Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

Model-Based Design of a Dynamic Voltage Scaling Controller Based on Online Gradient Estimation Using SimEvents

Wei Li, Christos G. Cassandras, and Michael Clune

Abstract – **This paper demonstrates SimEvents^{TM}, a model-based design and simulation tool, and its application in assisting power awareness micro-controller design. A Dynamic Voltage Scaling (DVS) controller based on online gradient estimation is proposed. Simulation models are built in SimEvents, which proves the effectiveness of the control scheme.**

I. Introduction

Recent developments in embedded systems and radio-frequency communication enable the application of distributed control systems over a wireless network. However, important issues arise during the design and operation of such systems, such as multi-process scheduling of the Electronic Control Unit (ECU), real-time control over a network and power management.

Software tools are needed to assist the effective analysis and solution of such problems. The concept of Model-Based Design and related software tools fits these requirements and becomes commonly accepted by industrial and scientific communities.

This paper presents an application of on-line gradient estimation techniques (i.e. Infinitesimal Perturbation Analysis or IPA) to the power control of ECUs [1]. A Dynamic Voltage Scaling (DVS) controller is designed and evaluated using SimEvents^{TM}, a discrete event simulator in the MATLAB®/ Simulink® family. By monitoring the current workload of the system and performing on-line gradient estimations, this DVS controller dynamically updates the input voltage of the ECU of the device, so that the power consumed by the ECU is optimized and at the same time the quality of service measured by the average system time of a job is guaranteed.

SimEvents contains libraries and block sets (e.g. Queues, Servers, Gates and Switches) that model the basic components of a Discrete Event System (DES). By inter-connecting these building blocks, one can easily model a DES such as a communication network or a manufacturing system. SimEvents can also be jointly used with Simulink and facilitate hybrid system design and simulation.

This section first introduces the IPA technique and its importance in the design and performance optimization of discrete event systems. Then, the motivation for DVS is briefly reviewed as the background for the proceeding sections that follows.

A. Sensitivity Analysis and the IPA Technique

During system design, maximizing performance while minimizing cost are always the most critical objectives. Performance is normally measured by the quality of service a system can supply. For a DES, commonly applied performance metrics include average system time, utilization and packet loss rate. Cost, on the other hand, is usually measured by the expenditure to construct and operate the system or the resources that the system utilizes, such as communication channels, space, time and energy. When designing a system, we face the question of how to assign limited resources and maximize the performance of the system.

For a complex engineering system, simulation is a powerful tool to assist the design process. Through model-based design and simulation, one can efficiently estimate the performance and corresponding cost of a system, and obtain accurate insights from the results. However, a well-recognized difficulty in model-based design is the parameter tuning of the systems. Since the operation of a system may depend on the configuration of many parameters, and traditional simulation techniques can only replicate a single trace of the system at a time, it is always a painful task to repeatly execute the simulation and find a set of parameters that optimally fits the design requirements.

When designing a DES, the optimal parameter configuration issue becomes more severe. Since a DES is normally subjected to stochastic inputs and random noise, considerably greater effort (in terms of simulation time and computation complexity) is needed to obtain a precise evaluation of the system. In addition, sensitivity information of performance metrics cannot be easily obtained through approaches that are commonly used in continuous systems. The seemingly universally applicable "finite difference" approach is both time-consuming and error-prone.

Infinitesimal Perturbation Analysis (IPA) techniques have proved to be an effective tool to resolve the optimal parameter configuration problem. By using IPA, one can obtain sensitivity information of the parameterized system from a

Manuscript received March 19, 2006.
Wei Li is with The MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760, USA (phone: 508-647-7704; e-mail: Wei.Li@mathworks.com).
Christos G. Cassandras is with the Department of Manufacturing Engineering and Center for Information and Systems Engineering, Boston University, Boston, MA 02215, USA (e-mail: cgc@bu.edu). His work is supported in part by NSF under Grant DMI-0330171, and by AFOSR under grants FA9550-04-1-0133 and FA9550-04-1-0208
Michael Clune is with The MathWorks Inc., 3 Apple Hill Drive, Natick, MA 01760, USA (e-mail: Michael.Clune@mathworks.com).

1-4244-0171-2/06/$20.00 ©2006 IEEE.

single simulation. With such sensitivity information, we obtain from a simulation not only the current performance of the system, but also a first order approximation of the performance metrics as a function of the configurable parameters.

In this paper, we will present an application of the IPA technique to the DVS of ECUs. To offer a background, the rest of this section goes over the basic concept of DVS.

B. Dynamic Voltage Scaling

Today's technology enables the integration of computation, communication and control into a compact and economical device. For example, emerging sensor networks can monitor the environment and report collected data wirelessly.

At medium load, a wirelessly connected sensor node is supposed to work consecutively for 1~2 years without recharging. This implies that the power consumption of the electronic device must be carefully managed. Statistics show that the ECU accounts for about 50% of the total power consumption of these devices [2]. For this reason, the power management of the ECU becomes a common interest of chip manufactures, device designers and software providers.

Traditional approaches use on-off control to manage the duty-cycle of the ECU. According to this approach, when the workload of the ECU is low or when it is idle, the processor's circuit will be completely shut down. When a new job arrives, the processor is awakened and the job is processed at full speed.

In addition to on-off control, the DVS techniques further manage energy usage by changing the working frequency (i.e. speed of processing) of the processor. In this way, 50% more energy is saved than the case where only on-off control is applied.

The basic motivation of DVS power control is easy to understand. When the workload of the ECU is low, the DVS controller lowers the input voltage of the processor so that it works at a relatively slow pace, thus less energy is needed. Whereas when the workload is high, DVS controls the processor working at a higher frequency, so that the QoS can be ensured. However, due to the stochastic nature of job arrival processes and the randomness of the workload of incoming jobs, the DVS controller needs to be carefully designed so that it can lower power usage without impairing the overall performance of the system [3].

The remainder of the paper presents the design of a DVS controller based on IPA gradient estimation. The queueing model and DVS controller are discussed in Section 2. Simulation results and comparisons to analytical solutions are shown in Section 3.

II. Queueing Model and DVS Controller

As shown in Fig. 1, the ECU of a portable device is modeled as a single server queueing system. Incoming jobs are stored in a buffer that can be modeled as a non-preemptive FIFO queue with infinite capacity. The arrival process is stochastic. The inter-arrival times are random variables that are independent and identically distributed (i.i.d.). Except for the i.i.d. property, we assume that we know nothing about the arrival process, neither the distribution of the inter-arrival time, nor the rate of arrival.

Fig. 1. Queueing Model of an ECU

In addition to the randomness in the arrival process, the size (or workload) of each individual job is random. We assume that workloads of jobs are i.i.d. random variables with a known distribution. For example, the workload of a job may be an exponentially distributed random variable, with average load of 10^6 operations.

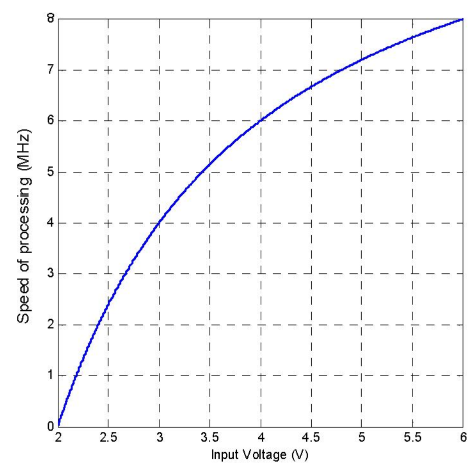

Fig. 2. Processing speed of an ECU at different voltage levels

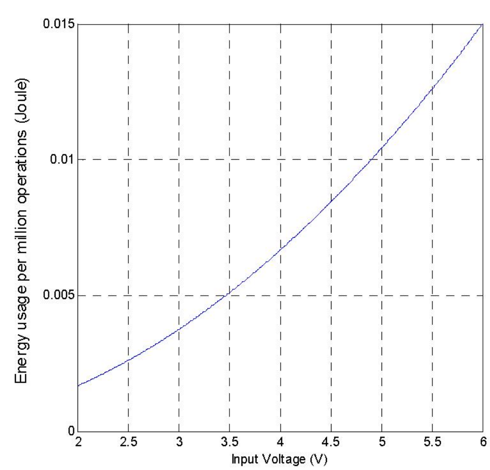

Fig. 3. Energy usage of an ECU at different voltage level

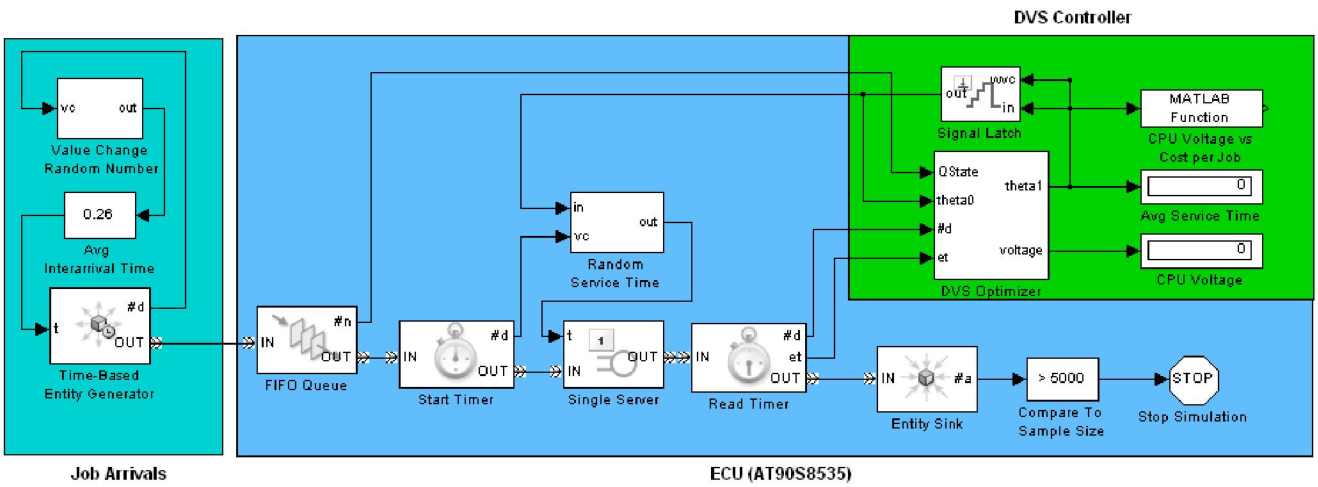

Fig. 4. Model-based design of DVS controller using SimEvents and MATLAB/Simulink

A. ECU Model

The ECU can be modeled as a single server with adjustable processing rate. As presented in Section I, DVS manages power usage through dynamically changing the processing rate. For current devices, this can be realized by controlling the input voltage of the processing unit.

Let us use a specific product as an example. According to the data sheet of AT90S8535, the speed of processing can be selected from 0~8MHz with corresponding input voltage 2.7V~6.0V. It is shown that the processing speed is a function of the input voltage. For the specific processor, this function is given by,

$$V = \frac{V_t}{1-c_1 f}, \text{ or } f = \frac{1}{c_1}\left(1-\frac{V_t}{V}\right) \qquad (1)$$

Here, f is the processing speed (in MHz), V is the input voltage (in Volts). V_t is the device's threshold voltage (minimal input voltage), and c_1 is a device dependent constant. For AT90S8535, V_t =2V c_1 =0.0833. Fig. 2 illustrates the functional relationship between V and f.

On the other hand, the energy the processor consumes to process a job can be expressed by the following formula,

$$P = c_2 N V^2 \qquad (2)$$

where $c_2 = 0.4167 \times 10^{-3}$ is a device dependent constant, N is the number of operations needed to process the job (in 10^6 operations), V is the input voltage and P is the energy usage (in Joules). For a job with 1M operations, Fig. 3 presents its energy usage at different voltages.

The processing speed model together with the energy model (i.e., eq. (1) and (2)) enables DVS power control. For some job with 1M operations, when input voltage increases from 4V to 6V, the energy usage changes from 6.67mJ to 15mJ at corresponding processing speeds of 6MHz and 8MHz.

B. SimEvents Model of the DVS Controller

Fig. 4 is a SimEvents/Simulink model of a DVS controller for the single server queueing system presented in figure 1. The SimEvents is a "queues and servers" style discrete event simulation extension of the MATLAB/Simulink product family. It allows entities to be passed from block to block to represent the movement of jobs through the microprocessor. This model generates entities using the Time-Based Entity Generator using an intergeneration time from its 't' port. The generated entities are stored in the FIFO Queue block before being delayed by the Single Server block and subsequently being sent to the Entity Sink. The time for the entity to pass through the FIFO Queue block and Single Server block is marked by the Start Timer and End Timer blocks. This time is used in conjunction with the counts of entities passing through other blocks by the DVS optimizer to calculate performance metric below (see eqs. (3) through (9)). The perturbation is supplied by the Random Service Time subsystem block that varies the service time used by the Single Server block.

C. DVS Controller based on IPA

When operating, the DVS controller will monitor the current workload of the processor and dynamically adjust the input voltage (thus the processing speed) of the processor.

In order to offer energy savings and ensure QoS at the same time, an optimizer is constructed to find the optimal input voltage. During optimization, the performance metric is given by,

$$J(V) = wP(V) + S(V) \qquad (3)$$

$J(V)$ comes from two parts: (*i*) $P(V)$ –the average energy consumption of a job, and (*ii*) $S(V)$ –the average system time of a job. In (3), w is a weighting parameter. It can be thought of as the price of energy relative to processing delay.

By using (3), we can optimize energy usage without impairing the QoS offered by the processor (i.e. if there is no $S(V)$ term in (3), the optimized input voltage will be the smallest feasible voltage level, which may incur an unacceptable delay).

We assume that the average workload of a job is 1M operations. By combining (1), (2) and (3), the performance metric can be rewritten as,

$$J(\theta) = wc_2\left(\frac{V_t}{1-c_1/\theta}\right)^2 + S(\theta) \qquad (4)$$

Note in (4), we replace V by θ--the average service time of a job ($\theta = 1/f$). We perform this substitution for mathematical simplicity. Because of the one-to-one mapping between θ and V, this substitution does not change the nature of our optimization problem.

According to the necessary condition for optimality, when minimizing (4), the optimal solution must satisfy

$$\frac{dJ}{d\theta} = -\frac{2wc_1c_2V_t^2\theta}{(\theta-c_1)^3} + \frac{dS}{d\theta} = 0 \qquad (5)$$

During simulation, if we can estimate $dS/d\theta$ and derive $dJ/d\theta$, a gradient method then can be applied to find a θ that satisfies (5). In the DVS controller, we adopt a gradient method with constant step size. That is, in the kth iteration of the optimization, start from θ_k, we derive a new average service time by,

$$\theta_{k+1} = \theta_k - \Delta\left[\frac{dJ}{d\theta_k}\right] \qquad (6)$$

Where Δ is a constant step size ($\Delta = 0.001$ in the proposed DVS controller). (6) is the well-known "steepest decent method" and by applying it, it is guaranteed that $J(\theta_{k+1}) \le J(\theta_k)$. In (6), the $dJ/d\theta_k$ is estimated using,

$$\left[\frac{dJ}{d\theta_k}\right] = -\frac{2wc_1c_2V_t^2\theta_k}{(\theta_k-c_1)^3} + \left[\frac{dS}{d\theta_k}\right]_{IPA} \qquad (7)$$

with $[dS/d\theta_k]_{IPA}$ being the gradient estimate generated by an IPA estimator [1]. By combining (6) and (7), we can summarize the iterative process to find an optimal θ (and thus V) to minimize the average cost of a job (i.e. $J(\cdot)$ in (3) and (4)).

TABLE I
DVS CONTROL ALGORITHM DESCRIPTION

At $t = t_0 = 0$	Start simulation with an arbitrary initial input voltage V_0 and corresponding average service time θ_0
Upon the departure of the kth job ($t = t_k$)	• Derive $[dS/d\theta_k]_{IPA}$ based on new departure information • Obtain $[dJ/d\theta_k]$ by using (7) • Update input voltage V_{k+1} and average service time θ_{k+1} by using (6)
…Process repeats until the end of the simulation…	

III. Simulation Results

This section derives a theoretical optimal solution of (4) by assuming both the inter-arrival time and the workload of each job are exponential distributed random variables (thus the resulting queueing model is M/M/1). Then simulation results are compared with the corresponding theoretical values. It can be shown that the proposed DVS controller converges to theoretical values.

A. Theoretical Solution

When the queueing system is M/M/1, (4) can be rewritten as,

$$J(\theta) = wc_2\left(\frac{V_t}{1-c_1/\theta}\right)^2 + \frac{\theta}{1-\lambda\theta} \qquad (8)$$

Where λ is the arrival rate of jobs. (5), on the other hand, can be expressed by,

$$\frac{dJ}{d\theta} = -\frac{2wc_1c_2V_t^2\theta}{(\theta-c_1)^3} + \frac{1}{(1-\lambda\theta)^2} = 0 \qquad (9)$$

It can be seen that (9) is reduced to a high order polynomial equation. A theoretical solution can be derived for arbitrary arrival rate. Here, we give some of the theoretical solutions.

TABLE II
THEORETICAL RESULTS

λ (Jobs/s)	V^* (Volt)	θ^* (s)	f^* (MHz)
6.85	6	0.125	8
3.88	4	0.167	6
0.73	3	0.25	4

The parameters we adopted in (9) are: w=100, $c_1 = 0.0833$, $c_2 = 0.4167\times10^{-3}$, $V_t = 2$ and we assume that the average size of a job is 1M (1×10^6) operations.

B. Simulation Results and Comparison

The designed DVS controller and IPA estimator are constructed using SimEvents. The following table compares the optimal solution we obtained during simulation with the theoretical value. The simulated and theoretical values are very close demonstrating that SimEvents and Simulink can be used to find the optimal voltage for a wide range of job arrival rates.

TABLE III
COMPARE SIMULATION RESULTS WITH THEORETICAL VALUES

λ (Jobs/s)	V^{Simu}	V^* (Volt)	θ^{Simu}	θ^* (s)
6.85	6.014	6	0.1249	0.125
3.88	3.996	4	0.1668	0.167
0.73	3	3	0.25	0.25

Fig. 5 illustrates the convergence of ECU input voltage to the theoretical value during a simulation.

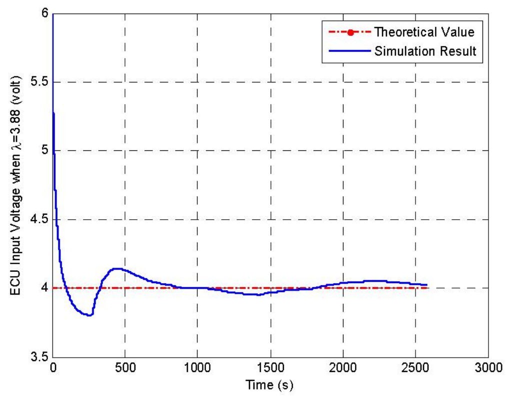

Fig. 5. Compare simulation results with theoretical value

IV. Conclusion

This paper demonstrates that SimEvents can be used to design microprocessor controllers that optimize power usage using the variations inherent in the inter-arrival time of jobs. The result shows the convergence of the voltage value to the theoretical optimum, as expected by the fact that IPA provides unbiased estimates. Note that it is only in the simple M/M/1 case that a theoretical value for $J(\theta)$ can be evaluated. SimEvents and IPA, on the other hand, are not dependent on such modeling assumptions and may be used for general inter-arrival and processing time distributions and provide information such as that shown in Fig. 5. These results are also an indication that IPA techniques can be applied in SimEvents to more complex models to efficiently find optimal values, determine sensitivity to changes in inputs values, and observe convergence properties.

References

[1] C. G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Boston, Kluwer Academic Publishers, 1999.
[2] M. Weiser, B. Welch, A. Demers, and S. Shenker, “Scheduling for reduced CPU energy,” in Proceedings of the 1st Symposium on Operating Systems Design and Implementation, pp. 13-23, 1994.
[3] S. Zhuang and C. G. Cassandras, “Optimal Dynamic Voltage Scaling for Wireless Sensor Networks with Real-Time Constraints,” in Proc. of SPIE Conf. on Intelligent Systems in Design and Manufacturing VI, Oct. 2005.