

## Stochastic Learning Feedback Hybrid Automata For Dynamic Power Management in Embedded Systems

Teodora Erbes

Multi-Media Department  
Eurecom Institute  
Sophia-Antipolis, FRANCE  
Phone: +33 493002604,  
Fax: +33 493002627,  
E-mail: terbes@free.fr

Sandeep K. Shukla

Bradley Department of Computer Engineering  
Virginia Polytechnic Institute and State University  
Blacksburg, VA 24060 USA

Pushkin Kachroo

**Abstract** - Dynamic Power Management (DPM) refers to the strategies employed at system level to reduce energy expenditure (i.e. to prolong battery life) in embedded systems. The trade-off involved in DPM techniques is between the reductions of energy consumption and latency suffered by the tasks. Such trade-offs need to be decided at runtime, making DPM an on-line problem. We formulate DPM as a hybrid automaton control problem and integrate stochastic control. The control strategy is learnt dynamically using Stochastic Learning Hybrid Automata (SLHA) with feedback learning algorithms. Simulation-based experiments show the expediency of the feedback systems in stationary environments. Further experiments reveal that SLHA attains better trade-offs than several former predictive algorithms under certain trace data.

### I. INTRODUCTION

The increasing usage of mobile electronic equipment has led energy efficiency to become an increasingly significant consideration in system design. Portable systems have limited energy supply, thus reducing power dissipation directly results in an extension of battery life, and consequently represents an increase in the autonomy of the device. System devices, such as disk drives, microphones and modems, are built with multiple power states, which are accessible for management by the operating system through industry standard APIs [5] [6]. DPM refers to strategies for reducing system level power dissipation by switching system components to lower power modes when idle, and reviving them to the active state to service incoming requests. DPM has widely been researched for deriving techniques of device administration that yield the most reduction in energy consumption with the least amount of runtime computational effort. It is an online problem since an algorithm that administers power management must operate with no knowledge of the future. As a result, deterministic or stochastic predictions about the future are needed. The problem is hence translated into deciding whether to switch the system devices to lower power modes while the system is idle to reduce energy dissipation while maintaining functionality requirements.

Earlier research on prediction-based dynamic power management can be classified in two categories: adaptive [7] [8] [11] [12] [13] [14] and non-adaptive. Most adaptive DPM strategies base their prediction on a sequence of previous idle period lengths, and express their prediction of the next idle period with a single value. However, a problem arises when two different idle period lengths are predicted to be equally likely since the transition can be made only according to one predicted value and a penalty is endured in the case that the idle period was of the length of the other prediction. The probability-based strategies [9] [10] [15] [16] handle this uncertainty in the prediction by discovering a probability distribution for the idle periods from the input sequence: the algorithms base their decisions according to the characteristics of the prediction, which allows for a larger flexibility in the estimates. Two different research approaches arise from this probabilistic strategy: A first category of algorithms assumes a certain density function for the input and sets the thresholds accordingly [17], and a second technique attempts to learn the probability distribution online and adapts the model parameters dynamically [2].

From a DPM viewpoint, an embedded system is an association of discrete states, the different power modes, and continuous dynamics, the power consumption rates. Consequently, we model such systems with hybrid automata, which are as well composed of discrete states, the states of the automaton, and continuous dynamics, differential equations that govern the continuous variables in each state. Given that power management is severely handicapped when incorrect prediction of the idle periods is assumed, we add control to the model to guide the automaton through power modes while the system is idle. We use stochastic control in the mathematical model to attempt to predict probabilistically the range of lengths of the future idle periods. Several feedback learning algorithms are incorporated in the final SLHA model, attempting to teach the automaton the characteristics of

the idle periods and hence the correct behavior during idle time.

## II. MATHEMATICAL MODEL

We use a timed hybrid automaton to model a system with multiple power-down modes. The discrete states of the hybrid automaton are used to model the power modes of the system, while the continuous dynamics account for the power consumed in each mode. No specific control theory was formulated in this initial stage of the study: external control management was assumed for the purpose of developing and describing the internal behavior of the mathematical model. Analysis of the control synthesis is detailed in the subsequent section of this paper.

There are  $n+1$  main states in the model, each representing a power mode of the system. State  $S_0$ , the Active state, is the initial state of the system. The system needs to be in this main state to process requests. The states labeled  $S_i \forall i \in [1, n]$  represent the lower power modes of the system. The states are ordered from highest power consumption to lowest power consumption, such that the lowest power mode of the system is represented by the state with the highest index:  $S_n$ . Three constants are associated with each state:

- $P_i$ , the Power Consumption, is the power consumed while in state  $S_i$ ;
- $E_i$ , the Start-Up Energy, is the energy required to power-up from state  $S_i$  to state  $S_0$ ;
- $t_i$ , the Start-Up Time, is the time that it takes the system to activate from state  $S_i$ .

The following classifications are implied:

$$\forall i, j \text{ where } j > i, P_i > P_j, E_i < E_j, \text{ and } t_i < t_j,$$

such that the states with lower power consumptions have higher start-up energies and times.

In addition, the model includes intermediate states for transitioning from a lower power state to the active state.

## III. STOCHASTIC LEARNING FEEDBACK HYBRID AUTOMATA FOR DPM

For the purpose of the mathematical model, the control variable was assumed to be handled externally, and focus was given on the analysis of the internal behavior of the hybrid automaton. The next objective of this research was to devise a method for choosing a value for  $u$  at every instant of time. In this section, a solution is proposed for the management of the control variable: control theory is made probabilistic, by formulating probabilities of switching between states. Consequently, stochastic control is incorporated to the hybrid automaton, and learning feedback is added to the mathematical model.

The hybrid model described in section 2 was adapted such that the control variable  $u$  was customized to be represented by switching probabilities in the SLHA model. All the system parameters are as described in section 2. The external variable  $u$  was however replaced by variables  $p_{ij} \forall i, j \in [0, n], j \neq i$ , the action probabilities. Every allowed main-state transition  $S_i \rightarrow S_j \forall i, j \in [0, n], j \neq i$  is labeled by a probability  $p_{ij}$  that represents the probability of switching from state  $S_i$  to state  $S_j$ . These probabilities hold the following property:

$$\forall i \in [1, n], \sum_{j=1}^n p_{ij} = 1,$$

given that transition  $S_i \rightarrow S_j$  is allowed.

In order to enable the online learning of the input probability distributions, we developed our SLHA model as variable structure automata whose action probabilities are frequently recomputed using reinforcement techniques. We have incorporated several feedback stochastic learning algorithms, as described below, to study the behavior of our SLHA model for DPM.

### A. General Linear Reward-Penalty Scheme

For linear learning schemes, the following reinforcement functions are chosen:

$$g_j[p(n)] = a \cdot p_j(n), \quad h_j[p(n)] = \frac{b}{r-1} - b \cdot p_j(n),$$

where  $0 < a < 1, 0 \leq b < 1$

### Symmetric Linear Reward-Penalty Scheme ( $L_{RP}$ )

The symmetric linear reward-penalty scheme is equivalent to the general linear reward-penalty scheme, with the particular condition that the reward and penalty parameters are equal:  $a=b$ . This clause engenders symmetric reward and penalty updates such that the learning for the probability  $p_i$  of action  $\alpha_i$  in the case when the application of action  $\alpha_i$  results in a success is identical to the learning engendered when the application of action  $\alpha_i$  results in a failure.

### Linear Reward-Inaction Scheme ( $L_{RI}$ )

The linear reward-inaction scheme is a special case of the general linear reward-penalty scheme, with the stipulation that there is no learning penalty in the case of failure:  $b=0$ .

Following are the reinforcement functions for the non-linear learning scheme employed:

### B. Nonlinear Scheme 1

$$g_j[p(n)] = \frac{a}{r-1} \cdot p_i(n) \cdot (1 - p_i(n)),$$

$$h_j[p(n)] = \frac{b}{r-1} \cdot p_i(n) \cdot (1 - p_i(n)),$$

where  $0 < a \leq 1$ ,  $0 < b \leq 1$ .

### C. Nonlinear Scheme 2

$$g_j[p(n)] = p_j(n) - \phi[p_j(n)],$$

$$h_j[p(n)] = \frac{p_i(n) - \phi[p_i(n)]}{r-1},$$

where  $0 \leq \phi[p_j(n)] \leq p_j(n)$ .

$\phi$  is usually chosen as:  $\phi(x) = ax^m$ , where  $0 < a \leq 1$ ,  $m \in [2, \infty)$ .

### D. Hybrid Scheme H

$$g_j[p(n)] = a \cdot p_j(n),$$

$$h_j[p(n)] = \begin{cases} a \cdot p_j(n) & \text{if } p_i(n) \in \left[ \frac{a}{1+a}, \frac{1}{1+a} \right], \\ 0 & \text{otherwise} \end{cases}$$

where  $0 < a < 1$ .

If  $\frac{a}{1+a} \leq p_i(n) \leq \frac{1}{1+a}$ , this scheme is equivalent to the linear reward-reward penalty algorithm. Otherwise, this scheme follows the principles of the linear reward-inaction updates.

## IV. EXPERIMENTAL RESULTS WITH SLHA AND CONCLUSIONS

In order to examine the suitability of the mathematical model for DPM, we developed software in C++ to simulate the SLHA system when presented with input traces. The simulator follows the guidelines for DPM, as detailed earlier in this work and its behavior is controlled stochastically by switching probabilities, where learning is performed by selected reinforcement schemes. Once the system configurations are entered, the software runs the input file for DPM and outputs the total time and consumption details of the specified system for the tested input file. A variety of parameters can be entered to allow for a substantial flexibility in the systems to simulate. The specifics of the parameters and operation of the simulator can be found in detail in [1].

To study the behavior of the SLHA systems given various input distributions, simulations were performed with several combinations of the mathematical model parameters, and the results are analyzed based on the consumption of each system studied. A SLHA model of a four-state mobile hard-drive from IBM [3] was employed for simulating DPM. The time and energy specifications of the embedded system are shown in Table I, following the format detailed in section II.

Table I. Power-Mode Characteristics for IBM HardDrive

State	Power Consumption (Watts)	Start-Up Energy (Joules)	Start-Up Time (ms)
Active $S_0$	1.9	0	0
Idle $S_1$	0.9	0.56	40
Stand-By $S_2$	0.2	1.575	1500
Sleep $S_3$	0	4.75	5000

### A. Expediency Analysis: Two-State Automata

Firstly, simulations were performed to determine the optimal parameters of the SLHA model to reach correct convergence in stationary environments. These were undertaken with constant and bipolar input distributions on two-state automata containing an active state and a sleep state, where the latter corresponds to the lowest power mode of the modeled IBM system. Two separate simulations were run for the *Preemptive* method and the *On Demand* method respectively and the experiments were realized in three different consumption-optimization categories: optimize energy and latency, optimize only energy, and optimize only latency. Energy, latency and consumption competitive ratios were used as metrics for the assessment of the systems: The energy, latency and consumption results for each configuration were summed over all the input files, and the outcomes of each quantity were divided by the corresponding results of the optimal algorithm.

Observation was made that the systems tend to reach convergence more easily when configured with low reward updating parameters. Additionally, the systems that use the second non-linear learning scheme converged with a high reward parameter, for all degrees of non-linearity, and with a low reward parameter for high degrees of non-linearity. Finally, the first non-linear learning scheme appeared to be robust for the input distributions examined.

Furthermore, the two wake-up methods presented equivalent energy results. In addition, the configurations yielded equivalent results for systems that optimize energy and latency, and for systems that optimize only energy. However, when optimizing latency, no configuration brought the systems to correct convergence.

### B. Real Trace Analysis: Four-State SLHA

To simulate the SLHA model with real input distributions, we used input files that were adapted from trace data obtained from the auspex file server archive [4]. It is noted that the different input files have significantly different idle-period distributions, which yielded noticeably different behaviors of the systems. In addition, we used the configurations that yielded convergence in the preliminary simulation sets for this part of the experimentation. Several sets of simulations were run for different values of the time increment, and simulations were performed in two categories: optimization of energy and latency, and optimization of only energy.

It was observed that *On Demand* wake-up tends to better minimize energy and latency expenditure than the *Preemptive* wake-up method. This is explained by the fact that with such a wake-up method the optimal offline algorithm also provokes latency, such that the quotient of the two costs produces a smaller value than the corresponding ratio of the *Preemptive* method.

Moreover, configurations corresponding to the first non-linear reinforcement scheme with high reward and penalty parameters perform the best minimization of energy for the presented traces. Furthermore, configurations corresponding to the second non-linear updating scheme with a high reward parameter and a high degree of nonlinearity perform the best minimization of latency for the presented traces. The observed contrast between energy and latency is explained by the definition of consumption, which is a weighted sum of the two factors. Hence, a low consumption can equally be reached with high energy and low latency expenditures, or conversely with low energy and high latency costs.

It was further observed that the systems produce slightly different results according to the refreshing frequency  $T$ . This is related to the idle-period lengths of the input traces. If  $T$  is often longer than the idle-periods, the system has a lower opportunity to switch states during the idle times, which makes it harder to reach optimality. On the other hand, if the time increment  $T$  is too low, the system will more favorably reach optimality but will however require more, and possibly unnecessary, processing time and power.

Finally, observation was made that optimizing only energy yields similar energy expenditures, but significantly lower latency costs than the method for the optimization of both energy and latency.

### C. Competitiveness compared to Former DPM strategies

Finally, we compared the results of former DPM strategies [2] to the performance of the SLHA model, given real-trace inputs. The chosen SLHA systems

presented the lowest results with respect to either energy expenditure, latency incurred or consumption cost for each wake-up method in the real-trace experiment. The details of the six SLHA configurations are given in Table II. The algorithms were run to optimize energy for the DPM problem.

Table II. Configuration Details of the Best SLHA Systems

	Updating Algorithm	$a$	$b$	$m$	$T$ (ms)
SLHA <sub>1</sub>	Non-Linear 1	0.9	0.1	-	1
SLHA <sub>2</sub>	Non-Linear 1	0.9	0.9	-	10
SLHA <sub>3</sub>	Non-Linear 1	0.9	0.9	-	100
SLHA <sub>4</sub>	Non-Linear 2	1	-	5	1
SLHA <sub>5</sub>	Non-Linear 2	1	-	5	10
SLHA <sub>6</sub>	Non-Linear 2	1	-	5	100

Fig. 1 and Fig. 3 illustrate the total energy ratios versus total latency ratios for all the studied systems for *Preemptive* wake-up and wake-up *On Demand* respectively. Fig. 2 is a detailed view of Fig. 1 around the result of the optimal offline algorithm.

When comparing the competitive ratios of the DPM strategies it can be observed that the competitive costs of the SLHA systems are significantly lower than those of the former DPM strategies with the *Preemptive* wake-up method. Indeed, Fig. 2. shows that the systems with configuration  $\{SHLA_1, SHLA_2, SLHA_3\}$ , using the first non-linear reinforcement scheme with high reward and penalty parameters, yielded the lowest results among all the studied former strategies. Additionally, configurations  $\{SLHA_4, SLHA_5, SLHA_6\}$ , using the second non-linear updating algorithm with a high reward parameter and a high degree of non-linearity, yielded the lowest total latency cost while keeping the energy ratio around unity. Moreover, these configurations outperform the former algorithms *LAST* and *TREE* in total energy expenditure.

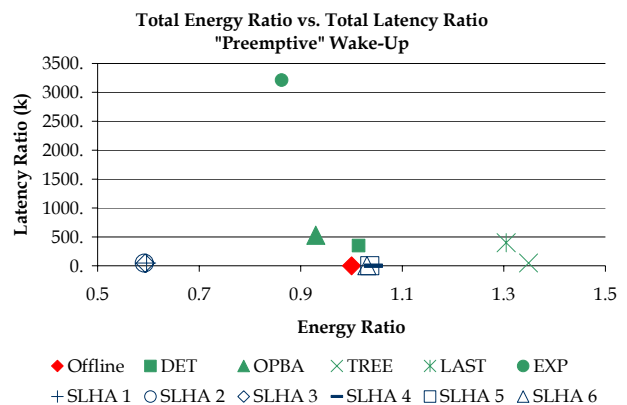


Fig. 1: Total Energy Ratio vs. Total Latency Ratio for Preemptive wake-up

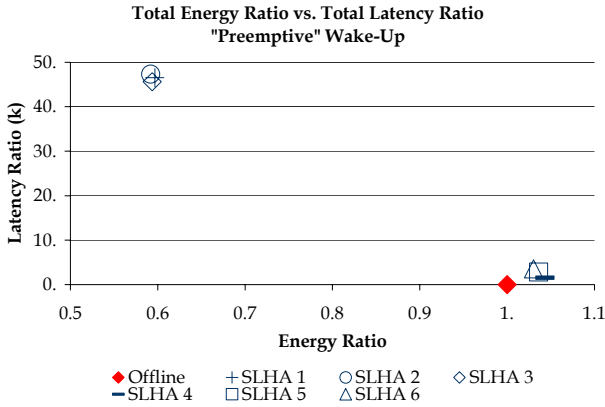


Fig. 2: Close-Up of the Total Energy Ratio vs. Total Latency Ratio for Preemptive wake-up

Furthermore, the energy competitive ratios of the selected SLHA systems are all less than or equal to unity, reaching as low as 0.59. This indicates a superior performance in energy efficiency than the offline algorithm. This phenomenon is once more due to the fact that consumption is a weighted sum of energy and latency. A reduced consumption can hence be attained by a reduction either in energy or in latency, potentially retaining the other cost high.

When examining the results of the *On Demand* method, Fig. 4. illustrates that the SLHA systems with configurations  $\{SHLA_1, SHLA_2, SHLA_3\}$  also yielded lower energy expenditures than all of the former algorithms except *EXP*, reaching as low as 0.6. They however presented significantly higher latency costs. Furthermore, the SLHA system with configuration  $\{SLHA_4\}$  presented the lowest energy cost among all the studied models, except *DET*. Its energy expenditure is however much higher than any of the other models examined.

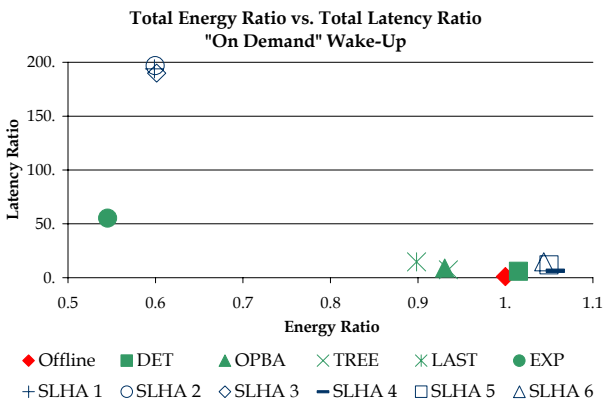


Fig. 3: Total Energy Ratio vs. Total Latency Ratio for On Demand wake-up

#### D. Conclusions

From the experimental results, we observed that the SLHA mathematical model for the Dynamic Power Management of Embedded Systems is competitive and most of the times yields better average results than the former strategies presented in literature. In particular, SLHA proved its superiority with *Preemptive* wake-up, for the examined input patterns. Furthermore, results were observed to be enhanced for either the conservation of energy or the prevention of latency, for wake-up *On Demand*.

Added to its superior performance, SLHA offers a high versatility that places this model as extremely competitive for the DPM problem.

Given a DPM problem for embedded systems, choice should therefore be made on selecting the presented SLHA model, configured either with the first non-linear learning algorithm with a low reward parameter, or with the second non-linear learning scheme with a high reward parameter and a high degree of nonlinearity. Selection should also be made on performing with the *Preemptive* wake-up strategy and optimization should only be performed for the minimization of energy expenditure.

#### REFERENCES

- [1] T. Erbes, "Stochastic Learning Feedback Hybrid Automata for Dynamic Power Management in Embedded Systems". <http://fermat.ece.vt.edu/teodora>, 2003.
- [2] S. Irani, S. Shukla, and R. Gupta, "Online Strategies for Dynamic Power Management in Systems with Multiple Power Saving States", ACM Transactions on Embedded Computing Systems, Special Issue on Power-Aware Embedded Computing, Aug. 2003.
- [3] Technical specifications of hard drive IBM Travelstar VP 2.5inch, <http://www.hgst.com/hdd/support/table.htm#Travelsta>, 1996. – 2004.
- [4] Auspex File Traces from the NOW project, available at, <http://now.cs.berkeley.edu/Xfs/AuspexTraces/auspex.html>, 1993.
- [5] Intel, Microsoft, and Toshiba, "Advanced Configuration and Power Interface Specification". <http://www.acpi.info>, Dec. 1996.
- [6] C. Pereira, R. Gupta, P. Spanos, and M. Srivastava, "A Power Aware API". Power Aware Computing, 2002.
- [7] Chi-Hong Hwang, C. Allen, and H. Wu, "A Predictive System Shutdown Method For Energy Saving of Event-Driven Computation", in Proceedings of the IEEE/ACM International Conference on Computer Aided Design, pp. 28-32, Nov. 1996.
- [8] M. B. Srivastava, A. P. Chandrakasan, and R. W. Broderson, "Predictive Shutdown and Other Architectural Techniques for Energy Efficient Programmable Computation", IEEE Transactions on VLSI Systems, 4(1):42–54, Mar. 1996.
- [9] Q. Qiu, and M. Pedram, "Dynamic Power Management Based on Continuous Time Markov Decision Processes", In Proceedings of the Design Automation Conference, 1999.
- [10] Q. Qiu, Q. Wu, and M. Pedram, "Stochastic Modeling of Power Managed Systems: Construction and Optimization", in Proceedings of the International Symposium of Low Power Electronics and Design, pp. 194-199, Aug. 1999.
- [11] R. Karlin, M. Manasse, L. McGeoch, and S. Owicki, "Randomized competitive algorithms for non-uniform problems", in 1st Annual ACM/SIAM Symposium on Discrete Algorithms, pp. 301-309, Jan. 1990.
- [12] D. Ramanathan, S. Irani, and R. Gupta, "An Analysis of System Level Power Management Algorithms and their effects on Latency", Accepted to be published at the IEEE Transactions on Computer Aided Design.

- [13] L. Benini, A. Bogliolo, G. Paleologo, and G. D. Micheli. "Policy Optimization for Dynamic Power Management", IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 18(6): 813-833, 1999.
- [14] E. Y. Chung, L. Benini, and G. D. Micheli, "Dynamic Power Management using Adaptive Learning Tree", In proceedings of International Conference on Computer Aided Design, 1999.
- [15] Y. Lu, E. Chung, T. Simunic, L. Benini, and G. De Micheli, "Quantitative Comparison of Power Management Algorithms", Proceedings of Design Automation and Test, Europe, 2000.
- [16] T. Simunic, "Energy Efficient System Design and Utilization", Ph.D Thesis, Stanford University, 2001.
- [17] R. Gupta, S. Irani and S. Shukla, "Formal Methods for Dynamic Power Management", In proceedings of International Conference on Computer Aided Design, 2003.