Power Systems Stability Control : Reinforcement Learning Framework

Damien Ernst, *Member, IEEE,* Mevludin Glavic, and Louis Wehenkel, *Member, IEEE*

***Abstract*— In this paper we explore how a computational approach to learning from interactions, called Reinforcement Learning (RL), can be applied to control power systems. We describe some challenges in power system control and discuss how some of those challenges could be met by using these RL methods. The difficulties associated with their application to control power systems are described and discussed as well as strategies that can be adopted to overcome them. Two reinforcement learning modes are considered : the on-line mode in which the interaction occurs with the real power system and the off-line mode in which the interaction occurs with a simulation model of the real power system. We present two case studies made on a 4-machine power system model. The first one concerns the design by means of RL algorithms used in off-line mode of a dynamic brake controller. The second concerns RL methods used in on-line mode when applied to control a Thyristor Controlled Series Capacitor (TCSC) aimed to damp power system oscillations.**

***Index Terms*— Power system control, Reinforcement learning, Agent, Optimal control, Transient stability, Power system oscillations.**

I. Introduction

POWER system stability is the property of a power system which enables it to remain in a state of equilibrium under normal operating conditions and to regain an acceptable state of equilibrium after a disturbance. All around the world power system stability margins can be observed decreasing. Among the many reasons for this, we point out three main ones.

- *The inhibition of further transmission or generation constructions by economical and environmental restrictions*. As a consequence, power systems must be operated with smaller security margins [1, 2].
- *The restructuring of the electric power industry*. The restructuring processes decrease the stability margins due to the fact that power systems are not operated in a cooperative way anymore [3].
- *The multiplication of pathological characteristics when power system complexity increases*. These include : large scale oscillations originating from nonlinear phenomena, frequency differences between weakly tied power system areas, interactions with saturated devices, interactions among power system controls, $\cdots$ [2, 3].

Beyond a certain level, the decrease of power system stability margins can lead to unacceptable operating conditions and/or to frequent power system collapses. One way to avoid these phenomena i.e., to increase power system stability margins, is to control power systems more efficiently.

The state-of-the-art in power system stability controls, including some recommendations for research and development, are comprehensively described in [4–6], and further extended in [7, 8]. The availability of phasor measurement units was recently exploited in [9] for the design of an improved stabilizing control based on decentralized/hierarchical approach. Also, an application of multi-agent systems to the development of a new defense system able to assess power system vulnerability, monitor hidden failures of protection devices, and provide adaptive control actions to prevent catastrophic failures and cascading sequences of events, is proposed in [10].

Most notably, in [4, 7, 10, 11] the need for an intelligent and systematic learning method for power system controllers so that they can learn and update their decision-making capability, was identified. Given a large power system operating with the aid of new control devices, advanced communications, and computer hardware a learning to control approach emerges as an attractive way to cope with increasing complexity in power system stability control. In this paper we introduce a methodology based on Reinforcement Learning (RL), a computational approach to learn from interactions with a real power system or its simulation model, as a framework that provides a systematic approach to design power system stability control agents.

The paper is organized as follows : section 2 discusses the overall problem of designing power system stability control schemes; section 3 introduces the theoretical foundation of reinforcement learning; sections 4 and 5 consider the application of this approach to power system problems while sections 6, 7 and 8 provide case studies in this context; sections 9 and 10 discuss related work and provide concluding remarks.

II. Power system control : a generic description

All power system control schemes are characterized by three basic elements :

- The device(s) that is (are) utilized to influence the power system dynamics. It can be a breaker, the excitation system of generator, a FACTS, $\cdots$.
- The agent(s) that controls (control) the device(s). It can be the logical rule that switches on/off a breaker, a computer that determines new generation patterns from the analysis of system security margins with respect to credible contingencies, a PID controller, etc.
- The observations realized on the power system and sent to the agent(s). These carry information about the system topology, voltage at some buses, the system frequency, etc.

This research was supported in part by EXaMINE Project funded by the European Union (IST 2000 26116).

The authors are with the Electrical Engineering and Computer Science Department, University of Liège, Sart-Tilman B28, 4000 Liège, BELGIUM. (e-mail : {ernst,glavic,lwh}@montefiore.ulg.ac.be).

D. Ernst is a Research Fellow FNRS.

Today, there are new possibilities to implement control schemes at our disposal, a few of which are highlighted below.

- *New control devices* that can influence the power system dynamics, mainly through power electronics (SVC, TCSC, UPFC, etc).
- *Better communication and data acquisition techniques*, in particular the Phasor Measurements Units (PMU) [12].
- *Better computational capabilities*, which allow one to carry out more systematic off-line studies to design control schemes, and to build control agents using intensive on-line computation to process large amounts of observations.

A generic power system stability control scheme is illustrated in Fig. 1. Observations are realized on the power system and transmitted to agents that process them in order to control appropriately the devices they are responsible for.

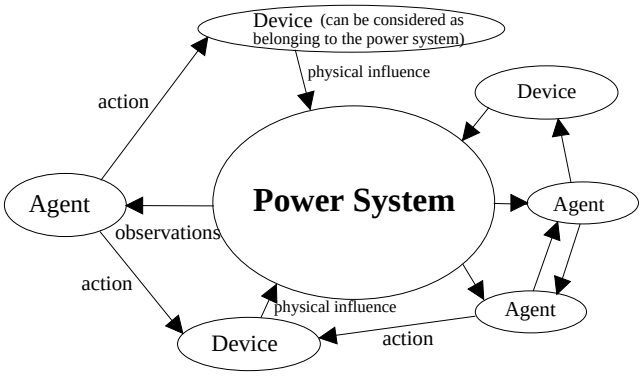


Fig. 1. Power system stability control : observations, agent, action, device and physical influence

To design a control scheme, one has to start with a set of observations O and a set of devices D (these devices already exist or can potentially be installed at particular locations of the system under consideration). Let $CS^n = \{(o_1, a_1, d_1), \cdots, (o_n, a_n, d_n)\}$ represent an n-agent control scheme, where o_i and d_i are subsets of respectively O and D, and where each a_i denotes one agent that processes the information contained in o_i to control the devices comprised in d_i. To identify the best control scheme, one should (in principle) iterate on all possible combinations $\{(o_1, d_1), \cdots, (o_n, d_n)\}$, identify for each one the best combination of controlling agents a_1, $\cdots$, a_n, evaluate each scheme and sort out the best one. Clearly, in any realistic application the number of possible control schemes is virtually infinite. But it is also true that, in most cases, some engineering knowledge, cost considerations and technical constraints can help to strongly reduce the number of candidate combinations on which one would actually have to iterate. Note however that the design of the agents can reveal itself to be much more difficult.

Within this context, RL offers a panel of methods that allow agents to learn a goal oriented control law from interaction with a system or a simulator. The RL driven agents observe the system state, take actions and observe the effects of these actions. By processing the experience they accumulate in this way they progressively learn an appropriate control law i.e., an algorithm to associate suitable actions to their observations in order to fulfill a pre-specified objective. The more experience they accumulate, the better the quality of the control law they learn. The learning of the control law from interaction with the system or with a simulator, the goal oriented aspect of the control law and the ability to handle stochastic and nonlinear problems are three distinguishing characteristics of RL.

III. Reinforcement learning : theory

RL is a general algorithmic approach to solve stochastic optimal control problems by trial-and-error. We present it in the context of a *deterministic* time-invariant system, sampled at constant rate. If x_t and u_t denote the system state and control at time t, the state at time $t+1$ is given by :

$$x_{t+1} = f(x_t, u_t) \tag{1}$$

where we assume that $u_t \in U$, $\forall t \geq 0$ and that U is finite.

A. Dynamic programming and optimal control

We use the framework of discounted infinite time-horizon optimal control to formulate the problem mathematically. Let $r(x,u) \leq B$ be a *reward* function, $\gamma \in]0,1[$ a discount factor, and denote by $u_{\{t\}} = (u_0, u_1, u_2, \cdots)$ a sequence of control actions applied to the system. The objective is to define, for every possible initial state x_0, an optimal control sequence $u^*_{\{t\}}(x_0)$ maximizing the discounted return :

$$R(x_0, u_{\{t\}}) = \sum_{t=0}^{\infty} \gamma^t r(x_t, u_t). \tag{2}$$

One can show that these optimal control sequences can be expressed in the form of a single time-invariant closed-loop control policy $u^*(.)$ i.e., $u^*_{\{t\}}(x_0) = u^*(x_t), \forall x_0 \forall t \geq 0$. In order to determine this policy one defines the value function :

$$V(x) = \max_{u_{\{t\}}} R(x, u_{\{t\}}), \tag{3}$$

which is the solution of the Bellman equation [13] :

$$V(x) = \max_{u \in U} [r(x,u) + \gamma V(f(x,u))], \tag{4}$$

and from which one can deduce the optimal policy by :

$$u^*(x) = \arg\max_{u \in U} [r(x,u) + \gamma V(f(x,u))]. \tag{5}$$

Alternatively, one can define the so-called Q-function by :

$$Q(x,u) = r(x,u) + \gamma V(f(x,u)), \tag{6}$$

and re-express $V(x)$ in terms of this function by :

$$V(x) = \max_{u \in U} Q(x,u), \tag{7}$$

and the optimal control policy by :

$$u^*(x) = \arg\max_{u \in U} Q(x,u). \tag{8}$$

Equation (8) provides a straightforward way to determine the optimal control law from the knowledge of Q. RL algorithms estimate the Q-function by interacting with the system.

B. Closed-loop versus open-loop control policies

We saw that for a deterministic system optimal open-loop and closed-loop policies are (strictly) equivalent. For stochastic systems, however, closed-loop policies yield in general better performances. For example, for controlled Markov processes the best closed-loop policy is optimal among all possible non-anticipating policies, and, because it takes into account real-time information to take decisions, it is in general superior to the best open-loop control policy. It is obtained by solving a generalized form of the Bellman equation (see section III-C). This supports the intuitive and often quoted idea that closed-loop policies are more robust than open-loop ones.

C. State space discretization

When the state space is finite, the Q-function can be represented exactly in tabular form. If the state space is infinite (as in most power system problems) the Q-function has to be approximated [14]. In our applications, we use a state space discretization technique to this end, since this technique is easy to implement, numerically stable and allows the use of model based learning algorithms. It consists in dividing the state space into a finite number of regions and considering that on each region the Q-function depends only on u. Then, in the RL algorithms, the notion of state used is not the real state of the system x but rather the region of the state space to which x belongs. We will use the letter s to denote a discretized state (or region), $s(x)$ the region to which the (true) state x belongs, and S the finite set of discretized states. Notice that the sole knowledge of the region $s(x_t)$ at some time instant t together with the control value u is not sufficient (in general) to predict with certainty the region to which the system will move at time $t+1$. We model this uncertainty by assuming that the sequence of discretized states followed by our system under a certain control sequence is a Markov chain characterized by time-invariant transition probabilities $p(s'|s,u)$, which define the probability to jump to a state $s_{t+1} = s'$ given that $s_t = s$ and $u_t = u$. Given these transition probabilities and a discretized reward signal i.e., a function $r(s,u)$, we reformulate our initial control problem as a Markov Decision Process (MDP) and search for a (closed-loop) control policy defined over the set of discrete states S, that maximizes the *expected* return (with respect to the probabilistic model defined by the controlled Markov chain). The corresponding Q-function is characterized by the following Bellman equation :

$$Q(s,u) = r(s,u) + \gamma \sum_{s' \in S} p(s'|s,u) \max_{u \in U} Q(s',u), \qquad (9)$$

the solution of which can be estimated by a classical dynamic programming algorithm like the value iteration or the policy iteration algorithm [13, 14]. The corresponding optimal control policy is extended to the original control problem, which is thus defined by the following rule :

$$\hat{u}^*(x) = u^*(s(x)) = \arg\max_{u \in U} Q(s(x),u). \qquad (10)$$

D. Learning the Q-function from interaction

RL methods either estimate the transition probabilities and the associated rewards (model based learning methods) and then compute the Q-function, or learn directly the Q-function without learning any model (non-model based learning methods). For the purpose of this paper we use a model based algorithm because these algorithms make more efficient use of data gathered, they find better policies, and handle changes in the environment more efficiently [15, 16]. The basic iteration of a model based RL algorithm is then as follows [14–16] :

1) at time t, the algorithm observes the state x_t sends a control signal u_t, it receives information back from the system in terms of the successor state x_{t+1} and reward $r_t = r(x_t, u_t)$;
2) it uses these four values to update the model of the discrete system (transition probabilities and reward) :

$$N_1(s(x_t),u_t) \leftarrow N_1(s(x_t),u_t) + 1 \qquad (11)$$

$$N_2(s(x_t),u_t,s(x_{t+1})) \leftarrow N_2(s(x_t),u_t,s(x_{t+1})) + 1 \qquad (12)$$

$$r(s(x_t),u_t) \leftarrow \frac{(N_1(s(x_t),u_t)-1)r(s(x_t),u_t) + r_t}{N_1(s(x_t),u_t)} \qquad (13)$$

$$p(s'|s(x_t),u_t) \leftarrow \frac{N_2(s(x_t),u_t,s')}{N_1(s(x_t),u_t)} \quad \forall s' \in S \qquad (14)$$

   where $N_1(s,u)$ and $N_2(s,u,s')$ are initialized to zero everywhere at the beginning of the learning[1].
3) it solves (partially) the Bellman equation (9) in order to update the estimate of the Q-function.

Note that at each time-step the algorithm selects a control signal, by using the so-called ϵ-*greedy* policy. This consists in choosing with a probability of ϵ a control action at random in U, and with a probability of $1-\epsilon$ the "optimal" control action associated with the current state by the current estimate of the Q-function. The value of ϵ defines the so-called "exploration-exploitation" tradeoff used by the algorithm : the smaller the value of ϵ, the better the RL algorithms exploit the control law they have learned and the less they explore their environment.

E. Comments

The convergence of the RL algorithms and the optimality of the policy to which they converge can be shown under some restrictive assumptions[2]. These assumptions are normally not satisfied in real-life, but nevertheless, in many practical situations these algorithms are able to produce good control policies in acceptable time. Of course, these issues depend on the problem considered, on the discretization method used, and on the values of algorithm parameters such as ϵ.

IV. Using RL in power system control

As in any other dynamic system the RL methods in power systems can be applied in two modes : on-line and off-line. These two modes of application are outlined in Fig. 2. The on-line mode consists in using the RL driven agent directly on the

[1]This estimation is known as the Maximum Likehood Estimation (MLE). Other algorithms described in [16] can be used to estimate the structure.

[2]Among these assumptions, there is notably the necessity for each state-action pair to be visited an infinite number of times.

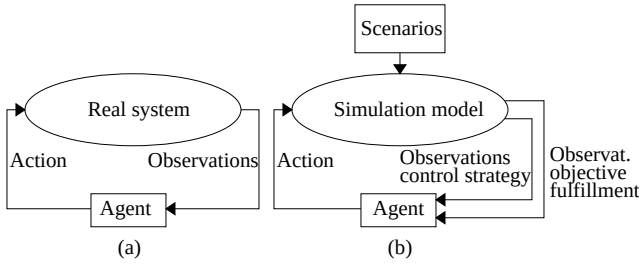


Fig. 2. Two modes of application of RL methods : (a) on-line, (b) off-line

real system. In this case there is no need for a power system model. This is particularly interesting when it is difficult to model the power system or when some phenomena are difficult to reproduce in a simulation environment. Moreover, with the agent learning continuously, it can adapt to changing operating conditions. The main drawback of the on-line mode originates from the fact that the agent may jeopardize system stability because at the beginning of the interaction no experience is available to the RL driven agent to control adequately the system. One solution to this problem is to use the agent in a simulation environment first (off-line mode). In this mode, the RL driven agent interacts with a simulation model of the system. Once the agent behavior is sufficiently good,

- one may implement the RL driven agent on the real system where it will benefit from the experience it has acquired in the simulation environment and still be able to improve its behavior from interaction with the real system;
- one may extract the off-line learned control policy so as to implement it on the real system, without further learning on the real system. In this case the observations used by control agent interacting with the real system can be reduced with respect to the ones required by the RL driven agent in the simulation environment. Indeed, the observations identified as "Observations objective fulfillment" in Fig. 2b (rewards) are not needed anymore in the real-time mode, since no further learning is carried out.

V. Practical problems in application of reinforcement learning to power system control and solutions

Successful application of RL methods to real power system control requires that some practical issues inherent to the methods should be solved. We now describe and discuss different types of problems met when using RL methods to control a real power system and propose strategies that can be adopted to overcome them.

A. Curse of dimensionality problem

The discretization strategy used to apply RL algorithms to continuous state space control problems makes sense if the finite MDP learned by interacting with the power system is able to approximate well the initial control problem. One can assume that this is indeed satisfied if the discretization step according to each state variable is sufficiently fine. But when the number of state variables becomes too high, the finite MDP can be composed of too many states to hope to obtain a good approximation of the optimal stationary policy in a reasonable learning time or even to match computer capabilities. The approach we propose to overcome this difficulty consists to "preprocess" the high dimensional system state to extract a (much) lower dimensional input signal (referred to as the pseudo-state and denoted by $\tilde{x}$) and proceed exactly as if it were the real state of the system. Such an approach makes sense if the selected input signal catches the correct information for the considered problem. The choice of the appropriate input signal is usually based on the engineering knowledge of the control problem considered.

B. Partially observable system states

Usually, the observation of the system state is incomplete and noisy. The strategy we propose to cope with this problem consists of using information from several successive time-steps to choose a control action. More precisely, we define a pseudo-state from the history of the observations done and the actions taken and proceed exactly as if it were really the real state of the system. The pseudo-state of the system at time t is defined by the following equation :

$$\tilde{x}_t \quad = \quad (o_t, \cdots, o_{\max(0,t-Nbo+1)}, u_{t-1}, \cdots, u_{\max(0,t-Nbu)})$$

where o represents the observation done on the system, Nbo and Nbu determine respectively the number of successive observations and the number of successive actions taken by the RL algorithm that are used in the definition of the pseudo-state. In principle, the larger the values of Nbo and Nbu are, the better is the information about the system state contained in the pseudo-state. But increasing these numbers also increases the dimensionality of the (pseudo) state space and may therefore penalize the learning speed.

C. Non-stationarity of the system

The theory underlying the RL algorithms assumes that the system dynamics and the reward function do not depend explicitly on time (time invariance). Unfortunately, for many systems and especially power systems, this assumption does not hold in practice. The strategy we propose here to deal with this difficulty assumes that the system changes "slowly" with respect to the speed of learning. Under these conditions we use slightly modified versions of reinforcement learning algorithms, which increase the relative weight of the most recent observations done on the system with respect to less recent ones. For example, this can be achieved by replacing iterations (11) and (12) by :

$$N_1(s(x_t), u_t) \leftarrow \beta N_1(s(x_t), u_t) + 1 \tag{15}$$

$$N_2(s(x_t), u_t, s') \leftarrow \beta N_2(s(x_t), u_t, s') \quad \forall s' \in S \tag{16}$$

$$N_2(s(x_t), u_t, s(x_{t+1})) \leftarrow N_2(s(x_t), u_t, s(x_{t+1})) + 1 \tag{17}$$

where $0 < \beta < 1$.

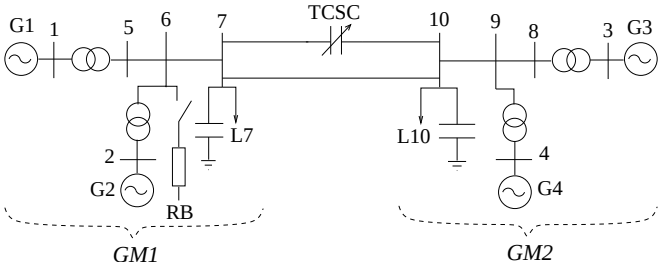


Fig. 3. A four-machine power system

VI. Description of the test power system model

To illustrate capabilities of the proposed framework to control power system stability, we make use of the four-machine power system model described in Fig. 3. Its characteristics are mainly inspired from [1]. All the machines are modeled with a detailed generator model, slow direct current exciter, automatic voltage regulator (AVR), and speed regulator. The loads are modeled as constant current (active part) and constant impedance (reactive part). When the system operates in steady-state conditions, the generators G1, G2 (hydro) and G3, G4 (thermal) produce approximately the same active powers (700 MW) and the two loads L7, L10 consume respectively 990 and 1790 MW. Below we present simulation results obtained by applying RL to two problems : dynamic brake control in an off-line mode and TCSC control in an on-line mode.

VII. Off-line design of a dynamic brake controller

The agent that controls the dynamic brake has a threefold objective : to damp large electromechanical oscillations, to avoid the loss of synchronism between the generators when a severe incident occurs, and to limit the time the dynamic brake is switched on. The resistive brake (RB) is located at bus 6 (Fig. 3) and sized as $g = 5.0\,p.u.\,mhos$ on a 100 MVA base (500 MW). This is a reasonable value in view of the fact that a 1400 MW braking resistor is presently in use [1, 5].

A. Pseudo-state and reward definition

The control scheme we propose assumes that the system can be decomposed into two areas (identified by *GM1* and *GM2* in Fig. 3) such that only the relative motion of these two areas provides interesting information, both to decide control actions (pseudo-state definition) and to measure performance (reward definition). The 60-dimensional state space of the system is thus a priori reduced to a 2-dimensional signal composed of relative angle and relative speed of the two groups of machines. The pseudo-state at time t is thus represented as :

$$\tilde{x}_t = (\delta_t, \omega_t) \tag{18}$$

where δ_t and ω_t are equivalent angle and speed [17][3].

[3]The determination of δ_t and ω_t requires the knowledge of the angle and the speed of each generator. These variables can be either measured directly or estimated, but we neglect transmission delays and measurement errors in our study.

The control objective of the agent is defined by the discount factor γ and the reward function. We define the reward function as follows :

$$r(x,u) = \begin{cases} -|\omega| - cu & \text{if } |\delta| \leq \pi\, rad \\ -1000 & \text{if } |\delta| > \pi\, rad \end{cases}$$

where the $u \in \{0,1\}$ (0 meaning that the brake is switched off and 1 that it is switched on) and where c determines how much we penalize the fact that the brake is on ($c = 2.0$ in the simulations reported below). With this criterion, large electromechanical oscillations correspond to large negative rewards. Furthermore, to strongly penalize unstable operation, a very negative reward (-1000) is obtained when the system has lost synchronism (we consider this to be the case when $|\delta|$ is greater than $\pi\, rad$). When the loss of stability is observed a terminal state is reached and the algorithms stop interacting with the system until they are reinitialized.

The sampling period is chosen equal to $100\,ms$ which means that data is acquired and the value of the control could change every $100\,ms$. The discount factor γ of the return computation has been fixed to 0.95, which corresponds to a 90% discount after about 4.5 seconds of real-time.

B. Learning scenario description

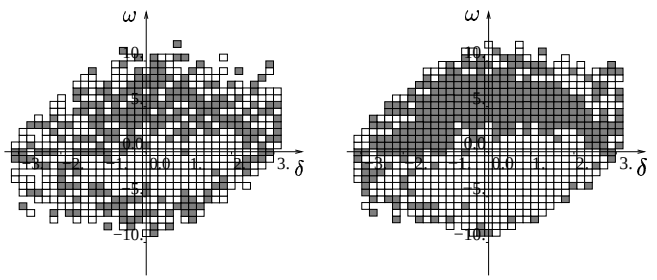
The RL algorithm is used to learn a closed-loop control law able to avoid loss of synchronism and damp large oscillations. Since combinations of various pre-fault configurations (topology and operating point) and fault clearing schemes may lead to a variety of post-fault configurations, we need to check the robustness of the control law obtained after training. Thus, although we will realize the learning by using always the same configuration, after convergence of the RL algorithm we will assess the resulting control law robustness on other scenarios corresponding to configurations different from the one used for learning.

The learning period is partitioned into different scenarios. Each scenario starts with the power system being at rest and is such that at $10\,s$ a short-circuit near bus 10 occurs. The fault duration is chosen at random in the interval $[250, 350]\,ms$, and the fault is self-cleared. The simulation then proceeds in the post-fault configuration until either instability is detected or else the time is greater than $60\,s$. Since we want to train the controller in the post-fault configuration, no learning (i.e. no model update and no Q-function update) is done in the during fault period. A total number of 1000 learning scenarios are generated, out of which 163 were unstable.

C. Algorithm parameters and learned control policy

During learning the $\epsilon - greedy$ factor is set to 0.1 which corresponds to a relatively high exploration rate, liable to accelerate convergence speed. The pseudo-state space is discretized in a rectangular and uniform way, with a discretization step of 0.18 for the angle and of 0.75 for the speed.

Figure 4a shows the control law obtained in the (δ, ω) plane after 100 scenarios have been presented to the RL algorithm. On this figure, each tile corresponds to a discretized state. Note that only the tiles that have been visited during the learning



(a) After 100 scenarios (b) After 1000 scenarios

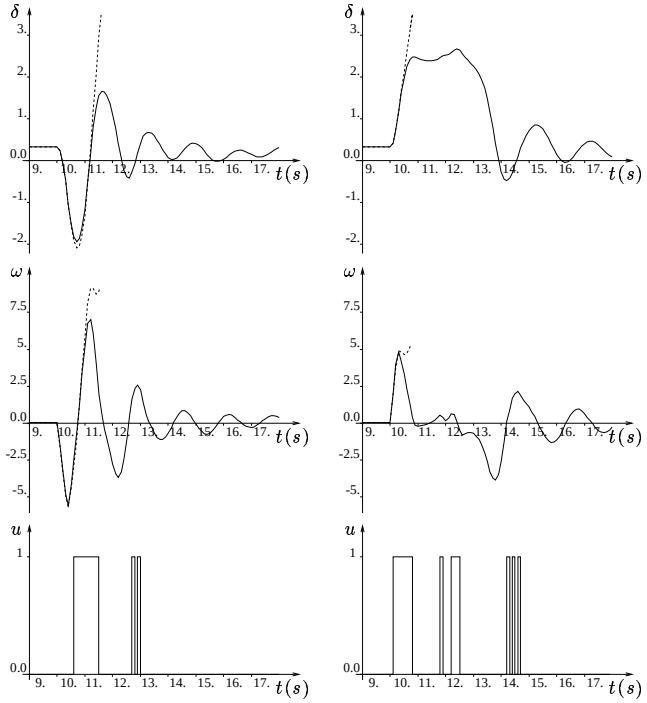
Fig. 4. The learned control strategy . δ is expressed in rad and ω in rad/s.

are represented. The dark tiles correspond to states where the control value is 1 (brake on) and the light ones to the opposite case. We observe that after 100 scenarios, the control law still seems rather erratic, which is due to the fact the RL algorithm has not yet converged. After 1000 scenarios (Fig. 4b), one can observe that an organized structure has appeared in the way the tiles are distributed. At this stage, additional learning can only bring minor changes to the learned control law.

D. Effectiveness and robustness of the learned control policy

When the uncontrolled system is subjected to the fault scenario used during the learning (a self-cleared short-circuit at bus 10), the maximum fault duration it can withstand without losing stability is $215\,ms$. On the other hand, when the dynamic brake is used with the control law represented on Fig. 4b, the system is stable even for a $350\,ms$ fault duration (the evolution of δ and u is represented on Fig. 5a).

To assess the control law robustness with respect to a fault scenario not met during the learning we consider the sequence of events that consists in applying a fault near bus 7 and in clearing it by opening one of the two lines connecting bus 7 to bus 10. The maximum fault duration the uncontrolled system can withstand when subjected to such a fault scenario is $141\,ms$ while it is $252\,ms$ for the controlled system, which illustrates the control law robustness. The corresponding behavior (controlled vs uncontrolled) is shown on Fig. 5b.



$350\,ms$ self-clearing fault at bus 10.

(a)

$250\,ms$ fault near bus # 7 cleared by opening a line connecting bus 7 to bus 10.

(b)

Fig. 5. Evolution of δ (rad), ω (rad/s) and u (Ω) for two different fault scenarios. The fault is applied at $t = 10\,s$. The control strategy used is the one represented on figure 4b. The dashed curve represents the δ/ω evolution that would have been obtained in the case of an uncontrolled system ($u = 0$).

VIII. On-line learning to control a TCSC

In this section we focus on how to control by means of RL algorithms a TCSC device in order to damp power system oscillations, a phenomenon becoming even more important with the growth of extensive power systems and especially with the interconnection of these systems with ties of limited capacity. The TCSC is considered as a variable reactance placed in series with a transmission line (line 7-10 in Fig. 3). The reactance of the TCSC, denoted by X_{FACTS}, responds to the first order differential equation :

$$\frac{dX_{FACTS}}{dt} = \frac{X_{ref} - X_{FACTS}}{T_{FACTS}} \tag{19}$$

where X_{ref} represents the FACTS reactance reference and where T_{FACTS} has been chosen, in accordance with the technical specifications of such a FACTS device [18], equal to $60\,ms$. The control variable for this system is X_{ref} and it is supposed to belong to the interval $[-61.57, 0]\,\Omega$. A value of $-61.57\,\Omega$ for X_{FACTS} corresponds approximately to a 30 % compensation of the line on which the FACTS is installed. Our aim is to control this device by using only locally available measurements and to show how the RL algorithm would operate in on-line mode, in particular how it could adapt the control strategy to changing operating conditions.

To make these simulations more interesting, we start by modifying the gains of the machines AVR in order to yield a system which is originally negatively damped. Figure 6 shows under these conditions the power flowing through the line 7-10 when $X_{FACTS} = -61.57$; it corresponds to a stable limit cycle with amplitude governed by the excitation current limitation.

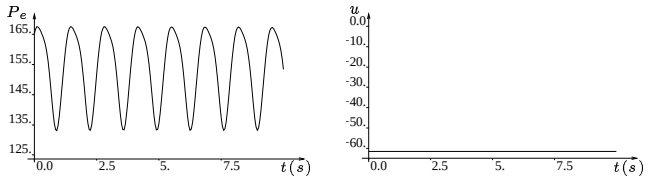


Fig. 6. Electrical power oscillations (MW) occurring when u (Ω) is constant and equal to $-61.57\,\Omega$

A. State and reward definition

In order to enable the proper operation of the RL algorithm in on-line mode all the quantities used by this algorithm must be defined on the basis of real-time measurements that are used as inputs to the controller and to the learning agent. Since we want a local control algorithm we need to use

measurements available close to the location of the FACTS device. We chose a minimal set of a single local measurements, namely of the active power flow through the line in which the FACTS is installed. This quantity is obtained at each time step of $50\,ms$. It is used to define the rewards and pseudo-states used by the RL algorithm (including the detection of loss of synchronism). To construct the pseudo-state that will be used inside the RL algorithm, we further need to define Nbo and Nbu. Preliminary simulations have shown that a choice $Nbo = 3$ and $Nbu = 2$ leads to a good compromise between information and convergence speed of RL. Thus the pseudo-state at time t is defined by the following expression :

$$\tilde{x}_t = (P_{e_t}, P_{e_{t-1}}, P_{e_{t-2}}, u_{t-1}, u_{t-2}) \tag{20}$$

The aim of the control is to maximize damping of the electrical power oscillations in the line. This choice is motivated by the fact that damping improvement of the electrical power oscillations should also lead to an overall improvement of the power system damping. Thus, we define the reward by :

$$r(x,u) = \begin{cases} -|P_e - \overline{P}_e| & \text{if } |P_e| \leq 250\,\text{MW} \\ -1000 & \text{if } |P_e| > 250\,\text{MW} \end{cases} \tag{21}$$

where $\overline{P_e}$ represents the steady-state value of the electric power transmitted through the line, and the condition $|P_e| > 250\,\text{MW}$ is used to detect instability. When this latter condition is reached the learning and control algorithms stop interacting with the system until they are reinitialized. The discount factor γ is set to 0.98, which corresponds to a 90 % discount after about 5.5 seconds of real-time.

Note that the steady-state value of the electrical power is dependent on several aspects (operating point, steady-state value of X_{ref}) and so cannot be fixed before-hand. Thus, rather than to use a fixed value of $\overline{P}_e$, we estimate its value on-line using the following equation :

$$\overline{P}_e = \frac{1}{1200} \sum_{k=0}^{1199} P_{e_{t+1-k}}, \tag{22}$$

which is a moving average over the last $1200 * 50\,ms = 60\,s$. This limited window provides the algorithm with some adaptive behavior, which is preferable when the power system operating conditions change.

B. The value of parameters and cases considered

The control set is discretized in five values equal to $U = \{-61.57, -46.18, -30.78, -15.39, 0.\}$ while electrical power transmitted in the line is discretized in 100 values within interval $[-250, 250]$ MW.

A relatively small value (0.01) is chosen for ϵ in order to guarantee that the RL algorithm exploits almost at its best the control law it has already learned.

We first apply the RL algorithm in "steady-state" conditions, which means that the system operates on the stable limit cycle as depicted on Fig. 6a. The second case we consider is when the system load is not constant (non-autonomous environment). The load variation is cyclic with period of 5 hours, and it has been modeled according to the equation :

$$z(t) = z(0) - 0.3z(0)\sin(2\pi\frac{1}{5*3600}t) \tag{23}$$

where z stands for the active or reactive parts of the load. Moreover, in order to follow the load, the electric power production reference on each machine has also been modeled by equation (23). We have chosen a $5\,h$ period of the load curve rather than $24\,h$ period to lighten the computational burdens needed to simulate this 61 state variable power system during several periods. When dealing with a non-autonomous environment the RL algorithm will use iterations (15), (16) and (17) with $\beta = 0.95$ in the model update process in order to give more weight to the most recent observations.

C. Results in steady-state conditions of the power demand

The progressive learning of the control agent is illustrated on Figs. 7 to 9, in terms of the power flow in the line and in terms of the variation of the control variable over a period of 10 seconds. They are further commented below.

After $10\,min$ of learning control, we see on Figs. 7a and 7b that the magnitude of the P_e oscillations is still very large and the evolution of the action u seems to be driven by an almost random process. The RL algorithm does not have yet a sufficient knowledge about the system to act efficiently.

After $1\,h$ of control (Figs. 8a and 8b), the electrical power transferred in the line starts being well damped. An organized structure appears in the sequence of actions taken.

After $10\,h$ of control (Figs. 9a and 9b), the results are more impressive. The magnitude of the electrical power oscillations has strongly decreased. The variation of the control variable u has a periodic behaviour of approximately the same frequency

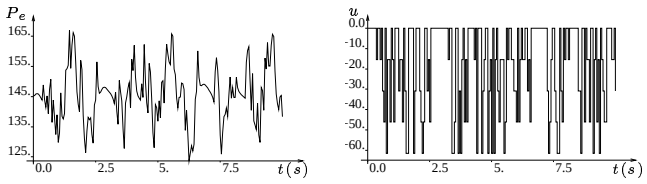


Fig. 7. Electrical power (MW) and u (Ω) evolution after $10\,min$ of control

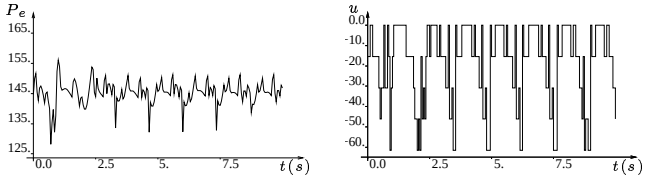


Fig. 8. Electrical power (MW) and u (Ω) evolution after $1\,h$ of control

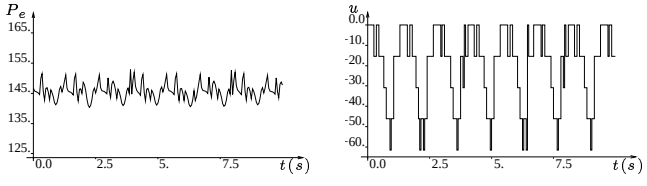


Fig. 9. Electrical power (MW) and u (Ω) evolution after $10\,h$ of control

($0.8\,Hz$) as the electrical power oscillations observed when no control occurs. The harsh aspect of the electrical power observed comes from the discontinuous variation of the control variable u. Such behaviour could be circumvented by increasing the time delay of the FACTS (image of the time needed by the TCSC to meet the reactance reference u) or by imposing a continuous variation of u by means of an integrator.

D. Results under cyclic power demand variations

Let us see how the learning algorithm is able to adapt its behavior to changing operating conditions. To this end, let us consider the second case i.e., a scenario where in addition to the short term dynamics (limit cycle) we superpose on the system a periodic change in system demand according to eqn. (23). In addition to this cyclic trend, we introduce into the simulations some stochastic behavior by superposing to the individual load values some white noise. To see how the learning algorithm reacts we introduce the following performance measure :

$$\overline{R}_t = \frac{1}{1200} \sum_{t'=t}^{t+1199} \sum_{k=t'}^{\infty} \gamma^{k-t'} r_{k+1}, \tag{24}$$

which is, at a certain time t, an average over the next minute of the total return obtained by the algorithm. This quantity will be largely negative if the control law is not satisfactory and ideally close to zero if the control performance is optimal. The three curves depicted on Fig. 10 represent the evolution for three successive $5\,h$ periods (t_i represents the beginning time of each cycle) :

- the first cycle of 5 hours (curve labeled $t_i = 0\,h$) : during this cycle the RL algorithm is kept inactive and thus the return directly reflects the amplitude and periodicity of the limit cycle which follow the load demand,
- the second cycle of 5 hours (curve labeled $t_i = 5\,h$) : here learning is active and slowly improves the behavior,
- the third cycle of 5 hours (curve labeled $t_i = 10\,h$) : learning is still active and still continues to improve the behavior, by almost completely removing the periodic component.

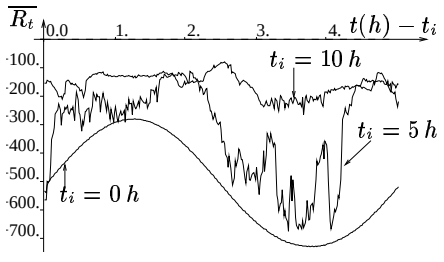


Fig. 10. $\overline{R}_t$ as a function of the learning time

The conclusion that can be drawn from these results is that the RL algorithm is indeed able to adapt to time-varying operating conditions and that even after several hours of control it continues to improve the quality of the control policy. The stochastic aspect introduced by the loads has only a second order influence on the control quality due to the fact that RL algorithms are from the beginning shaped for control in stochastic environments and are thus well adapted to stochastic perturbations.

IX. Related work

The application of RL algorithms to power system stability control is still in its infancy. Considerable research efforts have been done at the University of Li`ege [16, 19–21] and this paper is a result of those efforts.

The earliest research reports date back to 1999 and 2000 [19, 22, 23]. In [19] the use of RL in electric power system closed-loop emergency control, were investigated. A RL algorithm was used to determine a discrete switching control law to trip generators so as to avoid loss of synchronism. An investigation of a learning coordinated fuzzy-logic control strategy, based on the interconnected learning automata, for the control of dynamic quadrature boosters, installed distributively in a power system, to enhance power system stability is reported in [23]. A non-model based RL technique was employed to search for optimal fuzzy-logic controller parameters according to a given performance index, to control the boosters in a coordinated fashion. Using a non-model based RL algorithm to optimize synchronous generator PID controller parameters was explored in [22]. The works presented in [22, 23] are good examples on how the RL methods can be combined with existing local controllers where the learning component learns only a setting of the parameters while the local controller assures a baseline control performance. In all the mentioned reports only off-line mode of RL application were employed.

In this paper the application of the RL algorithms to solve power system stability problems is generalized by discussing and suggesting solutions to practical problems that can be met in the RL application to power system control, issues not tackled earlier.

The conceptual design of a hybrid multi-agent system for self-healing power infrastructure defense system presented in [10] consists of three layers : reactive, coordination, and deliberative layer. The reactive layer (low level layer), includes a set of heterogeneous agents acting locally over a particular set of power system components, plants or substations. The agents placed on the high-level layer, the deliberative layer, can analyze and monitor the power system from a wide-area point of view. The coordination of a number of agents is an important issue. This task is envisioned in this publication to be assigned to the agents in the coordination layer. Further consideration on the robustness of the team of agents ended with the conclusion that the agents within the proposed system, in all three layers, need an intelligent and systematic learning method to learn and update their decision-making capability through direct interaction with the dynamic environment. The RL methods are mentioned as a possible approach accompanied with a note that RL application within the proposed system should be done with great care, especially in the reactive layer, and intensive research in the field should be done prior its real application.

The work from [10] is further extended in [24] as a feasibility study of a RL method for an agent's adaptive learning capability with load shedding control schemes.

We strongly believe that the RL is an effective computational approach to cope with these technical challenges. The framework presented in this paper is also a contribution to the design of reactive agents (the most difficult to design) acting over a particular set of power system components.

X. Conclusions

Reinforcement learning methods can reveal themselves to be an interesting tool for power system agents design for several reasons enumerated below.

- These methods do not make any strong assumptions on the system dynamics. In particular, they can cope with partial information and non-linear and stochastic behaviors. They can therefore be applied to design many, if not all, practical types of control schemes.
- They learn closed-loop control laws known to be robust. This aspect is important notably when the real power system is facing situations that were not accounted for in the simulation model.
- RL methods open avenues to adaptive control since the RL driven agents learn continuously and can adapt to changing operating conditions or system dynamics.
- They can be used in combination with traditional control methods to improve performances. As an example, they could be used to determine parameters of control laws obtained by a linear analysis in which case the RL driven agent does not control directly the device but rather some parameters of another agent responsible for the device control.

Along with basic research on the reinforcement learning algorithms (in particular, state space discretization techniques), we believe that future research should be oriented towards applications in power systems of these approaches.

In particular, in the applications studied in the present paper, we suggest to further test these approaches on variants of these problems and on more large scale power system models, taking into account also some important aspects such as measurement errors and communication delays (specially in the context of centralized schemes).

More generally, we believe that it would be of interest to study the behavior of multi-agent reinforcement systems in terms of interactions, convergence and potential conflicts. Finally, we also believe that a more in depth consideration of ways to combine reinforcement learning algorithms with classical control theory methods would be specially valuable in the case of power systems stability control applications.

References

[1] P. Kundur, *Power System Stability and Control*. McGraw-Hill, 1994.

[2] CIGRE Task Force 38.02.16, "Impact of the Interaction Among Power System Controls, Technical Report," 2000.

[3] C. L. Demarco, "The Threat of Predatory Generation Control : Can ISO Police Fast Time Scale Misbehavior ?" in *Proceedings of Bulk Power System Dynamics and Control IV-Restructuring (IREP 1998)*, Santorini, Greece, August 1998, pp. 281–289.

[4] L. Wehenkel, "Emergency control and its strategies," in *Proceedings of the 13-th PSCC*, Trondheim, Norway, 1999, pp. 35–48.

[5] C. Taylor (Convener), "Advanced Angle Stability Controls," CIGRE, Available : http://transmission.bpa.gov/orgs/opi/CIGRE, Technical Report 155, 1999.

[6] D. Karlsson (Convener) CIGRE Task Force 38.02.19, "System Proctection Scheme in Power Networks," Available : http://transmission.bpa.gov/orgs/opi/CIGRE, Technical Report 187, June 2001.

[7] C. W. Taylor, "Response-Based, Feedforward Wide-Area Control," position paper for NSF/DOE/EPRI Sponsored Workshop on Future Research Directions for Complex Interactive Networks, Washington DC, USA, 16-17 November 2000.

[8] S. Rovnyak, "Discussion to Ref. 7," NSF/DOE/EPRI Sponsored Workshop on Future Research Directions for Complex Interactive Networks, Washington DC, USA, 16-17 November 2000.

[9] I. Kamwa, R. Grondin, and Y. Hebert, "Wide-Area Measurement Based Stabilizing Control of Large Power Systems - A Decentralized/Hierarchical Approach," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 136–153, 2001.

[10] C. C. Liu, J. Jung, G. T. Heydt, and V. Vittal, "The Strategic Power Infrastructure Defense (SPID) System," *IEEE Control System Magazine*, vol. 20, pp. 40–52, 2000.

[11] A. Diu and L. Wehenkel, "EXaMINE - Experimentation of a Monitoring and Control System for Managing Vulnerabilities of the European Infrastructure for Electrical Power Exchange," in *Proceedings of the IEEE PES Summer Meeting, panel Session on Power System Security in the New Market Environment*, Chicago, USA, 2002.

[12] A. G. Phadke, "Synchronized Phasor Measurements in Power," *IEEE Computer Applications in Power*, vol. 6, no. 2, pp. 10–15, April 1993.

[13] R. Bellman, *Dynamic Programming*. Princeton University Press, 1957.

[14] R. S. Sutton and A. G. Barto, *Reinforcement Learning, an Introduction*. MIT Press, 1998.

[15] A. Moore and C. Atkeson, "Prioritized Sweeping: Reinforcement Learning with Less Data and Less Real Time," *Machine Learning*, vol. 13, pp. 103–130, 1993.

[16] D. Ernst, "Near optimal closed-loop control. Application to electric power systems." Ph.D. dissertation, University of Liège, 2003.

[17] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient Stability of Power System. A Unifi ed Approach to Assessment and Control*, ser. Power Electronics and Power Systems. Kluwer Academic Publishers, 2000.

[18] N. G. Hingorani and L. Gyugyi, *Understanding FACTS*. IEEE press, 2000.

[19] C. Druet, D. Ernst, and L. Wehenkel, "Application of reinforcement learning to electrical power system closed-loop emergency control," in *Proceedings of PKDD'2000*, September 2000, pp. 86–95.

[20] D. Ernst and L. Wehenkel, "FACTS devices controlled by means of reinforcement learning algorithms," in *Proceedings of PSCC'2002*, Sevilla, Spain, June 2002.

[21] M. Glavic, D. Ernst, and L. Wehenkel, "A Reinforcement Learning Based Discrete Supplementary Control for Power System Transient Stability Enhancement. To be presented at ISAP'2003," Vollos, Greece, 2003.

[22] B. H. Li and Q. H. Wu, "Learning coordinated fuzzy logic control of dynamic quadrature boosters in multimachine power systems," *IEE Part C-Generation, Transmission, and Distribution*, vol. 146, no. 6, pp. 577–585, 1999.

[23] K. H. Chan, L. Jiang, P. Tilloston, and Q. H. Wu, "Reinforcement Learning for the Control of Large-Scale Power Systems," in *Proceedings of EIS'2000*, Paisley, UK, 2000.

[24] J. Jung, C. C. Liu, S. L. Tanimoto, and V. Vittal, "Adaptation in Load Shedding Under Vulnerable Operating Conditions," *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 1199–1205, 2002.

Damien Ernst graduated as an Electrical Engineer from the University of Liège, in 1998. He is currently PhD student at University of Liège and a FNRS Research Fellow. His research interests lie in the field of optimal control, reinforcement learning and power system control.

Mevludin Glavic received MSc and PhD degrees from the University of Belgrade, Yugoslavia, and the University of Tuzla, Bosnia. As a postdoctoral researcher, within the Fulbright Program, he spent academic year 1999/2000 with the University of Wisconsin-Madison. Presently, he is Research Fellow at University of Liège, Department of Electrical Engineering and Computer Science. His fields of interest include power system control and optimization.

Louis Wehenkel received the PhD degree and the "Agrégation de l'Enseignement Supérieur" from the University of Liège in 1990 and 1994, respectively, where he is Professor in the Department of Electrical Engineering and Computer Science. Dr. Wehenkel's research interests lie in the field of stochastic methods, in particular automatic learning and data mining, and power systems control, operation and planning.