CONTROL OF A POWER PLANT USING MDP AND POMDP

Mario E. Agueda*
Instituto de Investigaciones Eléctricas
Av. Reforma 113, Palmira
Temixco, Mor., 62490, México
55058mea@iie.org.mx

Pablo H. Ibargüengoytia
Instituto de Investigaciones Eléctricas
Av. Reforma 113, Palmira,
Temixco, Mor., 62490, México
pibar@iie.org.mx

Abstract

This paper presents experiments in the use of Markov Decision Processes (MDP) and Partially Observed MDP (POMDP) for guiding a classical controller (PID) in a critical loop of a combined cycle power plant. The graphics of the optimal trajectory of control are discretized in order to form a finite set of states in the space. The actions are the increments or decrements of the variables that produce the movements in the control space. The transition matrices are obtained with real data from different operation conditions of the plant. The reward functions are obtained from experimented operators of power plants. A software architecture, where the interaction between the knowledge, decision, and action modules is described. Experiments to control the level of water of the drum in a heat recovery steam generator (HRSG) of a combined cycle power plant are presented.

1 Introduction

The use of artificial intelligence (AI) techniques in real domains is more common every day. However, there still are some challenges that need to be tackled. For example, uncertainty management is required since models, knowledge and data from industrial applications are approximate, incomplete, and unreliable. Real time behavior is also required, especially in applications where continuous processes are being observed. Additionally, another challenge that this research group has faced is to understand the distinction between the problems that AI techniques can solve, and problems where traditional techniques are still better options. For example, the control of temperature in a turbine of a fossil power plant is carried out with a *proportional, integrative and derivative* or PID control. However, the coordinated control between gas and steam turbines, to obtain a nominal power, requires the use of optimization techniques that AI has proven to give better results.

This research group has been working with experiments about the applications of AI techniques in the generation of electricity processes. The experiments presented in this paper focused in the support to the control of the water level in a special tank of the plant called the drum. This control problem can be seen as a stochastic domain given the thermodynamic conditions that are present in that process.

Markov Decision Process (MDP) ia a tool that can be used to get optimal actions policies for this *stochastic* domain. Additionally, Partially Observable Markov Decision Processes (POMDP), are used under domains where there is not certainty about the actual state of the system, i.e, it is a partially observable domain. The agent can do observations from which, it computes a probabilistic distribution of states where the system can be. Based on this position, the agent designs the optimal sequence of actions to reach a specific goal. Both tools are typically used in robot navigation control, even though they can also be utilized in control and planning problems.

In order to use these tools in the control of water level of the drum of a power plant, some modules are necessary. First, a knowledge module that manages the transition, the reward and the observation matrices is required. These matrices can be established once that a finite set of actions on each state and a finite set of observations are defined. Second, a decision process module is required where the MDP and POMDP algorithms are implemented and interfaced with the knowledge and actors modules. Third, an actor module is the direct interface with the environment. In this case, the steam generator of a power plant is the environ-

* CINVESTAV Unit Guadalajara, Guadalajara, Jalisco, México.

ment and the operator is the actor. The actor provides the goals required and executes the commands that modify the environment. Finally, a learning module allows the system to refine the knowledge about the application domain, given the difficult thermodynamic conditions of the process. All these modules are integrated in a software architecture presented in this paper.

This paper is organized as follows. Section 2 introduces the MDP and POMDP techniques. Section 3 describes the basics of the application domain, namely the operation of a combined cycle power plant and specifically, the steam generator. Next, section 4 describes the adaptation of the application domain elements to the different sets and matrices required by the MDP and POMDP. Section 5 describes the proposed architecture for the implementation of the MDP and POMDPs for the generation of policies in the optimum control of the generation plant. Section 6 presents the experiments than were carried out and discusses the main results. Section 7 concludes the paper and explain the future work in this area.

2 Markov Decision Processes

Markov decision processes are utilized as a stochastic model of an agent that is interacting with the world. Given an action executed in a known state of the world, it is possible to calculate the probability of the next state in the world. The probability of reaching a state q' when the action α has occurred is calculated by the summation of the conditional probability of reaching q' from every possible state q_i of the world given the action a.

A formal description of a MDP is the tuple (S, A, T, R) where:

- S is a finite set of states of the world.
- A is a finite set of actions that the agent can do.
- $T : S \times A \rightarrow P$ is the transition function of states. For each action and state of the world, there is a probabilistic distribution over the states of the world that can be reached after the actions. The function $T(s \mid s', a)$ is defined as the probability of reaching state s starting in state s' and given the action a.
- $R : S \times A \rightarrow \Re$ is a reward function. For each action in each state of the world, it assigns a real number. The function $r(s, a)$ is defined as the reward of executing action a in the state s.

The MDP technique provides a policy of actions such that the sum of rewards obtained when the actions are executed is maximal. The following definitions establish the equations for the MDP algorithm.

Definition 1 : *non-stationary policy. A policy π is non-stationary iff π is a sequence of situation-action mappings, indexed by time, i.e., the optimal sequence of actions may change when time advances.*

This is an obvious property of dynamic systems where the occurrence of events, change the old beliefs about the state of the system.

Definition 2 : *non-stationary expected discounted sum of reward. Let $V_{\pi,t}$ be the expected sum of rewards gained executing non-stationary policy π for t steps. The cumulative reward $V_{\pi,t}$ starting in state s is defined by:*

$$V_{\pi,t}(s) = \left[R(s, \pi_t(s)) + \gamma \sum_{s' \in S} T(s' \mid s, \pi_t(s)) V_{\pi,t-1}(s') \right]$$

where γ $(0 \leq \gamma < 1)$ is a discount factor, and $\pi_t(s)$ is the action over state s in step t in the policy π.

The discount factor γ is required to consider a $infinite - horizon$ discounted model [Kaelbling et al. 1998] in which the number of iteration are in principle, infinite. However, the discount factor γ provokes the convergence on a polynomial number of steps [Givan et al. 2000].

The maximal expected discounted sum of future reward, starting in state s, can be computed as follows:

$$V_t(s) = max_{a \in A(s)} \left[R(s, a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V_{t-1}(s') \right]$$

Definition 3 : *Optimal policy calculation. Given any V function, optimal policy π_V is computed as follows:*

$$\pi_V(s) = argmax_{a \in A(s)} \left[R(s, a) + \gamma \sum_{s' \in S} T(s' \mid s, a) V(s') \right]$$

These three equations: the calculus of a cumulative reward, the calculus of future reward, and the optimal policy calculation, form the basis of the MDP algorithm [Kaelbling et al. 1998] implemented in this project.

2.1 Partially Observable MDPs (POMDPs)

MDPs provide an optimal policy for solving a specific problem if the state of the world is known in every step of the policy. However, in some stochastic models, it is difficult to know with certainty the actual state of the world. The tool utilized to estimate the state of the world at certain time is the Partially Observable MDP or POMDP. The POMDPs provide the formula to determine the probability of every actual state.

A formal description of a POMDP is the tuple (S, A, T, R, Θ, O) where:

- S, A, T, R are as defined for MDP.
- Θ is a finite set of observations that the agent makes about the world.
- $O : \Theta \times S \rightarrow P$ is the observation function. In each state, it calculates the probability of making an observation. The function $O(o \mid s)$ is defined as the probability of making an observation $o \in \Theta$ over state $s \in S$.

POMDPs provide a probabilistic distribution E over the states of the world, i.e. it represents the probability of being in each state $s \in S$. E is modified after an action $a \in A$ has been done or after an observation has been done. Let $E'(s)$ be the probability of staying in state s, and let $E(s)$ be the probability of reaching the state s. This probability is calculated considering all the probabilities of arriving to the state s from the rest of the states given an action a or given a new observation of the process. In the case that E is modified by an action, $E(s)$ is given by:

$$E(s) = \frac{1}{\beta} \sum_{s' \in S} T(s \mid s', a) E'(s')$$

where β is a normalization constant.

After making an observation $o \in \Theta$, the probability distribution $E(s)$ considers the conditional probability of making this observation over the state s. In the case that E is modified by an observation, $E(s)$ is given by:

$$E(s) = \frac{1}{\beta} O(o \mid s) E'(s)$$

These two equations for the calculus of the probabilistic distribution at a given time, form the basis of the POMDP algorithm [Kaelbling et al. 1998] implemented in this project. Both algorithms, MDP and POMDP are utilized in this work for the guide of a classic controller in a loop of a combined cycle power plant. For more information about these algorithms, refer to [White 1993] and [Puterman 1995].

The next section briefly describes this application domain.

3 Control of Power Plants

This section explains the basis of the problem tackled by this work, namely the guiding of the water level control of the drum in a combined cycle power plant.

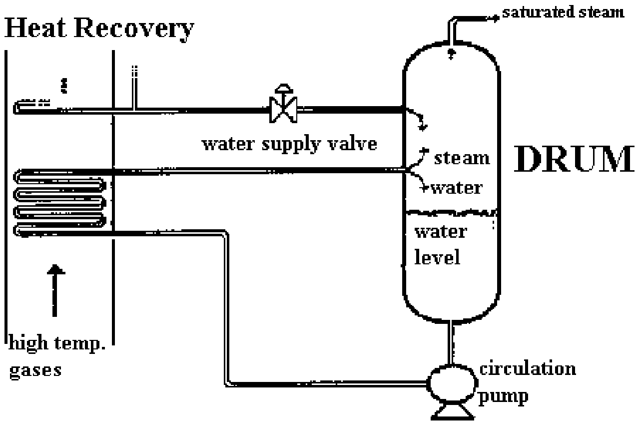

Figure 1: Simplified diagram of the heat recovery.

A combined cycle power plant consists mainly of four equipments: the gas turbine, the heat recovery steam generator (HRSG), the steam turbine and the generators. There is usually one generator per turbine. The gas turbine receives air and gas in a specific proportion to produce high pressure gases at high temperature. These gases produce the rotation that moves the generator corresponding to this turbine. After producing mechanical work, the gases are still at very high temperature, so that energy can still be exploited. The gases are introduced to the heat recovery that, sometimes with the assistance of extra burners called after burners, produces steam from the feed-water system. After the steam is superheated, it is introduced to the steam turbine to convert the energy carried out by the steam in work and finally, in electricity through the corresponding steam generator. This plant is called combined cycle since there are two types of generators combined to produce the complete unit power.

Figure 1 shows the basics of the HRSG and the water cycle. The heat recovery steam generator consists of a huge boiler with an array of tubes, the drum and the circulation pump. The feed-water flows through the tubes receiving the heat provided by the gases from the gas turbine and the burners. Part of the water mass becomes steam and is separated from the water in a special tank called the drum. Here, water is stored to provide the boiler tubes with the appropriate volume of liquid that will be evaporated and steam is stored to be sent to the steam turbine.

From the drum, water is supplied to the rising water tubes called water walls by means of the water circulating pump, where it will be evaporated, and water-steam mixture reaches the drum. From here, steam is supplied to the steam turbine. The conversion of liquid water to steam is carried out at a specific saturation condition of pressure and temperature. In this condition, water and saturated steam are at the same temperature. This must be the stable condition where the volume of water supply is commanded by the feed-water control system. Also, the valves that permit the steam supply to the turbine are controlled in order to manipulate the values of pressure in the drum.

The level of the drum is one of the more important variables in the generation process. A decrease of the level may cause that not enough water is supplied to the rising tubes and the excess of heat and lack of cooling water may destroy the tubes. On the contrary, an excess of level in the drum may drag water as humidity in the steam provided to the turbine and cause severe damage to the blades. In both cases, a degradation of the performance of the generation cycle is experimented.

Even with a very well calibrated instrument, the level control of the drum is one of the more complicated and uncertain tasks of the whole generation system. This is because the mixture of steam and water makes very difficult the reading of the exact level of mass. Sometimes, the level sensor indicates a high level but in reality, what happens is that pressure decreases for some reason and the saturation condition changes. This causes that an amount of water mass evaporates instantaneously in the rising tubes, pushing water to the drum and creating the impression that the level is high. The control system may respond closing the valves of water supply to the cycle when in reality, the required action is to increase the volume of liquid water in order to reach the new saturation condition. Expert operators driving the process in manual mode recognize this false level reading and decide the control actions that apply to this case.

Next section explains how the MDPs can model this expert knowledge and guide the control response in case of these uncertain events.

4 Uncertainty in the Control

Unfortunately, changes in the pressure conditions of the HRSG are common events, especially when changes in the load are required. This produces a conflictive situation for the PID control of the drum that commands the aperture of both valves: (i) water supply to the cycle, and (ii) steam supply to the steam turbine. Figure 2 shows the required curve of operation relating the flow of main steam f_{ms} and the pressure in the drum p_d under different conditions of required load [C.F.E. n.d.].

The specific control problem faced in this situation is to follow the recommended curve, given a certain load. This is in fact an uncertain process given the thermodynamic conditions of the process described above. If the load requirement changes, then a navigation in the space shown in Fig. 2 is required. This view of the problem is similar to the robot navigation. However, different trajectories can be implemented and hence, different actions are required. For example, if are mapped at the point **26**, and the requirement is to move to point **21**, then two of the possible trajectories are the following. First, open the principal steam valve in order to increase f_{ms} and then decrease the pressure p_d by decreasing the temperature. Second, decrease first p_d and then increment f_{ms}. Again, given the thermodynamic conditions, and according to the history of those movements in specific equipment, the two different policies may have different effects to the ones expected.

According to section 2, the utilization of MDP requires the following four sets:

a finite set of states: they are obtained with the discretization of both variables (flow of main steam and pressure in the drum), and forming a grid as shown in Fig. 2.

a finite set of actions: they are: an increment in the variable, a decrement or no move at all. This discretization of the actions utilized in this experiments was chosen to maintain the problem simple.

the transition functions of states: they are obtained with real data taken from the plant in different conditions, and in two steps. First, take f_{ms} as the independent variable and analyze what is the resultant action of p_d when f_{ms} changes. Notice that the action can only be increment, decrement or stay. Second, p_d is the independent and analyze the actions in f_{ms}.

the reward functions: They are assigned by an human expert operator with experience in the operation of the HRSG. Real data can also be utilized to detect registered successful transitions and assign these reward functions.

Table 1 shows an example of the set of actions that are considered in this application. $\Delta flow$ and $\Delta Pressure$ represent an increment (+) or decrement (-) or no change (0) in the variables pressure and flow of steam.

Thus, an example of the transition matrix corresponding to action a_1 is shown in table 2. The element $t_{i,j}$ represents the probability of the transition from state i to state j given the action a_1, i.e., an increment in the pressure. The states i and j refer to the defined states represented in Fig. 2. They are taken from real data from the plant as follows: First, a selection was

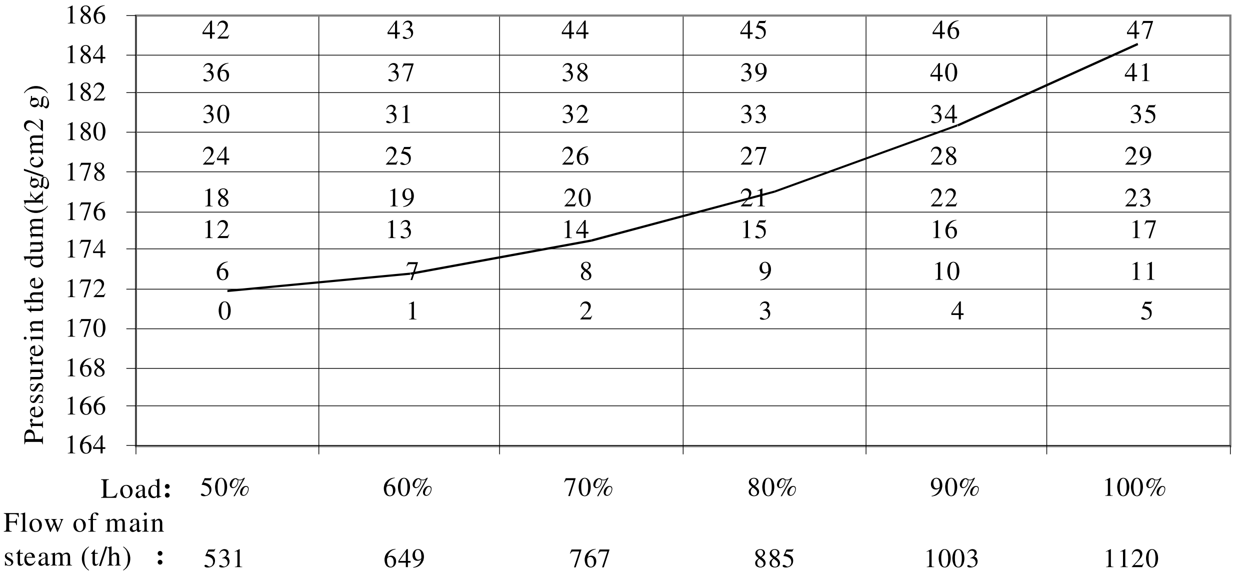

Figure 2: Curve of recommended operation of the plant with respect to pressure and flow of steam.

Table 1: Set of actions in this application.

Action	Δp_d	Δf_{ms}
a_1	+	0
a_2	-	0
a_3	0	+
a_4	0	-
a_5	0	0

Table 2: Transition matrix corresponding to action a_1.

$t_{15,23} = 0.17$		
$t_{15,22} = 0.67$	$t_{16,22} = 0.70$	$t_{8,22} = 0.50$
$t_{15,28} = 0.08$	$t_{16,29} = 0.10$	$t_{8,16} = 0.25$
$t_{15,21} = 0.08$	$t_{16,23} = 0.20$	$t_{8,15} = 0.25$
$t_{1,7} = 0.33$	$t_{7,14} = 0.50$	
$t_{1,8} = 0.67$	$t_{7,16} = 0.50$	$t_{14,22} = 1.00$
$t_{0,15} = 1.00$	$t_{22,29} = 1.00$	$t_{17,22} = 1.00$

made, from all the samples where the pressure was incremented, comparing two contiguous samples. Second, localization was made of the initial state number and the destination state number. Notice that in some instnces, e.g., $t_{0,15} = 1$, very few cases were available, so the probability distribution is not very accurate.

Given those sets of elements that convert a control problem in a MDP planning problem, a control schema is required. Figure 3 shows the assumed block diagram. It was designed in this form in order to achieve one of the goals of this project, namely the optimization of the operation of a PID controller. The PID

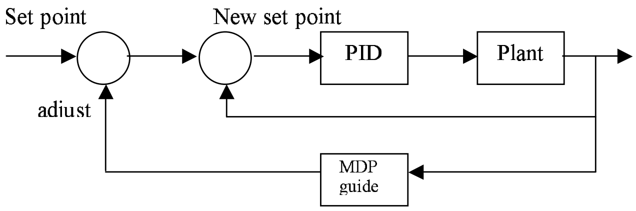

Figure 3: Proposed controller for drum water level.

control receives the feedback of three elements: the drum water level value, the flow of steam and the pressure, and of course the set up value. It calculates the error and issues the necessary commands for the modification of p_d and f_{ms} values to obtain a smaller error in the water level. The set point is established by the distributed control system or by the operator in manual mode.

The signals of the state of the process are also observed by the intelligent guide controller system that

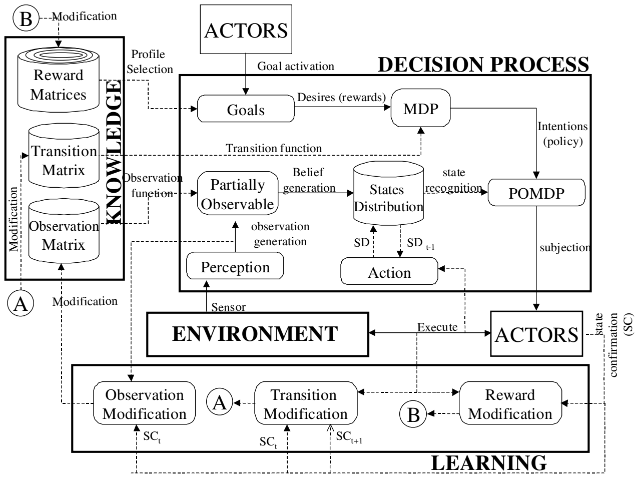

Figure 4: Proposed controller for the drum water level.

designs an optimal policy. The response of the system is an adjustment of the programmed set point, to form a new set point that uses the capabilities of the PID controller and the optimization techniques provided by the MDPs. This adjustment corresponds to the actions indicated in the policy. For example, if the policy indicates the action $a_2 = decrement\ p_d$, then a command to close the fuel valve is necessary in order to decrement the temperature and hence, the pressure in the drum. Notice that an extra mapping between the actions indicated in the policy and the commands of operation in the plant might be necessary. Specifically, a mapping between an action and the adjustment of the corresponding set point has to be defined. This paper only considers the generation of plans as a sequence of actions.

5 The Architecture

This section describes the software architecture implemented for the control of the drum in the power plant. Figure 4 shows the proposed architecture.

The **environment** in this work is precisely the power plant, and more specifically, the HRSG. The plant provides the signals that are necessary for the perception module and it receives the input commands that the actor issues. In this case, the **actors** can be the operators or the distributed control system. The actor also has the capability of changing the goals that will direct the policies designed by the decision process module. The signals that provide the environment are the flow of main steam, the flow of circulating water, the pressure in the drum, and the load values in mega watts. The commands that receive the environment are the movements of the valves that produce increments and decrements in the flow and pressure. Notice that the **learning** module also registers the commands executed by the actors. The learning module utilizes all the information that processes the environment and executes modification commands to the matrices that conform the **knowledge base**. Specifically, the reward matrices and the transition matrix are modified.

The **decision process** contains the main modules of the architecture. This module obtains the optimal policy of actions of the system, given the goal desired. The **goals** module holds the goal chosen by the actor and it chooses a reward matrix from the knowledge base (profile selection). The goals module sends this matrix as the desires to the MDP module. The **perception** module receives the sensor's information and generates an observation given by a combination of values in the state variables. The **partially observable** module receives the observation from the perception and uses the observation matrix in order to generate a belief given by a probability distribution of the current state of the system. This is stored in the **states distribution** module that performs the state recognition. The **MDP** module performs the Markov decision process algorithm. The set of states and actions were initially defined, so the MDP module receives the reward function and the transition function and calculates the optimal policy of actions assuming the different states of the world. This policy is represented in the form of intentions. The intentions are sent to the **POMDP** module that, together with the state recognition, defines the most probable current state and chooses the optimal action at this time. The set of actions given to the actor constitutes the exact sequence of actions that it must perform.

6 Experiments

This section describes the advances in the development of this project and presents the main results. Given the difficulty of showing all the partial results of a complete experiment, the description of one simulation of the prototype is given. First, the description of the transition matrices of the main actions is presented. Next, the matrix of rewards is shown, obtained from expert knowledge in the operation of power plants. Finally, the result of the optimal policy, defined for a certain goal is given. The experiments were executed utilizing real data from 260 hours of operation of a thermoelectrical power plant in México.

Figures 2 and 6 show the navigation space corresponding to the recommended operation conditions of the HRSG. The optimal curve was obtained from the manufacturer of the boiler. The learning module conforms the transition matrices for the five actions considered utilizing real data. Figure 5 shows the resultant five

Action a_1

	8	9	10	14	15	16	20	21	22
8	0	0	0	0.14	0.14	0.29	0	0.14	0.29
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0.33	0.64
15	0	0	0	0	0	0	0	0.21	0.79
16	0	0	0	0	0	0.18	0	0.82	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	1	0
22	0	0	0	0	0	0	0	0	1

Action a_2

	8	9	10	14	15	16	20	21	22
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0.5	0.5	0	0	0	0
22	0.12	0	0	0.08	0.48	0.32	0	0	0

Action a_3

	8	9	10	14	15	16	20	21	22
8	0	0	0	0	0.17	0.33	0	0.17	0.33
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0.33	0.17	0	0.17	0.33
15	0	0	0	0	0	0.29	0	0	0.71
16	0	0	0	0	0	0.8	0	0	0.2
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0.13	0.87
22	0	0	0	0	0	0	0	0	0

Action a_4

	8	9	10	14	15	16	20	21	22
8	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0.75	0	0	0.25	0	0	0	0	0
16	0.29	0	0	0	0.71	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	1	0	0	0	0	0
22	0.13	0	0	0.09	0.55	0	0	0.23	0

Action a_5

	8	9	10	14	15	16	20	21	22
8	1	0	0	0	0	0	0	0	0
9	0	1	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0
14	0	0	0	1	0	0	0	0	0
15	0	0	0	0	1	0	0	0	0
16	0	0	0	0	0	1	0	0	0
20	0	0	0	0	0	0	1	0	0
21	0	0	0	0	0	0	0	1	0
22	0	0	0	0	0	0	0	0	1

Rewards

	a1	a2	a3	a4	a5
8	0	-3	-1	-3	-1
9	3	-4	-6	-1	-2
10	2	-7	-5	3	-3
14	-4	-3	2	-5	0
15	5	-5	-3	-1	-1
16	0	-3	-4	3	-2
20	-8	2	5	-5	-3
21	-5	-4	-7	-4	10
22	-1	-3	-4	5	-2

Figure 5: Transition matrices for the five actions and the reward matrix.

transition matrices of the five actions considered, and the reward matrix considering the goal in state 21 of Fig. 2. This point corresponds to optimal values of pressure and flow when 80% of load is required.

Since most of the 260 data samples considered in this experiments are concentrated in one area, the experiments were also concentrated in the states that form the rectangle marked in Fig. 6, i.e., states 20 - 22, 14 - 16, and 8 - 10. This is the reason of the five matrices of size 9×9. The reward matrix was obtained with expert knowledge about the secure and optimal operation of the power plant. For example, it is always dangerous to maintain the operation conditions above the curve because higher pressure is a risky condition that may cause an accident or the decrement of lifetime of the equipment of the plant. On the contrary, the states below the curve are secure but they cause lower performance in the generation process. An optimal trajectory is the one that corrects the state of the process in the safest way, with the best performance and in the shortest time.

Figure 6 shows the results of the experiments, defining the optimal policy to reach the state 21, utilizing the learned matrices of Fig. 5.

The experiments were carried out in a restricted area of operation, with discretized values of the variables and considering simple actions. However, the results show the expected policy that experts would execute. Notice that all the commands consider the safety conditions expressed in the reward matrix. Additional experiments will consider a wider area of operation and will include more complex navigation processes that justify the use of MDP and POMDP in the control of power plants.

7 Conclusions and Future Work

This paper has shown the experiments carried out in the utilization of AI methods in the assistance of traditional methods for the control of power plants. Specifically, the control of water level of the drum in a heat recovery of a plant was studied.

This application was chosen because there is a natural uncertainty in both, the optimal control action required in response to changes in the load, and in the exact values provided by the sensors. This uncertainty is due to the thermodynamic conditions of the process involving water, steam and high pressure.

The transition matrices were obtained with real data sampled during 260 hours of operation of a plant in different conditions. The reward matrix was obtained with expert advice from the operator of the plant. The actions considered were operations of increment or decrement of the variables involved. A Software architecture was proposed that includes a learning module that refines the knowledge base according to the normal operation conditions of a specific equipment. Also, this architecture shows the modules necessary

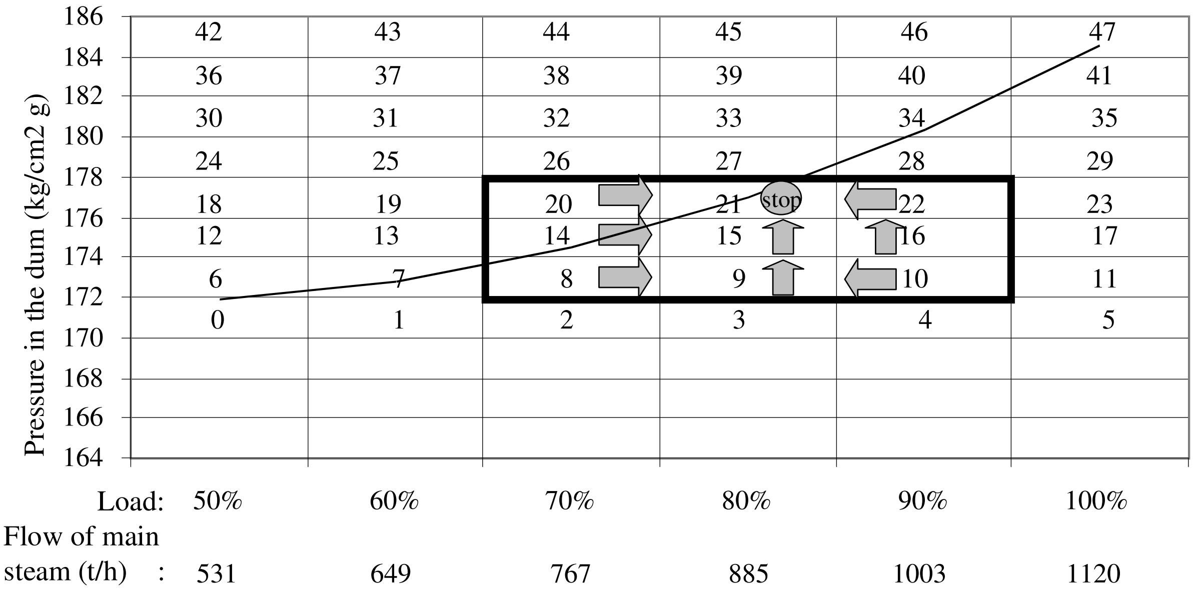

Figure 6: Experimental results: optimal policy to reach the state 21.

to integrate the environment of application, the actors that execute the commands recommended by the decision module, and the actors that establish the goals.

The proposed architecture has been utilized in the generation of a simple policy of actions, based on real data and the results are coherent according to the experts. Also, the experiments help this research group to define the roll that AI has in control applications.

The experiments shown in this paper correspond to the assumption of ideal sensors, i.e., there is no uncertainty about the state of the world, and so POMDP results are not yet presented. Future experiments will consider the characteristic of a partially observable world.

References

C.F.E. (n.d.), B&w boiler, reference manual, Technical report, Federal Commission of Electricity.

Givan, R., Leach, S. & Dean, T. (2000), 'Bounded-parameter markov decision processes', *Artificial Intelligence* **122**(1-2), 71–109.

Kaelbling, L. P., Littman, M. L. & Cassandra, A. R. (1998), 'Planning and acting in partially observable stochastic domains', *Artificial Intelligence* **101**(1-2), 99–134.

Puterman, M. L. (1995), *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, New York, N.Y., U.S.A.

White, D. (1993), *Markov Decision Processes*, Wiley, West Sussex, England.