenomena is possible during PEMFC system faults where the presence of liquid water is evacuated in a violent way by sudden openings of the outlet. The control system started to show larger errors between 11 and 20 minutes, then adapts itself after this period. The partial opening of the output pipe does not significantly interfere the deepADP controller and the regulation remains stable over time.

4.3.4. Experimentation S3 - Regulation under membrane drying fault operating conditions

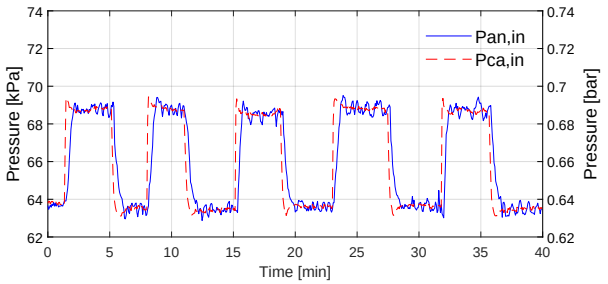
PEMFC system response in the presence of membrane drying fault is shown in Fig.18. The increase in temperature causes a temporary increase in voltage. Water accumulates which causes the cathode input pressure to rise and consequently the anode input pressure through the deepADP controller. Once the dewatering fault is set up the water content of the system decreases and the inlet pressures decrease. The disturbance rejection is performed by the deepADP controller.

The curve fracture observed in Fig.18c corresponds to an unexpectedly complete purge of the cathode channels. During this S3 test, four outlet openings (1 complete and 3 partials) are identified. These partial openings or purges are important disturbances for the regulation of the pressure difference. The results demonstrate a deepADP controller capable of adapting to these openings, with notably an elimination of overshoots from the third opening.

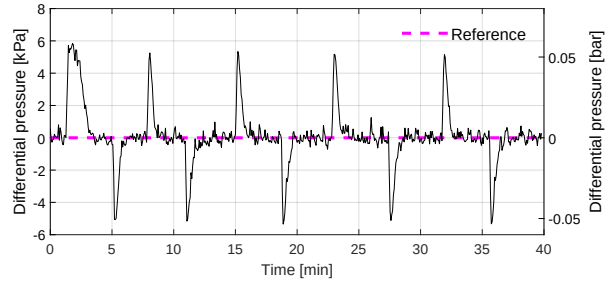
The proposed deepADP controller has been validated in experimentation. Some constraints due to the experimentation led to the different choice of control variables and commands from those used in the simulation. The proposed generic controller adapted to the experimental configuration without any change and similarly to the large sampling period of 4 seconds, but also to static internal controllers (back-pressure actuators) with nonlinear characteristics. The results in regulation showed very satisfactory performance in all the tests performed. However, it should be noted that during first perturbation occurence the reponse time of the proposed controller is larger compared to PID controller. An optimisation tool for ADP



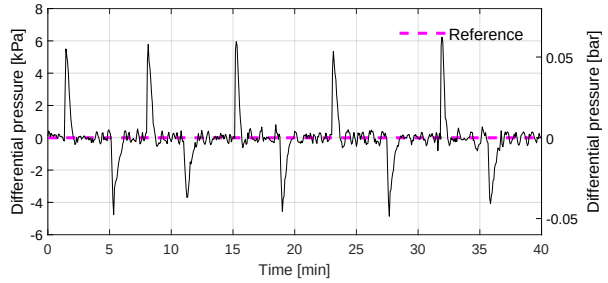
(a) deepADP controller - Anode and cathode inlet pressures



(b) PID controller - Anode and cathode inlet pressures

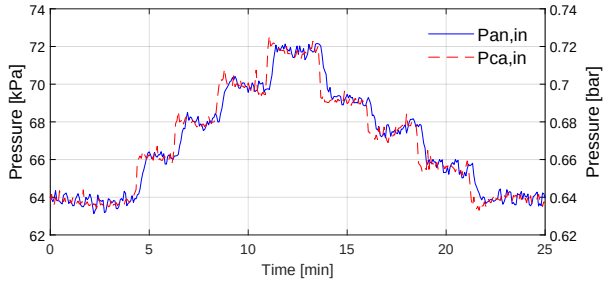


(c) deepADP controller - Difference pressure at the membrane inlet

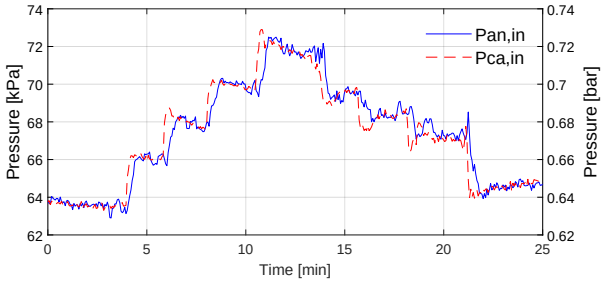


(d) PID controller - Difference pressure at the membrane inlet

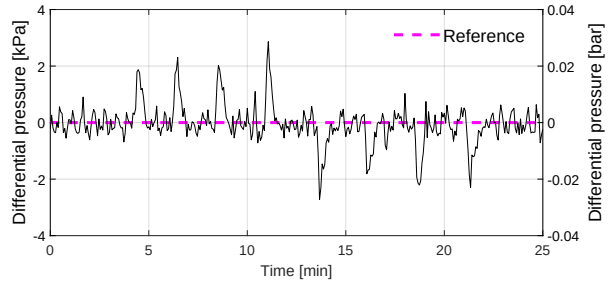
Figure 15. Experimentation results S1.1 - PEMFC response under repetitive disturbance and normal operating conditions (a, b) measured anode and cathode output pressures, (c, d) difference pressure at the membrane inlet



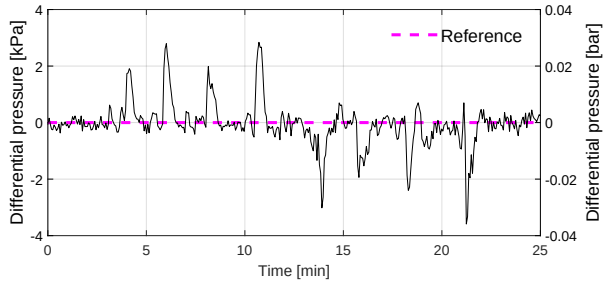
(a) deepADP controller - Anode and cathode inlet pressures



(b) PID controller - Anode and cathode inlet pressures

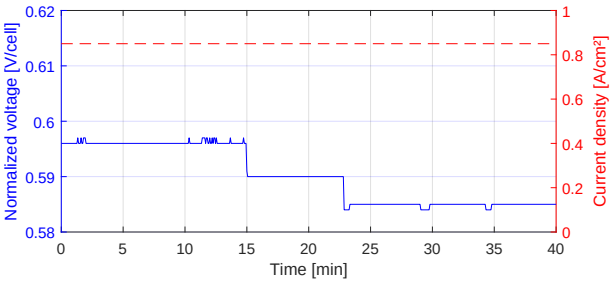


(c) deepADP controller - Difference pressure at the membrane inlet

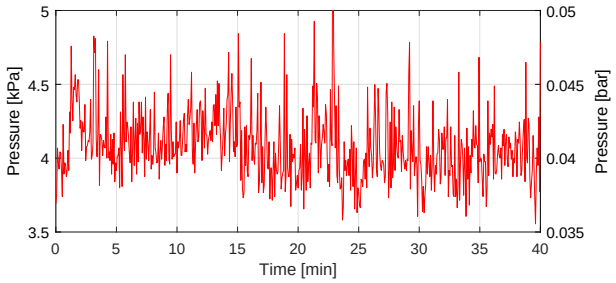


(d) PID controller - Difference pressure at the membrane inlet

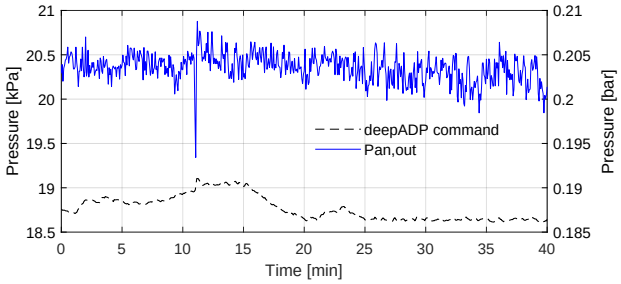
Figure 16. Experimentation results S1.2 - PEMFC response under stoichiometric disturbance and normal operating conditions (a, b) measured anode and cathode output pressures, (c, d) difference pressure at the membrane inlet



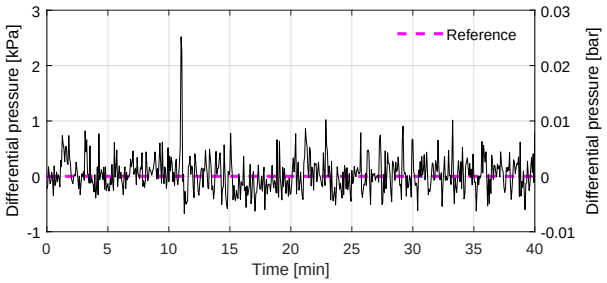
(a) Voltage and current



(b) Difference pressure at cathode

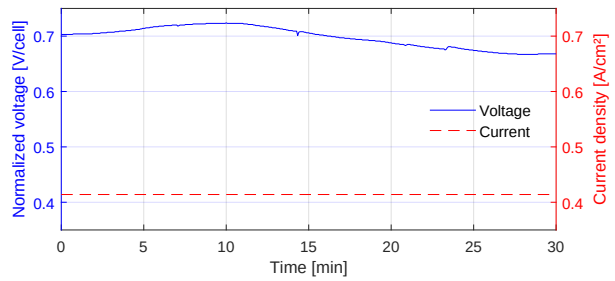


(c) Anode output pressure

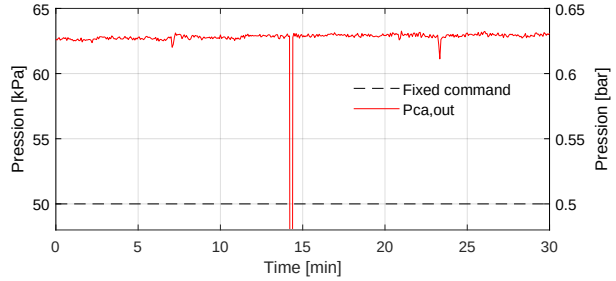


(d) Difference pressure at the membrane inlet

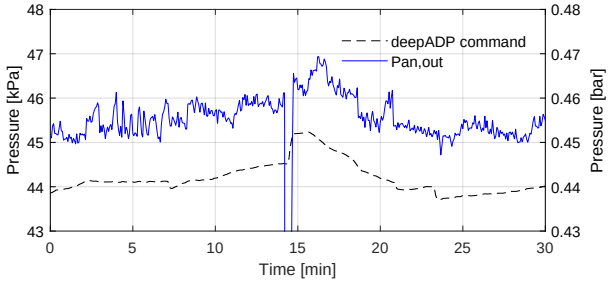
Figure 17. Experimentation results S2 - PEMFC response in presence of flooding fault with deepADP controller



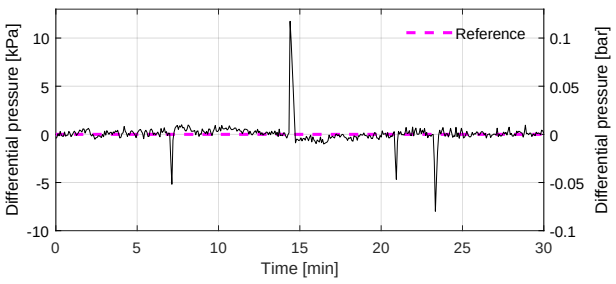
(a) Voltage and current



(b) Cathode output pressure



(c) Anode output pressure



(d) Difference pressure at the membrane inlet

Figure 18. Experimentation results S3 - PEMFC response in presence of membrane drying fault with deepADP controller

parameters ζ and γ can be integrated to improve control performance in future studies.

5. Conclusion and prospects

An adaptive and generic data-driven controller has been designed for PEMFC control. The proposed approach uses deepESN-type neural networks to estimate and predict dynamic programming functions, allowing a reinforcement approach for command generation. A fuzzy logic method has been implemented in order to speed up the learning process for an ideal application in real-time. Then, the reliability of the proposed method is guaranteed throught control stability verification. Simulation results have shown the approach efficiency over short sample times and better performance compared to traditional PID and HDP controllers. Deviations regulation have been significantly reduced in faults operating conditions. Under healthy system conditions, the control performance remains equivalent to the PID controller. However, the first error deviation requires more time to be corrected. Experimental results validated the proposed approach and demonstrated generic and adaptive capabilities to faulty PEMFCs but also to auxiliary system disturbances and nonlinearities. For the perspectives, more extensive tests should be conducted on a larger scale, such as PEMFC stack system. The next step will be to integrate the deepADP controller into a real AFTC strategy with the diagnostic and decision modules.

Acknowledgements

This works is financially supported by the Reunion Region (DIRED/20161446) and ERDF - European Commission (ARR/20170628-0013794).

Declaration of competing interest

None declared.

References

[1] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, Diagnosis and fault-tolerant control: with 121 examples, 5 application studies, and 36 exercisesOCLC: 180923420.

[2] D. Lee, J. Bae, Visualization of flooding in a single cell and stacks by using a newly-designed transparent PEMFC 37 (1) 422–435. doi:10.1016/j.ijhydene.2011.09.073.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319911021860

[3] J. Shen, L. Xu, H. Chang, Z. Tu, S. H. Chan, Partial flooding and its effect on the performance of a proton exchange membrane fuel cell 207 112537. doi:10.1016/j.enconman.2020.112537.
URL https://linkinghub.elsevier.com/retrieve/pii/S019689042030073X

[4] H. Chen, X. Zhao, T. Zhang, P. Pei, The reactant starvation of the proton exchange membrane fuel cells for vehicular applications: A review 182 282–298. doi:10.1016/j.enconman.2018.12.049.
URL https://linkinghub.elsevier.com/retrieve/pii/S0196890418313852

[5] J. Han, J. Han, S. Yu, Experimental analysis of performance degradation of 3-cell PEMFC stack under dynamic load cycle 45 (23) 13045–13054. doi:10.1016/j.ijhydene.2020.02.215.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319920308661

[6] J. Benziger, J. Nehlsen, D. Blackwell, T. Brennan, J. Itescu, Water flow in the gas diffusion layer of PEM fuel cells 261 (1) 98–106. doi:10.1016/j.memsci.2005.03.049.
URL http://linkinghub.elsevier.com/retrieve/pii/S0376738805002620

[7] L. Zhang, M. Pan, S. Quan, Model predictive control of water management in PEMFC 180 (1) 322–329. doi:10.1016/j.jpowsour.2008.01.088.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378775308002280

[8] J. Jiang, X. Yu, Fault-tolerant control systems: A comparative study between active and passive approaches 36 (1) 60–72. doi:10.1016/j.arcontrol.2012.03.005.
URL http://linkinghub.elsevier.com/retrieve/pii/S1367578812000065

[9] C. Lebreton, M. Benne, C. Damour, N. Yousfi-Steiner, B. Grondin-Perez, D. Hissel, J.-P. Chabriat, Fault tolerant control strategy applied to PEMFC water management 40 (33) 10636–10646. doi:10.1016/j.ijhydene.2015.06.115.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360319915016006

[10] C. Lebreton, C. Damour, M. Benne, B. Grondin-Perez, J.-P. Chabriat, Passive fault tolerant control of PEMFC air feeding system 41 (34) 15615–15621. doi:10.1016/j.ijhydene.2016.06.210.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360319916304153

[11] X. Wu, B. Zhou, Fault tolerance control for proton exchange membrane fuel cell systems 324 804–829. doi:10.1016/j.jpowsour.2016.05.066.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378775316306127

[12] A. Abbaspour, K. K. Yen, P. Forouzannezhad, A. Sargolzaei, An adaptive resilient control approach for pressure control in proton exchange membrane fuel cells 55 (6) 6344–6354. doi:10.1109/TIA.2019.2929256.
URL https://ieeexplore.ieee.org/document/8765382/

[13] D. Yang, Y. Wang, Z. Chen, Robust fault diagnosis and fault tolerant control for PEMFC system based on an augmented LPV observer 45 (24) 13508–13522. doi:10.1016/j.ijhydene.2020.03.063.
URL https://linkinghub.elsevier.com/retrieve/pii/S036031992030999X

[14] E. Kamal, A. Aitouche, Fuzzy observer-based fault tolerant control against sensor faults for proton exchange membrane fuel cells 45 (19) 11220–11232. doi:10.1016/j.ijhydene.2018.10.070.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319918332476

[15] C. Yan, J. Chen, H. Liu, H. Lu, Model-based fault tolerant control for the thermal management of PEMFC systems 67 (4) 2875–2884. doi:10.1109/TIE.2019.2912772.
URL https://ieeexplore.ieee.org/document/8701610/

[16] D. Rotondo, R. M. Fernandez-Canti, S. Tornil-Sin, J. Blesa, V. Puig, Robust fault diagnosis of proton exchange membrane fuel cells using a takagi-sugeno interval observer approach 41 (4) 2875–2886. doi:10.1016/j.ijhydene.2015.12.071.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319915301166

[17] T. Génevé, J. Régnier, C. Turpin, Fuel cell flooding diagnosis

based on time-constant spectrum analysis 41 (1) 516–523. doi:10.1016/j.ijhydene.2015.10.089.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360319915025860

[18] Y.-H. Lee, J. Kim, S. Yoo, On-line and real-time diagnosis method for proton membrane fuel cell (PEMFC) stack by the superposition principle 326 264–269. doi:10.1016/j.jpowsour.2016.06.113.
URL https://linkinghub.elsevier.com/retrieve/pii/S0378775316308254

[19] J. Kim, I. Lee, Y. Tak, B. Cho, State-of-health diagnosis based on hamming neural network using output voltage pattern recognition for a PEM fuel cell 37 (5) 4280–4289. doi:10.1016/j.ijhydene.2011.11.092.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319911025900

[20] S. Zhou, J. S. Dhupia, Online adaptive water management fault diagnosis of PEMFC based on orthogonal linear discriminant analysis and relevance vector machine 45 (11) 7005–7014. doi:10.1016/j.ijhydene.2019.12.193.
URL https://linkinghub.elsevier.com/retrieve/pii/S0360319919347664

[21] L. Ifrek, S. Rosini, G. Cauffet, O. Chadebec, L. Rouveyre, Y. Bultel, Fault detection for polymer electrolyte membrane fuel cell stack by external magnetic field 313 141–150. doi:10.1016/j.electacta.2019.04.193.
URL https://linkinghub.elsevier.com/retrieve/pii/S0013468619309016

[22] Z. Liu, M. Pei, Q. He, Q. Wu, L. Jackson, L. Mao, A novel method for polymer electrolyte membrane fuel cell fault diagnosis using 2d data 482 228894. doi:10.1016/j.jpowsour.2020.228894.
URL https://linkinghub.elsevier.com/retrieve/pii/S0378775320311976

[23] N. Y. Steiner, D. Candusso, D. Hissel, P. Moçoteguy, Model-based diagnosis for proton exchange membrane fuel cells 81 (2) 158–170. doi:10.1016/j.matcom.2010.02.006.
URL https://linkinghub.elsevier.com/retrieve/pii/S0378475410000509

[24] E. Dijoux, M. Benne, N. Y. Steiner, B. G. Pérez, M.-C. Péra, Active fault tolerant control strategy applied to PEMFC systems, IEEE.

[25] C.-h. Li, X.-j. Zhu, S. Sui, W.-q. Hu, M.-r. Hu, Adaptive inverse control of air supply flow for proton exchange membrane fuel cell systems 13 (6) 474–480. doi:10.1007/s11741-009-0610-3.
URL http://link.springer.com/10.1007/s11741-009-0610-3

[26] S. M. Rakhtala, R. Ghaderi, A. Ranjbar Noei, Proton exchange membrane fuel cell voltage-tracking using artificial neural networks 12 (4) 338–344. doi:10.1631/jzus.C0910683.
URL http://www.springerlink.com/index/10.1631/jzus.C0910683

[27] L. Liu, X. Li, Y.-J. Liu, S. Tong, Neural network based adaptive event trigger control for a class of electromagnetic suspension systems 106 104675. doi:10.1016/j.conengprac.2020.104675.
URL https://linkinghub.elsevier.com/retrieve/pii/S0967066120302458

[28] K. Sun, J. Qiu, H. Karimi, Neural adaptive fault-tolerant finite-time control for nonstrict feedback systems: An event-triggered mechanism 143 377–385. doi:10.1016/j.neunet.2021.06.019.
URL https://linkinghub.elsevier.com/retrieve/pii/S0893608021002513

[29] J. Lunze, J. Richter, Reconfigurable fault-tolerant control: A tutorial introduction 14 (5) 359–386. doi:10.3166/ejc.14.359-386.
URL https://linkinghub.elsevier.com/retrieve/pii/S0947358008707830

[30] P. Zhang, CHAPTER 2 - industrial control engineering, in: Advances Industrial Control Technology, william andrew Edition, pp. 41–70.
URL https://doi.org/10.1016/B978-1-4377-7807-6.10002-6.

[31] J. Shin, T. A. Badgwell, K.-H. Liu, J. H. Lee, Reinforcement learning – overview of recent progress and implications for process control 127 282–294. doi:10.1016/j.compchemeng.2019.05.029.
URL https://linkinghub.elsevier.com/retrieve/pii/S0098135419300754

[32] F. Lamnabhi-Lagarrigue, A. Annaswamy, S. Engell, A. Isaksson, P. Khargonekar, R. M. Murray, H. Nijmeijer, T. Samad, D. Tilbury, P. Van den Hof, Systems & control for the future of humanity, research agenda: Current and future roles, impact and grand challenges 43 1–64. doi:10.1016/j.arcontrol.2017.04.001.
URL http://linkinghub.elsevier.com/retrieve/pii/S1367578817300573

[33] R. Bellman, The theory of dynamic programming 60 (6) 503–516. doi:10.1090/S0002-9904-1954-09848-8.
URL http://www.ams.org/journal-getitem?pii=S0002-9904-1954-09848-8

[34] R. Nian, J. Liu, B. Huang, A review on reinforcement learning: Introduction and applications in industrial process control 139 106886. doi:10.1016/j.compchemeng.2020.106886.
URL https://linkinghub.elsevier.com/retrieve/pii/S0098135420300557

[35] X. Liu, B. Zhao, D. Liu, Fault tolerant tracking control for nonlinear systems with actuator failures through particle swarm optimization-based adaptive dynamic programming 97 106766. doi:10.1016/j.asoc.2020.106766.
URL https://linkinghub.elsevier.com/retrieve/pii/S1568494620307043

[36] Y.-C. Bo, X. Zhang, Online adaptive dynamic programming based on echo state networks for dissolved oxygen control 62 830–839. doi:10.1016/j.asoc.2017.09.015.
URL http://linkinghub.elsevier.com/retrieve/pii/S1568494617305549

[37] A. Gonzalez-Garcia, D. Barragan-Alcantar, I. Collado-Gonzalez, L. Garrido, Adaptive dynamic programming and deep reinforcement learning for the control of an unmanned surface vehicle: Experimental results 111 104807. doi:10.1016/j.conengprac.2021.104807.
URL https://linkinghub.elsevier.com/retrieve/pii/S0967066121000848

[38] L. Yin, L. Zhao, Rejectable deep differential dynamic programming for real-time integrated generation dispatch and control of micro-grids 225 120268. doi:10.1016/j.energy.2021.120268.
URL https://linkinghub.elsevier.com/retrieve/pii/S036054422100517X

[39] Z. Xu, L. Pan, T. Shen, Model-free reinforcement learning approach to optimal speed control of combustion engines in start-up mode 111 104791. doi:10.1016/j.conengprac.2021.104791.
URL https://linkinghub.elsevier.com/retrieve/pii/S096706612100068X

[40] S. Zhang, B. Zhao, Y. Zhang, Event-triggered control for input constrained non-affine nonlinear systems based on neuro-dynamic programming 440 175–184. doi:10.1016/j.neucom.2021.01.116.
URL https://linkinghub.elsevier.com/retrieve/pii/S0925231221002174

[41] N. Wu, H. Wang, Deep learning adaptive dynamic programming for real time energy management and control strategy of micro-grid 204 1169–1177. doi:10.1016/j.jclepro.2018.09.052.
URL https://linkinghub.elsevier.com/retrieve/pii/S0959652618327665

[42] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, A. Hirose, Recent advances in physical reservoir computing: A review 115 100–123. doi:10.1016/j.neunet.2019.03.005.
URL https://linkinghub.elsevier.com/retrieve/pii/S0893608019300784

[43] D. D. Fan, E. A. Theodorou, Differential dynamic programming for time-delayed systemsarXiv:1701.01882, doi:10.1109/CDC.

2016.7798330.
URL http://arxiv.org/abs/1701.01882
[44] T. Guinn, Reduction of delayed optimal control problems to nondelayed problems 18 (3) 7.
[45] C. Gallicchio, A. Micheli, Deep echo state network (DeepESN): A brief survey 11.
[46] C. Gallicchio, A. Micheli, L. Pedrelli, Design of deep echo state networks 108 33–47. doi:10.1016/j.neunet.2018.08.002.
URL https://linkinghub.elsevier.com/retrieve/pii/S0893608018302223
[47] Z. Huang, X. Xiong, W. Chen, Q. Zhang, Y. Liu, Y. Chen, Three bounded proofs for nonlinear multi-input multi-output approximate dynamic programming based on the lyapunov stability theory 39 (1) 35–50. doi:10.1002/oca.2332.
URL http://doi.wiley.com/10.1002/oca.2332
[48] B. Kim, D. Cha, Y. Kim, The effects of air stoichiometry and air excess ratio on the transient response of a PEMFC under load change conditions 138 143–149. doi:10.1016/j.apenergy.2014.10.046.
URL https://linkinghub.elsevier.com/retrieve/pii/S0306261914010964
[49] X. Chen, J. Xu, Q. Liu, Y. Chen, X. Wang, W. Li, Y. Ding, Z. Wan, Active disturbance rejection control strategy applied to cathode humidity control in PEMFC system 224 113389. doi:10.1016/j.enconman.2020.113389.
URL https://linkinghub.elsevier.com/retrieve/pii/S0196890420309250
[50] S. Chugh, C. Chaudhari, K. Sonkar, A. Sharma, G. Kapur, S. Ramakumar, Experimental and modelling studies of low temperature PEMFC performance 45 (15) 8866–8874. doi:10.1016/j.ijhydene.2020.01.019.
URL https://linkinghub.elsevier.com/retrieve/pii/S036031992030077X
[51] G. A. Futter, A. Latz, T. Jahnke, Physical modeling of chemical membrane degradation in polymer electrolyte membrane fuel cells: Influence of pressure, relative humidity and cell voltage 410-411 78–90. doi:10.1016/j.jpowsour.2018.10.085.
URL https://linkinghub.elsevier.com/retrieve/pii/S0378775318311935
[52] Y. Li, X. Zhao, S. Tao, Q. Li, W. Chen, Experimental study on anode and cathode pressure difference control and effects in a proton exchange membrane fuel cell system 3 (9) 946–954. doi:10.1002/ente.201500077.
URL http://doi.wiley.com/10.1002/ente.201500077
[53] H. Li, Y. Tang, Z. Wang, Z. Shi, S. Wu, D. Song, J. Zhang, K. Fatih, J. Zhang, H. Wang, Z. Liu, R. Abouatallah, A. Mazza, A review of water flooding issues in the proton exchange membrane fuel cell 178 (1) 103–117. doi:10.1016/j.jpowsour.2007.12.068.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378775307027991
[54] J. T. Pukrushpan, A. G. Stefanopoulou, H. Peng, Control of Fuel Cell Power Systems, Advances in Industrial Control, Springer London. doi:10.1007/978-1-4471-3792-4.
URL http://link.springer.com/10.1007/978-1-4471-3792-4