Automatic Learning of Sequential Decision Strategies for Dynamic Security Assessment and Control

Louis Wehenkel, Mevludin Glavic, Pierre Geurts, Damien Ernst
Department of Electrical Engineering and Computer Science
University of Liège - Sart-Tilman B28 - B-4000 Liège
{L.Wehenkel,P.Geurts,dernst}@ulg.ac.be , glavic@montefiore.ulg.ac.be

Abstract - This paper proposes to formulate security control as a sequential decision making problem and presents new developments in automatic learning of sequential decision making strategies from simulations and/or information collected from real-life system measurements. The exploitation of these methods for the design of decision making strategies and control policies in the context of preventive and emergency mode dynamic security assessment and control is discussed and further opportunities for research in this area are highlighted.

Keywords - Optimal control, sequential decision making, dynamic security assessment and control, automatic learning.

1 INTRODUCTION

Power system security control aims at taking (manually or automatically) decisions and actions in order to prevent the system from entering a mode where widespread disturbances and service interruptions become unavoidable. Typically, security control is carried out in two complementary frameworks, namely so-called preventive security assessment and control, and emergency control. While in preventive mode operator expertise together with extensive analytical calculations may be combined in order to tackle an optimal tradeoff between safety and economy, in emergency mode time and information limitations imply that only simple, fast, well prepared control actions are taken in semi or fully automatic mode [1]. In both contexts the environment within which security control decisions must be taken is full of complexities and generally subject to a significant amount of uncertainties concerning system state, ongoing processes in remote parts of the interconnection, and dynamic behavior of all the local protections and automatic or manual controls interfering with preventive and emergency control actions. Considering the quickly increasing risks of blackouts and the wealth of available but totally underused tools and techniques, it is clear that today security control in the broad sense is generally tackled in a highly suboptimal way with respect to what could and should be achieved.

In particular, since the late sixties (starting with the work of Tom DyLiacco [2, 3]), automatic learning has repeatedly been advocated as an appropriate tool in order to help coping with complexity, uncertainties and time constraints in many environments and particularly in the context of preventive and emergency control for voltage and transient stability. During the subsequent decades the progress in computational means (both hardware and software) has been tremendous, and automatic learning has matured as a rich and highly productive research field, today one of the most active areas within computer science, systems theory, computational mathematics and statistics. Nevertheless, real-world power systems security related applications of automatic learning are almost unexisting, with the notable exception of a few European TSOs (RTE, NGT, ELIA) in the context of planning and operational planning [4] and the use at Hydro-Québec for pre-setting of special protection systems [5].

One objective of this paper is to provide a review some of the new developments in automatic learning based sequential decision making, and to explain their relevance in the context of power system security control applications. We first recall classical results from (stochastic) dynamic programming and (stochastic) optimal control theory (section 2) and explain why to use this framework for power system security control (section 3). We then revisit the basic principles of supervised and reinforcement learning as two complementary frameworks to design dynamic closed loop control algorithms and sequential decision policies (section 4) and explain how to apply them to security control in various contexts with more or less available information and time for decision making (section 5). We end the paper with a discussion of the real-life applicability of the proposed framework and tools and some hints for further research and developments (section 6).

2 SEQUENTIAL DECISION PROBLEM STATEMENT

We recall well-known results from (stochastic) dynamic programming and (stochastic) optimal control theories. Our objective is mainly to set the framework under which the remainder of the paper is shaped while stressing the similarities of the problems of designing a human agent's decision making strategy and an automatic control device's control algorithm. A much more detailed description, including the mathematical assumptions, can be found in [6], from which we also borrow the notation.

2.1 General stochastic system model

Let us consider a discrete time state description of a stochastic dynamic system

$$x_{k+1} = f_k(x_k, d_k, w_k), \qquad k = 0, 1, \ldots, h-1 \quad (1)$$

where

k indexes the discrete time,

x_k is called the system state at time k,
d_k is the control or decision variable applied at time k,
w_k is a random process (also called disturbance),
h denotes the temporal horizon of the problem.

Furthermore, we denote by X_k the state space at time k, i.e. the mathematical structure describing a set of possible states of our system model at time k, by $D_k(x_k)$ the set of allowed control decisions at time k (possibly a function of the state x_k). The random disturbance model is given in the form of a conditional probability distribution $P_k(\cdot|x_k, d_k)$ that may depend explicitly on the current state and control decision, but not on past values of w.

Supposing that the system is initially (at step $k = 0$) in a given state x, its trajectory is then defined by the following random process:

1. set $k = 0$ and $x_k = x$;
2. the control agent selects a control $d_k \in D_k(x_k)$, and a random experiment selects a value of w_k distributed according to $P_k(\cdot|x_k, d_k)$;
3. at time $k + 1$ the system moves to state $x_{k+1} = f_k(x_k, d_k, w_k) \in X_{k+1}$ according to its dynamics,
4. the process repeats itself $h - 1$ further times, by replacing the index k by $k + 1$ at stage 2, yielding a trajectory of h stages.

Notice that the notion of state used here is an extension of the classical notion of state used in deterministic systems theory. More specifically the information encoded in the state is such that, once x_k and d_k are given, the subsequent states are stochastically independent of previous states and controls, although they are not necessarily perfectly predictable. This is also called the *Markov property*, because if d_k depends only on x_k the sequence of states $x_0, \ldots, x_{h-1}$ forms a *Markov chain*.

The system is said to be *time-invariant* if the function f_k (and the sets X_k and D_k) and the probability distribution P_k do not depend explicitly on time, in which case we can drop the subscript k in our notation.

2.2 Performance criterion and candidate strategies

We consider an *additive over time* return-criterion. Namely, we define the return over an h-stages "trajectory" $(x_0, d_0, w_0, \ldots, x_{h-1}, d_{h-1}, w_{h-1}, x_h)$ by

$$J_h(x_0, d_0, \ldots, x_h) = \sum_{k=0}^{h-1} r_k(x_k, d_k, w_k). \qquad (2)$$

Notice that this performance criterion takes into account at each time step an instantaneous reward $r_k(x_k, d_k, w_k)$, which is potentially stochastic (dependence on the random variable w_k) and in general time-dependent.

Often, the decision making agent does not have full information about the system state. Rather, it can observe at each time instant a reduced and often noisy set of measurements $y_k = g_k(x_k, w_k)$. Hence, the problem is to define control signals based on this information in such a way that the return is maximized. Furthermore, the control policy has to be causal which means that the control applied at time k is not allowed to depend on information which has not yet been gathered at time k. Assuming that the controller has the possibility to observe the system at time k and to store this value for future usage, the most general class of control policies that makes sense (we call these the *admissible* policies) is defined by a vector π of conditional probability distributions

$$\pi_k(d_k|y_k, d_{k-1}, \ldots, d_0, y_0), \;\; k = 0, 1, \ldots, h-1. \quad (3)$$

The controller can use such a policy to make a random draw of the control d_k at time k depending in some way on its current knowledge.

Once such a control strategy has been selected, the distribution of N-stages trajectories starting from a given initial state $x_0 = x$ is well defined. Thus, the so-called expected return over h-stages

$$J_h^\pi(x) = E\{\sum_{k=0}^{h-1} r_k(x_k, d_k, w_k)\}, \qquad (4)$$

where the expectation is taken according to the distribution of trajectories starting from $x_0 = x$ and induced by the system dynamics f_k, observation equation g_k, noise distribution P_k, and choice of control policy π.

The solution of the optimal control problem consists of exploiting the knowledge of the system dynamics, observation equation, and return function so as to define an optimal policy π^*, i.e. an admissible policy such that for any initial state $x_0 = x$ and any admissible policy π, $J_h^{\pi^*}(x) \geq J_h^\pi(x)$.

We call this very broad class of controllers the *non-anticipating* controllers, to distinguish them from *open loop* ones which only use the information about y_0 and *closed loop* ones which only use information about y_k to select d_k. Let us notice that if the system is deterministic and fully observable (i.e. $w_k = const$ and $y_k = x_k$), optimality may be reached within the class of (deterministic) open loop policies. For fully observable stochastic systems optimality may be reached within the class of closed loop deterministic policies. Furthermore, if the system and reward function are time invariant and the optimization horizon tends towards infinity, these policies may be restricted to time-invariant ones.

3 SECURITY CONTROL AS A SEQUENTIAL DECISION PROBLEM

3.1 Preventive control

For preventive security assessment and control, the information that is monitored and exploited (i.e. y_k in our notation) is the estimated state vector and topology computed every few minutes from raw measurements by a state estimation software. Classical approaches to this problem, like security constrained OPF (SCOPF) or dynamic security assessment and control (DSAC) tools, treat this problem as a *static* optimization problem, termed in the following way:

Given state estimator outputs, static and dynamic system models, and a list of contingencies, find a set of preventive control actions such that the induced instantaneous cost (due to generation rescheduling, load-shedding...) is minimal and all current security constraints are satisfied (pre- and post-contingency ones).[1]

While this classical approach treats the tradeoff between security and induced costs in a static (i.e. instantaneous way), the actually incurred costs as well as the resulting security (or the risk of blackouts) are perceived by system users only averaged over a certain time horizon. Thus, it may well be justified to take at some time of the day a rather expensive control action if this is compensated later by cost-savings; conversely, it may well be justified to accept the violation of some (N-1) security constraints when control means are scarce and expensive, and to force some more (say N-2) constraints at another time of the day, if this could lead to an overall reduction of the risk of blackouts. Notice also that a preventive security control action taken at some time instant may influence the security level of the system and security control costs at later times of the day.

Hence, security control in preventive mode would actually be more appropriately modeled as a sequential decision making problem over some time horizon (daily, weekly, etc.) where the performance criterion is the cost of using control means combined with the risk of blackouts (the conditional expectation of costs implied by blackouts in a given state resulting from a given situation and control action) *integrated* over an appropriately chosen time-horizon. Moreover, this problem is highly stochastic, not only because of the fact that contingencies may occur with different probabilities at different times of the day, but also because the operator while deciding on preventive security control actions has only partial information about the state and dynamics of the whole interconnection and about the state in which the system will be at some later time.

In practice, the preventive security control problem is indeed seen (intuitively) as a sequential decision making problem in an uncertain environment. Thus, an expert operator would probably not shed load if he believes that the situation is improving and he would not reschedule generation to relieve a minor violation if the implied costs were too high. But, while it is expected from operators to use their expertise so as to tackle such tradeoffs related to the sequential and uncertain nature of the problem, to our best knowledge no security code acknowledges explicitly security as a sequential decision making problem in uncertain environment.

Consequently, the tools and procedures that are used in operation for preventive security control may be highly suboptimal from the point of view of either security or economy, or both.

3.2 Emergency control

Emergency control aims at correctly reacting to the occurrence of disturbances by (most often automatically) triggering control actions such as generation and load shedding or coordinated switching of topology. At this stage, the cost of control actions is generally not explicitly taken into account, since the main objective is to prevent or mitigate loss of load and generation that would incur without control; rather the objective is to minimize the amount of load and generation eventually disconnected.

Many emergency control systems operate in a one-shot open loop mode and are triggered upon the recognition of the initiating events. More sophisticated and slower ones are operating in closed loop mode, and rely on relatively local measurements to decide their control action. The currently used design approach of these emergency control systems consists of tuning and validating them based on deterministic assumptions about system behavior, which leads to control systems which are essentially unable to adapt their control strategy to highly disturbed or otherwise unusual system conditions. Thus, these control systems typically fail to reach their objective as soon as the actual system behavior (including the behavior of its protections) departs significantly from the assumptions made during design.

Emergency control actually tackles an optimal control problem (of minimizing the amount of unserved energy) with partial information about the system state and highly uncertain dynamics. The dynamics are uncertain because the system usually does not operate in disturbed conditions and hence knowledge about its behavior in these conditions is scarce.[2] Furthermore, closed loop emergency control is limited by the quality and quantity of information at its disposal, because time and investment costs limit the geographical extent, temporal resolution, and accuracy with which the system can be monitored.

Again, we notice that the design procedures of emergency control systems used in industry generally do not acknowledge explicitly the sequential and uncertain nature of the control problem they tackle.

3.3 Security control as a single problem

The optimal combination of preventive and emergency control is also a sequential decision problem in uncertain environment. Here, uncertainty is mainly related to the next contingency. The sequential nature stems from the tradeoff of incurring costs in preventive mode with expected costs related to unserved energy in emergency mode. This tradeoff has been considered by the academic power systems community in different ways, under the name of *probabilistic security* assessment [7]. But, to our best knowledge not much of this work has actually been applied in real-life.

[1] We simplify here by post-poning (see section 3.3) the discussion of the possibility to allow the violation of some post-contingency constraints if the tool used in preventive mode is able to determine a feasible corrective (emergency mode) control action for these contingencies. We also note that although this is the objective tackled by the classical approach to preventive security control, existing tools and practices are far from reaching it fully.

[2] People have used the term of 'hidden failures' to refer to situations where the system behaves in an unexpected way which is impossible to anticipate from its operation in normal conditions.

4 AUTOMATIC LEARNING OF SEQUENTIAL DECISION POLICIES

Our objective is to define methods which could provide approximations of optimal sequential decision policies from a reasonable amount of data. To simplify our discussion we suppose that the system state is fully observed (i.e. $y_k = x_k$). This ideal condition is seldom satisfied in practice, but the methods that we describe can be adapted to cope with partial observability.

4.1 Information provided to the learning agent

We suppose that the sole information available to the learning agent is in the form of a sample of N system trajectories over h stages:

$$\{(x_0^i, d_0^i, r_0^i, x_1^i, d_1^i, r_1^i, \ldots, x_{h-1}^i, d_{h-1}^i, r_{h-1}^i, x_h^i)\}_{i=1}^N.$$

Each trajectory i provides information about the state x_t^i, decision d_t^i and instantaneous reward r_t^i obtained at each time step, as well as the terminal state reached. We next discuss how such samples can be exploited in order to extract from them an approximation of an optimal decision strategy.

4.2 Supervised learning from optimal decisions

Supervised learning is usually defined as follows [8]:

> *Given a training set of input/output pairs, determine a function (or model, or algorithm) to compute the outputs given the inputs which not only is accurate on the training set, but also generalizes well to unseen cases.*
>
> *If the output variable is discrete, one talks about classification, if it is numerical one talks about regression.*

In the context of supervised learning, it is thus assumed that there is a teacher which provides the learning agent with examples of correct decisions (outputs) for a representative set of states (inputs). The learning agent has then to generalize this information to unseen situations. In our context, this assumption corresponds to the case where the decisions d_t^i taken in the sample trajectories are all optimal, i.e.

$$\forall i = 1, \ldots, N; \forall t = 0, \ldots, h-1 : d_t^i = d^*(x_t^i, t).$$

Supervised learning can then be directly applied in order to derive from the original sample h decision rules approximating the h optimal decision functions $d^*(\cdot, t), \forall t = 0, \ldots, h-1$. More precisely, to extract $\hat{d}^*(\cdot, t)$ the supervised learning algorithm receives the sample

$$\{(x_t^i, d_t^i)\}_{i=1}^N.$$

In some problems, it is necessary to cope with situations where the inputs are not providing complete information about the outputs. This would be the case, for example, when the teacher uses complete information about the system state to determine its control actions while the learning agent has only access to partial information in the form of local measurements. In that case, the supervised learning algorithm will extract an approximate decision rule that will target the best guess for d^* given the inputs.

4.3 Reinforcement learning from arbitrary decisions

Supervised learning alone is unable to learn a good decision policy from a set of system trajectories strongly corrupted by erroneous (i.e. suboptimal) decisions. Thus, it is not sufficient when addressing a sequential decision problem for which there is no experience yet about the optimal way of solving it or when there is no alternative way to determine what is an optimal decision in a given context. In such situations, reinforcement learning should be used to extract approximations of optimal decision strategies [9, 10].

In essence, reinforcement learning exploits the sample of trajectories to extract from it approximations of the reward functions r_k and system dynamics f_k and combines these using the dynamic programming principle (see [6]) in order to derive an approximation of the sequence of optimal policies $d^*(\cdot, t)$.

In particular, the algorithm developed in [11, 12] uses a supervised learning method (for regression) to determine approximations (and generalizations) of the so-called Q_t-functions in an iterative fashion. By definition, the function $Q_t(x, d)$ computes the expected reward over t remaining stages when at time $h - t$ the system is in state x, decision d is taken and subsequently (over the remaining $h - t - 1$ stages) the optimal decision policy is used. Obviously, from $Q_t(x, d)$ it is possible to directly extract the optimal decision strategy at time t by

$$d^*(x, t) = \arg \max_{d \in D_t(x)} Q_t(x, d).$$

The dynamic programming principle implies the Bellman equation

$$Q_t(x, d) = E\{r_{h-t} + \max_{d'} Q_{t-1}(x_{h-t+1}, d')\},$$

where the expectation is taken under the condition that $x_{h-t} = x$ and $d_{h-t} = d$.

4.3.1 Fitted Q iteration algorithm

Fitted Q iteration exploits batch-mode supervised learning to yield a sequence of approximate Q_t-functions from a sample of trajectories in the following way:

- Initialization: Set $t = 0$ and $\hat{Q}_0(x, d) \equiv 0$.
- Basic iteration:
  - Set $t = t + 1$
  - Create a learning sample $ls_t = \{(\text{in}_t^i, \text{out}_t^i)\}_{i=1}^N$ of input/output pairs, where $\text{in}_t^i = x_{h-t}^i$ and $\text{out}_t^i = r_{h-t}^i + \max_d \hat{Q}_{t-1}(x_{h-t+1}^i, d)$.
  - Apply a supervised learning algorithm to build $\hat{Q}_t(x, u)$ from the learning sample ls_t.
- Finalization: if $t = h$ extract the approximate optimal decision strategies $\hat{d}^*(\cdot, t)$ from the sequence of $\hat{Q}_t$ functions by

$$\hat{d}^*(x, t) = \arg \max_d \hat{Q}_{h-t}(x, d).$$

4.3.2 Time-invariant case

If the system and reward function are time invariant, then the fitted Q iteration algorithm (and in general, any reinforcement learning algorithm) can take better advantage of the available sample of system trajectories. Indeed, in this case one can use at each iteration t a sample derived from all the system transitions

$$\{(\text{in}^k, \text{out}^k)\}_{k=1}^{Nh} = \bigcup_{t'=1}^{h} ls_{t',t},$$

where $ls_{t',t} = \{(\text{in}^i_{t',t}, \text{out}^i_{t',t})\}_{i=1}^{N}$, $\text{in}^i_{t',t} = x^i_{h-t'}$ and $\text{out}^i_{t',t} = r^i_{h-t'} + \max_d \hat{Q}_{t-1}(x^i_{h-t'+1}, d)$.

Time-invariance thus allows to make a more effective use of the information available in the sample of system trajectories.

4.3.3 Infinite horizon case

If the system is time-invariant and the reward function is in the form $r_k(x, d, w) = \gamma^k r(x, d, w)$ with $\gamma \in (0, 1)$, an approximation of an optimal time-invariant decision strategy may be obtained by a slightly modified version of the fitted Q iteration algorithm, where at each stage the outputs are defined by

$$\text{out}^i = r^i_{h-t} + \gamma \max_d Q_{t-1}(x^i_{h-t+1}, d),$$

and by approximating the infinite horizon policy by

$$\hat{d}^*(x, 0) = \arg\max_d \hat{Q}_H(x, d),$$

where H is sufficiently large to ensure convergence to the fixed point of the Bellman equation.

4.3.4 Supervised learning algorithm

In principle, the fitted Q iteration algorithm can be combined with any supervised learning algorithm. However, the quality of the resulting reinforcement learning strongly depends on the characteristics of the used supervised learning method. In particular, the class of so-called kernel-methods have interesting convergence and consistency properties [13]. An interesting sub-class of these latter uses so-called ensembles of randomized trees [14] which are able to handle efficiently large scale problems in a totally autonomous way, and are therefore very well suited to the reinforcement learning context [12].

5 APPLICATION TO SECURITY CONTROL

The basic assumption in the reinforcement learning framework is that the database is representative in the sense that the (x_k)-parts are representative of the interesting regions of the state-space, that the (d_k)-parts are sufficiently diverse to allow the identification of optimal control actions from the corresponding instantaneous rewards and successor states r_k and x_{k+1}. If the system is non-deterministic, then these latter should also be conditionally representative of the distribution of successor states and rewards for a given state-action pair. In practice this implies that the database size (number of system transitions) needed in this protocol is significantly larger than in the case of the supervised learning protocol (where, however, we need to be sure that the (d_k)-parts are the optimal actions for the corresponding state x_k).

In the context of power system modeling and control problems, reinforcement learning can be used to exploit information obtained from different contexts:

- *Learning from a power system simulator:* it is often interesting, although not strictly necessary, to couple the learning agent with the simulator so as to exploit the result of learning in order to influence the way the subsequent four-tuples are generated. Thus the learning from simulations can be totally autonomous (see e.g. [15, 16] for some examples) or it can take advantage of some "teaching" mechanism which chooses in some way the most interesting simulation scenarios, so as to speed up learning as much as possible.
- *Learning from an actual controlling agent:* it is also possible to exploit in reinforcement learning the information gathered from a real system monitoring. For example, one could collect information about the control decisions taken by a human operator or by an existing automatic control device (together with rewards and successor states) and then use a reinforcement learning agent to learn a control policy from this information. The resulting policy may in principle outperform the original controlling agent. In this way it is also possible to collect information from several suboptimal controllers of a system and inject it into the learning agent.
- *On-line learning:* finally, it is possible even to couple the reinforcement learning agent directly with a real system, provided that safeguards are imposed in order to avoid that the agent (initially far from an optimal controller) creates catastrophic situations.

Of course, data can also be collected from any combination of these contexts. Even, in the context of uncertain system dynamics one can generate simulated data under different modeling hypotheses and inject them into a batch mode reinforcement learning agent to infer a robust controller.

We refer the interested reader to some of our recent publications concerning the applications of reinforcement learning to power system control [16, 17, 18, 19].

5.1 Adaptive and distributed control

In the context of on-line learning, a reinforcement learning agent continuously collects four-tuples at each time step and can infer from the associated rewards and successor states information about the real system performance and adapt its control policy to it. Furthermore, if there are several control agents using reinforcement learning connected to a single system, they can learn all in parallel and each one of them can adapt its performance progressively, hopefully leading to some kind of coordinated distributed control. If possible, it is of course advised to pre-train such multi-agent systems using an off-line simulator before plugging them on a real system.

6 DISCUSSION

In this paper we revisited security control in order to highlight the sequential nature of this problem and stress the fact that optimal security control decisions need to be taken in spite of many sources of uncertainty. We have then explained how the combination of supervised learning and reinforcement learning protocols can be used within the framework of stochastic dynamic programming to learn from samples of recorded system trajectories approximations of optimal sequential decision strategies, within a wide class of possible conditions. Indeed, depending on the quality of information supervised learning could be applied directly to the available data only if decisions shown in the examples are close to optimal. On the other hand, combined with reinforcement learning it can be applied also if the data is corrupted by suboptimal decisions. We have also stressed the fact that during the last twenty years machine learning has made significant progress and become a mature discipline offering a broad class of principles, methods and algorithms that could handle many complex and large scale problems that need to be solved for designing robust and near-optimal security control algorithms.

Although automatic learning was proposed already in the late sixties for security control, its penetration in the industry practice is still very low. The well recognized excellence of power system engineers in exploiting advanced methods from applied mathematics, control theory and computer science to solve engineering problems has indeed been shadowed during the last fifteen years by the ongoing restructuring processes. While Internet and Multimedia companies have extensively exploited in their business large scale computational infrastructures where thousands of PCs are used to speed up search engines and computer animation software, the Power Industry today uses comparatively very modest amounts of computers and obsolete techniques to handle the more and more difficult tradeoff between security and economy.

Clearly, automatic learning has a very high potential in power systems security assessment. Indeed, it offers the required methodology and tools able to exploit data collected from field measurements and from simulations based on analytical methods, such as time domain simulation, contingency analysis and optimal power flow.

However, without significant efforts to scale up the computational infrastructure and practices of our industry, automatic learning and other analytical tools developed in the last twenty years will remain on the shelf of Universities for many more years.

In the meanwhile, the researchers who have developed these tools are turning to other application fields where the chances of using their work are higher.

ACKNOWLEDGMENTS

Damien Ernst and Pierre Geurts acknowledge the support of the Belgian FNRS (Fonds National de la Recherche Scientifique) where they are post-doctoral researchers.

REFERENCES

[1] L. Wehenkel, "Emergency control and its strategies," in *Proceedings of the 13th Power Systems Computation Conference, PSCC99*, June 1999, pp. 35–48.

[2] T. E. DyLiacco, "The adaptive reliability control system," *IEEE Transactions on Power Apparatus and Systems*, vol. 86, no. 5, pp. 517–531, 1967.

[3] ——, "Control of power systems via the multi-level concept," Ph.D. dissertation, Case Western Reserve University, Systems Research Center, 1968.

[4] S. Henry, J. Pompée, J. Paul, M.Béna, and K. Bell, "New trends for the assessment of power system security under uncertainty," in *Proceedings of Bulk Power System Dynamics and Control - VI, IREP04*, 2004, pp. 679–684.

[5] J. Huang, S. Harrison, G. Vanier, A. Valette, and L. Wehenkel, "Application of data mining techniques for settings of generator tripping and load shedding system in emergency control at hydro-québec," in *Proceedings IEEE PES Winter Meeting*, 2003, p. 6.

[6] D. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000, vol. I.

[7] R. Marceau and J. Endreyeni, "Power system security assessment," *Electra*, November 1997.

[8] L. Wehenkel, *Automatic Learning Techniques in Power Systems*. Kluwer Academic, 1998.

[9] L. Kaelbling, M. Littman, and A. Moore, "Reinforcement Learning: a Survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[10] R. Sutton and A. Barto, *Reinforcement Learning. An Introduction*. MIT Press, 1998.

[11] D. Ernst, P. Geurts, and L. Wehenkel, "Iteratively extending time horizon reinforcement learning," in *Proc. of the 14th European Conference on Machine Learning*, Dubrovnik, Croatia, September 2003.

[12] ——, "Tree-based batch mode reinforcement learning," *Journal of achine Learning Research*, vol. 6, pp. 503–556, April 2005.

[13] D. Ormoneit and S. Sen, "Kernel-Based Reinforcement Learning," *Machine Learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. (to appear), pp. 1 – 39, 2006.

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

[16] D. Ernst, M. Glavic, and L. Wehenkel, "Power system stability control: reinforcement learning framework," *IEEE Transactions on Power Systems*, vol. 19, no. 1, pp. 427–435, February 2004.

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

[18] M. Glavic, D. Ernst, and L. Wehenkel, "Combining a stability and a performance oriented control in power systems," *IEEE Trans. on Power Systems*, vol. 20, no. 1, 2005.

[19] D. Ernst, M. Glavic, P. Geurts, and L. Wehenkel, "Approximate value iteration in the reinforcement learning context. Application to electrical power system control," *International Journal of Emerging Electric Power Systems*, vol. 3, pp. Issue 1, Article 1066, 2005, 30 pages.