Proceedings of the 2006 American Control Conference
Minneapolis, Minnesota, USA, June 14-16, 2006

Low-Power Buffer Management Using Hybrid Control

Jason Ridenour, Jianghai Hu, and Yung-Hsiang Lu

***Abstract*— The power management problem is studied for a pipeline of streaming data consisting of multiple stages with buffer memories inserted between adjacent stages. The objective is to find the optimal strategy for dynamically changing the power states of the stages so that the power consumption of the overall system is minimized. By modeling the pipeline as a hybrid system, we derive various necessary conditions for the optimal solutions. In particular, an operation called folding is introduced that can improve the performance of an existing strategy. Analytic conditions are derived for determining whether folding can save power and the optimal number of folds.**

I. Introduction

Consider a pipeline of streaming data shown in Fig. 1, which consists of three components, X, Y, and Z, and two buffers in between. Component X generates data at the rate of α and stores them in the first buffer. Component Y retrieves data from the first buffer, processes them, and stores them in the second buffer, all at the rate of β. Component Z retrieves data from the second buffer at the rate of γ. This system can be the abstract model of many electronics and computer network applications. For example, in a PDA (Personal Digital Assist), X can be the network card downloading streaming video from the Internet; Y can be the processor to decode the data; and Z can be the display unit.

We assume that α, β, and γ are constants satisfying $\alpha > \gamma$ and $\beta > \gamma$. Thus there is an over-supply of data from upstream for Z to consume if all components are active. As a result, it is possible to reduce the system power consumption by turning off components X and Y at proper times, as they consume less power in the off state than in the normal operating state. This process is called *dynamic power management* [4]. In this process, the smooth operation of Z is ensured by the presence of the two buffers. For example, Z can retrieve and consume data already stored in the second buffer even if Y is turned off.

J. Ridenour and J. Hu are supported by Purdue Research Foundation. Y.-H. Lu is supported by the National Science Foundation under Grant No. 0347466. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

J. Ridenour, J. Hu, and Y.-H. Lu are all with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA. {ridenoja, jianghai, yunglu}@purdue.edu

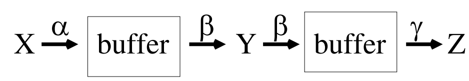

Fig. 1. Two buffers between three interacting components.

In this paper, we study the optimal power management problem for the pipeline of streaming data, which tries to find the strategy for coordinating the turning on and off of components X and Y so that maximal power reduction is achieved for the overall system. A solution will be presented by modeling the pipeline as a hybrid system.

Proposed to model dynamical systems with both continuous and discrete dynamics, hybrid systems have applications in many different engineering fields, such as Air Traffic Management (ATM) systems [13], automated highway systems [14], computer and communication networks [9], and auto and industrial systems [2]. We will use a version of the hybrid systems model called *multirate hybrid systems*. Our problem thus becomes an instance of the optimal control problems for hybrid systems [1], [3], [7], [8], [12], [15]. The difference of our problem with other optimal control problems for hybrid systems studied in the literature consists of the following two aspects: (i) our problem studies the optimal periodic solutions; (ii) in addition to the system trajectories, the shape and size of the state space of the hybrid system also need to be design at the same time. These make our problem a challenging one.

In our recent work [10], we present some preliminary results on applying the hybrid system model in the study of the dynamic power management problem for the pipeline of streaming data. Some necessary conditions for the optimal solutions are derived, and optimal periodic solutions are derived under the assumption that X and Y each turns on and off at most once in a period.

In this paper, we will extend our results in [10] by introducing an operation on an existing solution called *folding* that can generate another feasible solution with multiple switchings in a period but potentially lower average power than the original one. Conditions will be derived on whether folding will result in a better solution, and, if this is the case, the number of optimal folding operations one should carry out.

The rest of the paper is organized as follows. In Section II, the problem under study is formulated precisely. In Section III, we review some previous results on the optimal solutions in a constrained case. In Section IV, we introduce the folding operation and derive the main results on the optimal number of folds and scaling of a solution. Finally, in Section V, we outline the conclusions and some possible future directions.

II. Problem Formulation

We first analyze the power expenditure of the pipeline of streaming data shown in Fig. 1. The power consumed by each component consists of two parts: the dynamic power and the static power. The dynamic power includes the power to produce, process, and/or consume data, as well as the power for writing data into and retrieving data from the buffer memory. As the name suggests, the dynamic power depends on the data flow rate through the component. In contrast, the static power of the component is the part of the component's power used to sustain its normal operation, and is independent of the data rate. In this paper, we consider only the static powers, as it is shown in [5], [6], [11] that the dynamic powers do not affect the solution of the optimal buffer management problem defined later.

More precisely, each of the components X and Y has two power states: on and off. When X and Y are on, they consume the (static) powers p_x and p_y, respectively. When they are off, they consume no power. Changing the power states of these two components may incur extra energy. For simplicity, we assume that the energy overheads for turning on X and Y are k_x and k_y, respectively, while their turn-off energies are both zero. We also assume that any delay in changing the power states of X and Y is amortized by an early wakeup policy since the system is deterministic. Because the last component Z is assumed to be always turned on, it consumes a constant power. Thus its power contribution can be ignored in finding the optimal solutions.

In addition to component powers, the two buffers also consume powers that are proportional to their sizes. Denote by Q_1 and Q_2 the sizes of the two buffers. Then their powers are m_1Q_1 and m_2Q_2, respectively, where the constants m_1 and m_2 are the powers per unit size of the buffer memory (W/MB). If the two buffers use the same technology, we have $m_1 = m_2$.

Since $\alpha > \gamma$ and $\beta > \gamma$, depending on the amounts of data stored in the two buffers, the components X and Y can be turned on/off from time to time to save power without interrupting the smooth operation of Z. Thus the pipeline system operates in one of four possible modes: $S = \{00, 01, 10, 11\}$. These modes are:

- mode 00: X and Y are both off. The total component power is $p_{00} = 0$;
- mode 01: X is off and Y is on. The total component power is $p_{01} = p_y$;
- mode 10: X is on and Y is off. The total component power is $p_{10} = p_x$;
- mode 11: X and Y are both on. The total component power is $p_{11} = p_x + p_y$.

Denote by σ the system mode.

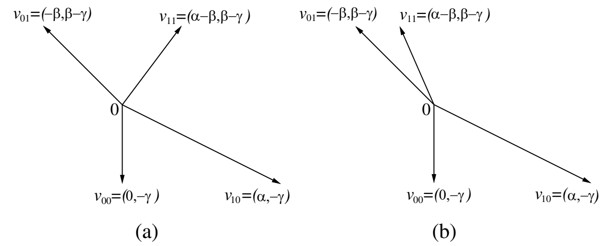

Fig. 2. Vector (q_1, q_2) representing the amounts of data in the two buffer buffers. Their values change according to X's and Y's power states S. Control vectors v_s, $s \in S$. (a) $\alpha > \beta$, (b) $\alpha < \beta$.

Let q_1 be the amount of data stored in the buffer between X and Y, and q_2 be the amount of data stored in the buffer between Y and Z. Depending on the system mode, q_1 and q_2 change at different constant rates. For example, when both X and Y are on, i.e., when the system is in mode 11, q_1 increases at the rate $\alpha - \beta$, and q_2 increases at the rate $\beta - \gamma$. Combining q_1 and q_2 into a vector $q = (q_1, q_2)$, the rates of change of q in mode $\sigma \in S$, denoted by v_σ, are given by:

$$\begin{aligned} v_{00} &= (0, -\gamma), \quad v_{01} = (-\beta, \beta - \gamma), \\ v_{11} &= (\alpha - \beta, \beta - \gamma), \quad v_{10} = (\alpha, -\gamma). \end{aligned} \tag{1}$$

The four vectors in equation (1) are depicted in Fig. 2, with the subfigures (a) and (b) corresponding to the cases when $\alpha > \beta$ and $\alpha < \beta$, respectively.

The mode σ and the variable q together specify the system state of the pipeline of streaming data. Note that $q = (q_1, q_2)$ is a continuous variable taking values in the rectangle $[0, Q_1] \times [0, Q_2]$, while σ is a discrete variable taking values in S. Thus the pipeline system can be modeled naturally as a hybrid system with state $z = (q, \sigma)$. Let $[0, t_f]$ be the time interval during which the average system power consumption needs to be minimized. Then a power management scheme for the pipeline system is given by a mapping $\sigma : [0, t_f] \to S$ so that the power states of the components X and Y at time $t \in [0, t_f]$ are specified by the mode $\sigma(t)$. Typically, there exists a sequence of time intervals $[t_i, t_{i+1})$, $i = 0, \ldots, n-1$, with $t_0 = 0 \leq t_1 \leq \cdots \leq t_n = t_f$, such that the system mode $\sigma(t)$ during the time interval $[t_i, t_{i+1})$ is a constant σ_i, and that immediately adjacent time intervals correspond to different modes. Under this scheme $\sigma(t)$, the continuous state $q(t)$ follows a continuous trajectory given by the differential equation:

$$\frac{dq(t)}{dt} = v_{\sigma(t)}, \quad t \in [0, t_f]. \tag{2}$$

Together, $z(t) = (q(t), \sigma(t))$ forms a (hybrid) solution, or an execution, of the hybrid system.

For a solution $z(t) = (q(t), \sigma(t))$ during the period $[0, t_f]$ of interest, the energy consumption of the whole pipeline system consists of the following three parts:

- static energy consumed by X and Y: $\int_0^{t_f} p_{\sigma(t)}\, dt$;
- energy for changing the power states of the components X and Y: $n_x k_x + n_y k_y$. Here n_x and n_y are the numbers of times that X and Y are turned on, respectively, during $[0, t_f]$;
- buffers' energy consumption: $(m_1 Q_1 + m_2 Q_2) t_f$.

As a result, for the solution $z(t)$, the average power consumption, $\bar{P}$, of the overall system is

$$\bar{P}(z; t_f, Q_1, Q_2) = \frac{1}{t_f}\Bigg[\int_0^{t_f} p_{\sigma(t)}\, dt + n_x k_x + n_y k_y + (m_1 Q_1 + m_2 Q_2) t_f\Bigg].$$

Note that the arguments of $\bar{P}$ are introduced to highlight the dependence of $\bar{P}$ on t_f, Q_1, and Q_2.

The dynamic power management problem is then to find $z(t)$ that can achieve the minimal average power. In particular, if the look ahead time horizon t_f is large enough, the problem becomes:

Problem 1: Find the hybrid trajectories $z(t)$ during $[0, \infty)$ and the buffer sizes Q_1 and Q_2 that minimize the average power $\lim_{t_f \to \infty} \bar{P}(z; t_f, Q_1, Q_2)$.

This is an optimal control problem for the hybrid system modeling the pipeline of streaming data.

III. Periodic Solutions

We will focus on periodic solution $z(t)$ in this paper. For periodic solutions $z(t)$ with period T, the average power equation reduces to

$$\bar{P}(z; T, Q_1, Q_2) = \frac{1}{T}\left(\sum_{\sigma \in S} T_\sigma p_\sigma + n_x k_x + n_y k_y\right) + m_1 Q_1 + m_2 Q_2, \tag{3}$$

where T_σ, $\sigma \in S$, is the amount of time the system is in mode σ in a single period.

We now summarize some useful results on optimal periodic solutions derived previously in [10].

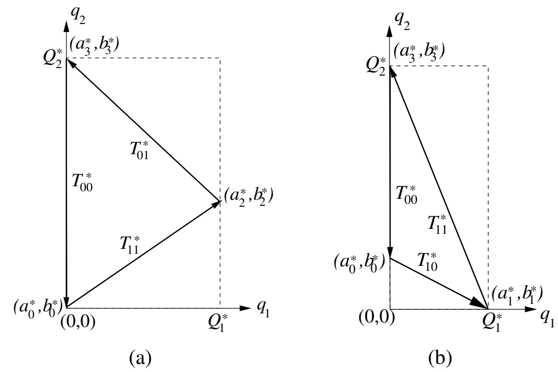

Fig. 3. Optimal trajectory $q^*(t)$, (a) $\alpha > \beta$, (b) $\alpha < \beta$.

A. Scaling of Trajectories

Let $z(t) = (q(t), \sigma(t))$ be a periodic hybrid trajectory with period T for buffer sizes of Q_1 and Q_2. Let $\lambda > 0$ be a positive number. Then, it can be verified that $\lambda q(t/\lambda)$ is a continuous trajectory in $[0, \lambda Q_1] \times [0, \lambda Q_2]$ satisfying equation (2) with $\sigma(t)$ replaced by $\sigma(t/\lambda)$. In other words, $(\lambda q(t/\lambda), \sigma(t/\lambda))$ is a periodic hybrid trajectory with the period λT when the sizes of the two buffers become λQ_1 and λQ_2, respectively. We call $z_\lambda(t) \triangleq (\lambda q(t/\lambda), \sigma(t/\lambda))$ the *scaling* of $z(t) = (q(t), \sigma(t))$ by λ. Compared with the original trajectory, the scaled one has the same switching sequence, and a minimal average power if and only if λ takes the value

$$\lambda^* = \sqrt{\frac{n_x k_x + n_y k_y}{(m_1 Q_1 + m_2 Q_2) T}}. \tag{4}$$

B. Tightness of Optimal $q(t)$ in Q

Since a buffer consumes power proportional to its size, unused buffer memory will unnecessarily waste power. As a result, an optimal trajectory z should fill and empty each of the two buffers at least once sometime during the period, or equivalently, the curve $q(t)$ must touch all four edges of the rectangle $[0, Q_1] \times [0, Q_2]$.

C. Optimal Periodic Solution with $n_x = n_y = 1$

Using the above necessary conditions, we derive in [10] the optimal periodic solutions (see Fig. 3) under the constraint that $n_x = n_y = 1$, i.e., both X and Y turn on and off only once per period. In particular, if $\alpha > \beta$, the optimal solution plotted in Fig. 3(a) has the

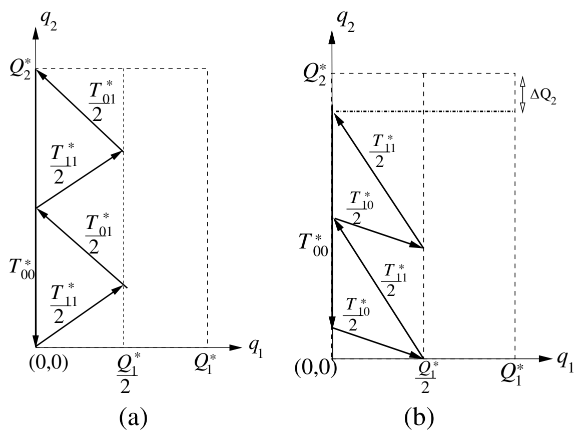

Fig. 4. Examples of integer folding with $n = 2$ for (a) $\alpha > \beta$ and (b) $\alpha < \beta$. Q_1^* and Q_2^* represent the optimal buffer sizes for the $n_x = n_y = 1$ periodic solution.

following parameters:

$$Q_2^* = (\beta - \gamma)\sqrt{\frac{\alpha\gamma(k_x + k_y)}{\beta^2(\alpha - \beta)m_1 + \alpha\beta(\beta - \gamma)m_2}},$$

$$Q_1^* = (\alpha - \beta)\sqrt{\frac{\beta\gamma(k_x + k_y)}{\alpha\beta(\alpha - \beta)m_1 + \alpha^2(\beta - \gamma)m_2}}.$$

While if $\alpha < \beta$, the optimal solution in Fig. 3(b) is:

$$Q_1^* = (\beta - \alpha)\sqrt{\frac{\gamma(k_x + k_y)}{\beta(\beta - \alpha)m_1 + \beta(\beta - \gamma)m_2}},$$

$$Q_2^* = (\beta - \gamma)\sqrt{\frac{\gamma(k_x + k_y)}{\beta(\beta - \alpha)m_1 + \beta(\beta - \gamma)m_2}}.$$

IV. Folding

In this section, we will introduce an operation called folding that may further decrease the system power consumption of an existing switching strategy. We will illustrate this operation through an example. Consider the optimal switching strategy $z^*(t)$ for the case $\alpha < \beta$ and $n_x = n_y = 1$ shown in Fig. 3a. Assume that at time 0, $z^*(t)$ starts from the point $(a_0^*, b_0^*) = (0, 0)$, i.e., with both buffers empty. Then a new strategy $\hat{z}(t)$, shown in Fig. 4a, can be constructed by following exactly the same route of $z^*(t)$ at all times, except that between the time $\frac{T_{11}^*}{2}$ and $T_{11}^* + \frac{T_{01}^*}{2}$, the switching order from mode 11 to mode 01 is reversed. In other words, at time $\frac{T_{11}^*}{2}$, $\hat{z}(t)$ will switch to mode 01 and follow the direction of v_{01} for $\frac{T_{01}^*}{2}$ time; at time $\frac{T_{11}^*}{2} + \frac{T_{01}^*}{2}$, it switches back to mode 11 and follow the direction of v_{11} for $\frac{T_{11}^*}{2}$ time; and then at time $T_{11}^* + \frac{T_{01}^*}{2}$, $\hat{z}(t)$ switches back to mode 01, and follows $z^*(t)$ thereafter. Intuitively, $\hat{z}(t)$ can be thought of as obtained from $z^*(t)$ by (twisting and) folding a corner of the triangular trajectory of $z^*(t)$ inward. Thus we call this operation *folding*. Similarly, the optimal periodic solution for the case $\alpha < \beta$ and $n_x = n_y = 1$ shown in Fig. 3(b) can also be folded, with the resulting strategy shown in Fig. 4b.

Comparing the strategy obtained by folding (e.g., the one in Fig. 4a) with the original one (e.g., the one in Fig. 3a), reveals they have the same period, and consume the same static energy during each period as the proportions of time the system spends in each mode remain unchanged in a period. Although the number k_x of switchings of component X is increased, which results in increased overhead energy, the required buffer size Q_1 is only half of the original Q_1^*. If the increased power from the extra switchings is amortized by the power savings resulting from reduction in buffer size, folding can reduce the power consumption of the system.

The folding operation can also be applied multiple times in a sequence. Fig. 5 shows an example of folding an existing trajectory in five subsequent steps. Note that after the folding, the required buffer size Q_1^* is reduced to $\frac{Q_1^*}{5}$, while the number of switching k_x is increased from 1 to 5. If one folds a given trajectory an integer n times, the buffer size Q_1^* can be reduced to $\frac{Q_1^*}{n}$, and k_x will increase to nk_x.

After folding, the shape of the trajectory is changed, and the new rectangular state space may no longer be of optimal scale as determined by equation (4). Thus one can rescale the state space to get a further improvement in average power consumption. In the following, we will study the power saving by performing the folding and rescaling operations at the same time.

A. Optimal Integer Folding when $\alpha > \beta$

Consider the trajectory in Fig. 3a with $\alpha > \beta$. Integer folding n times affects only the buffer size Q_1 as is demonstrated in Fig. 4a for $n = 2$. The average power of the folded and rescaled trajectory is described by,

$$\bar{P}(\lambda, n) = \lambda(\frac{m_1 Q_1^*}{n_x} + m_2 Q_2^*) + \frac{nk_x + k_y}{\lambda T} + \frac{1}{T}\sum_{s \in S} T_s p_s, \tag{5}$$

where Q_1^* and Q_2^* are the optimal buffer sizes for the $n_x = n_y = 1$ periodic solution, and n is the number of folds with $n = 1$ representing no folds, and λ is the scaling factor. We begin by finding the optimal scaling factor for a fixed n:

$$\frac{\partial \bar{P}(\lambda, n)}{\partial \lambda} = \frac{m_1 Q_1^*}{n} + m_2 Q_2^* - \frac{nk_x + k_y}{\lambda^2 T} = 0,$$

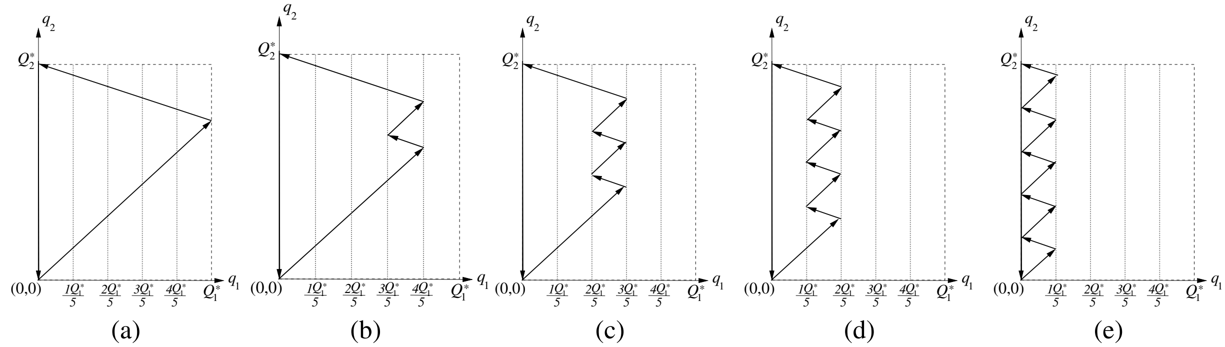

Fig. 5. Series of reversals resulting in an integer fold of 5. At each step, the solution uses $\frac{Q_1^*}{5}$ less buffer space while using k_x more switching power per period. This example does not consider rescaling which is necessary to determine the optimal number of folds.

which yields

$$\lambda_n^* = \sqrt{\frac{nk_x + k_y}{(\frac{m_1 Q_1^*}{n} + m_2 Q_2^*)T}}. \tag{6}$$

Next we replace λ in equation (5) with λ_n^* to obtain

$$\begin{aligned} \bar{P}(\lambda_n^*, n) =& 2\sqrt{\frac{(nk_x + k_y)(\frac{m_1 Q_1^*}{n} + m_2 Q_2^*)}{T}} \\ &+ \frac{1}{T}\sum_{s\in S} T_s p_s. \end{aligned} \tag{7}$$

Taking the derivative of (7) with respect to n yields

$$\frac{d\bar{P}(\lambda_n^*, n)}{dn} = \frac{n^2 k_x m_2 Q_2^* - k_y m_1 Q_1^*}{\sqrt{n^4(nk_x + k_y)(\frac{m_1 Q_1^*}{n} + m_2 Q_2^*)T}}.$$

Thus, the optimal value of n is given by,

$$n^* = \sqrt{\frac{k_y m_1 Q_1^*}{k_x m_2 Q_2^*}}. \tag{8}$$

Note that n^* can only take integer values, and the function $\bar{P}(\lambda_n^*, n)$ is convex with respect to n. Therefore, if n^* given by (8) is not an integer, the optimal feasible value of n^*, $\tilde{n}^*$, is whichever of the two closest integers to n^* that results in the smallest value for equation (7). To sum up,

Proposition 1 (Optimal Folding for the $\alpha > \beta$ Case): Suppose that $\alpha > \beta$. Then for the optimal periodic solution under the constraint $n_x = n_y = 1$ shown in Fig. 3a, the optimal number of folding is given by

$$\tilde{n}^* = \underset{n\in\{\lfloor n^*\rfloor, \lceil n^*\rceil\}}{\operatorname{argmin}} \bar{P}(\lambda_n^*, n),$$

where n^* is defined in (8), and the optimal rescaling factor is given by (6) with n replaced by $\tilde{n}^*$.

As a result, if the value of $\tilde{n}^*$ is greater than one, integer folding can indeed save more power than the optimal solution under the constraint $n_x = n_y = 1$.

B. Optimal Integer folding when $\alpha < \beta$

In this case, folding affects the buffer size Q_2 as well as Q_1, as can be seen in Fig. 4b. The average power as a function of the number n of folds and the rescaling factor λ is more complicated, and can be verified to be

$$\begin{aligned} \bar{P}(\lambda, n) =& \lambda(\frac{m_1 Q_1^*}{n} + m_2(Q_2^* - \frac{\gamma(n-1)Q_1^*}{\alpha n})) \\ &+ \frac{nk_x + k_y}{\lambda T} + \frac{1}{T}\sum_{s\in S} T_s p_s, \end{aligned} \tag{9}$$

where again Q_1^* and Q_2^* are the optimal buffer sizes for the $n_x = n_y = 1$ case shown in Fig. 3b. Following the same process as before, we obtain

$$\lambda_n^* = \sqrt{\frac{nk_x + k_y}{\left(\frac{m_1 Q_1^* + \frac{\gamma}{\alpha} m_2 Q_1^*}{n} + m_2 Q_2^* - \frac{\gamma}{\alpha} m_2 Q_1^*\right) T}}, \tag{10}$$

$$n^* = \sqrt{\frac{(m_1 + \frac{\gamma}{\alpha} m_2) k_y}{(m_2 \frac{Q_2^*}{Q_1^*} - \frac{\gamma}{\alpha} m_2) k_x}}. \tag{11}$$

Again, $\bar{P}(\lambda_n^*, n)$ is convex in n; so the optimal integer value $\tilde{n}^*$ lies is one of the two closest integers to n^*.

Proposition 2 (Optimal Folding for the $\alpha < \beta$ Case): Suppose that $\alpha < \beta$. Then for the optimal periodic solution under the constraint $n_x = n_y = 1$ shown in Fig. 3b, the optimal number of folds is given by

$$\tilde{n}^* = \underset{n\in\{\lfloor n^*\rfloor, \lceil n^*\rceil\}}{\operatorname{argmin}} \bar{P}(\lambda_n^*, n),$$

where n^* is defined in (11), and the optimal rescaling factor is given by (10) with n replaced by $\tilde{n}^*$.

C. Discussion and Possible Extensions

Examining the folding equations reveals several aspects of the nature of folding. For example, equations (8)

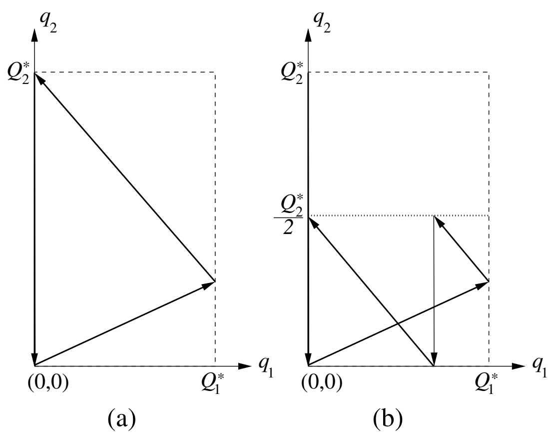

Fig. 6. Examples of folding across a horizontal line. (a)The original solution for $n_x = n_y = 1$. (b)The solution folded across a horizontal line. the value of n_x is still 1, but now $n_y = 2$

and (11) show that lower switching power in element X and higher memory cost in buffer Q_1 allows for more folds, which is intuitively easy to understand.

Equations (6) and (10) are monotonically increasing with respect to n for $n > 1$. This implies that for any number of folds, the scaling factor is always greater than one. In other words, for any number of folds, buffer size Q_2 will always need to be increased compared with the original one with no folds. Thus the new state space after the folding and rescaling is thinner and taller than the state space of the original unfolded trajectory.

In this paper we only consider folding across a vertical line. However, it is also possible to fold horizontally as shown in Fig. 6. In this case, the number of switchings for component Y is 2, while the number of switchings for component X remains one in a single period. Furthermore, it is worthwhile to consider simultaneous foldings in both horizontal and vertical directions.

Another natural extension is to consider systems consisting of more than three components and two buffers interconnected by a network (graph). In this case, the number of system modes will grow exponentially with the number of buffers. Thus techniques such as matrix optimization must be utilized to find approximate solutions to such problems. Research is currently being carried out along this direction.

V. Conclusion

This paper studies the dynamic power management problem of a pipeline of data streaming through three components and two buffers. We extend our previous work to consider the more general situation where solutions are allowed to switch on and off multiple times per period by introducing the operation of folding. We derive the condition under which folding can lower the system power consumption. The optimal number of folds and the optimal rescaling factor are derived analytically.

VI. Acknowledgment

The authors would like to thank the anonymous reviewers for their helpful comments.

References

[1] R. Alur, S. L. Torre, and G. Pappas. Optimal paths in weighted timed automata. In M. D. Benedetto and A. Sangiovanni-Vincentelli, editors, *Hybrid Systems: Computation and Control, 4th Int. Workshop*, volume 2034 of *Lecture Notes in Computer Science*, pages 49–62. Springer-Verlag, Rome, Italy, 2001.

[2] A. Balluchi, L. Benvenuti, M. D. D. Benedetto, G. M. Miconi, U. Pozzi, T. Villa, H. Wong-Toi, and A. L. Sangiovanni-Vincentelli. Maximal safe set computation for idle speed control of an automotive engine. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control, 3rd Int. Workshop (HSCC 2000)*, volume 1790 of *Lecture Notes in Computer Science*, pages 32–44. Springer Verlag, 2000.

[3] M. Baotić, F. J. Christophersen, and M. Morari. A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. In *Proc. of the European Control Conference*, Cambridge, UK, Sept. 2003.

[4] L. Benini and G. D. Micheli. System-Level Power Optimization: Techniques and Tools. *ACM Transactions on Design Automation of Electronic Systems*, 5(2):115–192, April 2000.

[5] L. Cai and Y.-H. Lu. Dynamic Power Management Using Data Buffers. In *Design Automation and Test in Europe*, pages 526–531, 2004.

[6] L. Cai and Y.-H. Lu. Energy Management Using Buffer Memory for Streaming Data. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(2):141–152, February 2005.

[7] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal control of switching times in switched dynamical systems. In *Proc. 42nd IEEE Int. Conf. on Decision and Control*, volume 3, pages 2138–2143, Maui, HI, Dec. 2003.

[8] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proc. 38th IEEE Int. Conf. on Decision and Control*, pages 3972–3977, Phoenex, AZ, Dec. 1999.

[9] J. Hespanha. A model for stochastic hybrid systems with application to communication networks. *To appear in Nonlinear Analysis, Special Issue on Hybrid Systems*, 2005.

[10] J. Hu and Y.-H. Lu. Buffer Management for Power Reduction Using Hybrid Control. In *IEEE Conference on Decision and Control and European Control Conference*, 2005.

[11] N. Pettis, L. Cai, and Y.-H. Lu. Dynamic Power Management for Streaming Data. In *International Symposium on Low Power Electronics and Design*, pages 62–65, 2004.

[12] H. Sussmann. A maximum principle for hybrid optimal control problems. In *Proc. 38th IEEE Int. Conf. on Decision and Control*, pages 425–430, Phoenex, AZ, Dec. 1999.

[13] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for air traffic management: A study in multi-agent hybrid systems. *IEEE Trans. on Automatic Control*, 43:509–521, 1998.

[14] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Trans. Automatic Control*, 38(2):195–207, 1993.

[15] X. Xu and P. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. Automatic Control*, 49(1):2–16, Jan. 2004.