Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Optimal Dynamic Bit Assignment in Second-order Noise-free Quantized Linear Control Systems

Qiang Ling and Michael D. Lemmon

***Abstract*— This paper introduces a dynamic bit assignment policy (DBAP) for second-order quantized feedback control systems without process or measurement noise. The proposed DBAP is a constant bit rate policy based on a similar policy analyzed in [1]. We prove that the new policy is optimal for diagonalizable systems in the sense of minimizing the summed square quantization error subject to a fixed number of quantization bits.**

I. Introduction

Consider the following discrete-time system,

$$\begin{aligned} x[k+1] &= Ax[k] + Bu[k] \\ u[k] &= Fx^q[k] \end{aligned} \tag{1}$$

where $x[k] \in \Re^2$ is the system state at time k, $x^q[k] \in \Re^2$ is a *quantized* version of that state, $u[k] \in \Re^m$ is the control at time k and A, B, and F are real-valued matrices of appropriate dimension. With regard to the preceding system, we make the following assumptions:

1) (A, B) is controllable and F is a stabilizing state feedback gain matrix.
2) A is diagonalizable and for simplicity we assume $A = \mathrm{diag}(\lambda_1, \lambda_2)$ where $\lambda_i > 1$ for $i = 1, 2$.
3) At every time step the system state, $x[k]$ is quantized into Q bits (fixed length coding) to generate the quantized state, $x^q[k]$.

The policy used in generating the quantized state $x^q[k]$ is called a **quantization policy**.

This paper asks and answers the following question; *what is the optimal "performance" achievable under a fixed number of quantization bits?*. In this paper the constant bit rate policy is characterized by the number of bits $b_i[k]$ assigned at time k to represent the ith component of the state vector $x[k]$. We measure performance with respect to the summed square quantization error over a finite horizon of N steps,

$$P_N = \sup_{x[0] \in x^q[0] + U[0]} \sum_{k=1}^{N} (|e_1[k]|^2 + |e_2[k]|^2) \tag{2}$$

where $U[0] \subset \Re^n$ is a bounded set centered at the origin, $e_i[k]$ is the ith component of the quantization error vector $e[k] = x[k] - x^q[k]$ and the supremum is taken over all possible initial states $x[0]$. This paper constructs a quantization policy, named dynamic bit assignment policy (DBAP), which minimizes the performance measure P_N subject to a fixed number of quantization bits, Q. All proofs are in the appendix.

The authors gratefully acknowledge the partial financial support of the National Science Foundation (ECS02-25265) and the Army Research Office (DAAD19-01-1-0743). Qiang Ling is with Seagate Research, Pittsburgh, PA 15222, USA. Michael Lemmon is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA.

II. Background

We may categorize quantization policies as either being memoryless or having memory. *Memoryless policies* map each bit to a specific subset of the state space such that the assignment is fixed for all time. The attraction of memoryless policies is the simplicity of their coding/decoding schemes. The main drawback of memoryless policies is that they require an infinite number of quantization bits to ensure asymptotic stability [2]. Elia and Mitter [3] derived the lowest quantization density for asymptotic stability with an infinite number of quantization bits. But with only a finite number of quantization bits, the best we can guarantee is ultimate boundedness of the state [4] [5] [6].

Quantization policies with memory (so-called *dynamic quantization policies*) have been shown to achieve asymptotic stability with a finite number of quantization bits [7]. These policies generate a sequence, $\{P[k]\}$, of *uncertainty sets*. It is presumed that $x[k]$ lies inside the set $P[k]$ at time k. The next uncertainty set is generated by first partitioning $P[k]$ into M smaller rectangles which we denote as $P_i[k]$ for $i = 1, \ldots, M$. If $x[k]$ lies in the set $P_j[k]$, then the index j is transmitted to the decoder and this set is propagated through the plant's dynamics to obtain the next uncertainty set $P[k+1]$. If this sequence of uncertainty sets converges to 0, then the system is asymptotically stable. Brockett and Liberzon [7] established sufficient conditions for asymptotic stability that were later tightened in [8].

The work in [7] was significantly extended by Tatikonda in [9] [10]. This work established necessary and sufficient conditions on general linear systems that characterize the minimum number of quantization bits required for asymptotic stability under time-varying bit rates. Related work was published in [11] for diagonalizable systems. Similar bounds on the minimumm number of quantization bits were also established in [12] for general linear systems in the stochastic sense. The aforementioned quantization policies presume time-varying bit rates, which are not desirable in real networks due to power and bandwidth inefficiency [13]. A necessary and sufficient condition for asymptotic stability under constant bit rates was established in [1].

The proofs in all of the aforecited works use constructive methods to guarantee asymptotic convergence of the noise-free quantized linear system. By constructive, we mean that these proofs construct a specific dynamic quantization policy

0-7803-9568-9/05/$20.00 ©2005 IEEE

that achieves the specified quantization bound. These policies vary considerably in their bit assignment policies. Let Q denote the number of bits to be assigned and let $b_i[k]$ denote the number of bits that a quantization policy uses to encode the ith commponent of the state $x[k]$. There are a number of bit assignment policies in the open literature that we refer to as being either *static*, *periodic*, *switching* or *dynamic*. *Static bit assignment policies* choose $b_i[k] = b_i$ $(i = 1, 2)$. It is proven [9] [10] that $\lim_{k\to\infty} L_i[k] = 0$ ($L_i[k]$ is an upper bound of the quantization error $e_i[k]$) if and only if $b_i > \log_2(\lambda_i)$ $(i = 1, 2)$ where λ_i are the unstable eigenvalues of a noise-free system. A special static policy with $b_1[k] = b_2[k] = b$ is considered in [7] [8]. *Periodic bit assignment policies* choose $b_i[k]$ such that $b_i[k] = b_i[k+lT]$ for all integers k and l where T is the *period*. The average bit rate for such policies is defined as $\overline{b}_i = \frac{1}{T}\sum_{k=0}^{T-1} b_i[k]$ and in [10] it is shown that $\overline{b}_i$ can approach $\log_2(\lambda_i)$ arbitrarily closely thereby ensuring that $L_i[k]$ converges exponentially to 0. A periodic policy for output quantization is considered in [14].

Switching bit assignment policies assign all Q bits to either $b_1[k]$ or $b_2[k]$ depending upon $P[k]$[1]. It is proven in [1] that $L_i[k]$ converges exponentially to zero if and only if $Q > \sum_{i=1}^{2} \log_2(\lambda_i)$.

While all of the above bit assignment strategies ensure asymptotic stability, these strategies are not equal. These policies differ in their convergence rates and ultimately in the performance they exhibit. This then brings us to the problem considered in this paper; namely *"What bit assignment policy assures asymptotic stability while optimizing some specified measure of the control system's performance?"*. The main result in this paper shows that a variation on the dynamic bit assignment policy used in [1] is indeed optimal in the sense of minimizing the summed square quantization error.

III. Dynamic Bit Assignment Policy

This paper studies a quantized feedback control system, which is shown in figure 1. The plant is a discrete-time linear

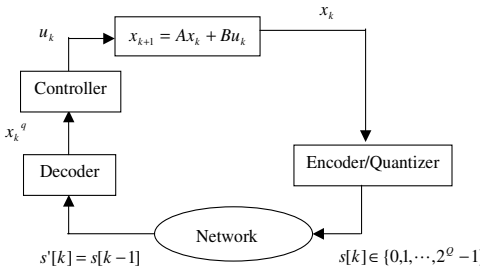

Fig. 1. Quantized feedback control system

system whose state equations are shown in equation 1. The state $x[k] \in \Re^n$ is quantized and encoded into a symbol $s[k]$ from a discrete set $\{0, 1, \cdots, 2^Q - 1\}$. Throughout this paper, the terms "quantizer" and "encoder" are used interchangeably. $s[k]$ is transmitted to the decoder over a communication network. We assume the network has one step delay. So the symbol received by the decoder, $s'[k]$, is a one-step delayed version of $s[k]$, i.e. $s'[k] = s[k-1]$. The decoder uses the received symbols to compute an estimate, $x^q[k]$, of the plant's true state, $x[k]$. The controller uses this estimate, $x^q[k]$ to compute the control signal $u[k]$.

The quantization method used in this paper originates in the uncertainty set evolution method introduced in [7] and [9]. This approach presumes that the encoder and decoder agree that the state lies within the set

$$x[k] \in x^q[k] + U[k], \forall k \geq 0. \tag{3}$$

In this paper we restrict our attention to a two dimensional system so that the uncertainty set may be characterized as

$$\begin{aligned} U[k] &= \text{rect}(L_1[k], L_2[k]) \\ &= [-L_1[k], L_1[k]] \times [-L_2[k], L_2[k]]. \end{aligned}$$

In this equation $L_1[k]$ and $L_2[k]$ are non-negative and they represent the half-length of the sides of the rectangular set $U[k]$. We define the *quantization error* as $e[k] = x[k] - x^q[k] = [e_1[k], e_2[k]]^T$. It is obvious that $L_1[k]$ and $L_2[k]$ also represent the upper bounds of the quantization errors $e_1[k]$ and $e_2[k]$. Just prior to time k we know that $e[k] \in U[k]$ where we refer to $U[k]$ as the *uncertainty set* at time k. We then partition both sides of $U[k]$. The first side, $L_1[k]$, is partitioned into $2^{b_1[k]}$ equal parts and the second side, $L_2[k]$, is partitioned into $2^{b_2[k]}$ equal parts. We impose a constant bit rate constraint on our bit assignment which requires that

$$b_1[k] + b_2[k] = Q \tag{4}$$

for all k. After a new measurement of the state $x[k]$ is made, then the encoder knows that

$$x[k] \in x^q_{s[k]}[k] + U_{s[k]}[k]$$

where $x^q_{s[k]}[k]$ is the center of the smaller subset and

$$U_{s[k]}[k] = \text{rect}\left(\frac{L_1[k]}{2^{b_1[k]}}, \frac{L_2[k]}{2^{b_2[k]}}\right)$$

The index, $s[k]$, for this smaller subset is transmitted across the channel and the decoder reconstructs the state at time $k+1$ using the equations

$$\left\{\begin{array}{rcl} x[k+1] & \in & x^q[k+1] + U[k+1] \\ U[k+1] & = & \text{rect}(L_1[k+1], L_2[k+1]) \\ x^q[k+1] & = & Ax^q_{s[k]}[k] + Bu[k] \\ u[k] & = & Fx^q[k] \\ L_1[k+1] & = & \frac{\lambda_1}{2^{b_1[k]}} L_1[k] \\ L_2[k+1] & = & \frac{\lambda_2}{2^{b_2[k]}} L_2[k] \end{array}\right. \tag{5}$$

The choice for $b_i[k]$ $(i = 1, 2)$ represents a *bit assignment policy*. With the requirement that $b_1[k] + b_2[k] = Q$, we're confining our attention to constant bit rate quantization schemes. The motivation for doing this is that many communication systems work best under a constant bit rate [13]. There may be many bit assignment policies that satisfy the necessary and sufficient conditions for asymptotic stability in [1]. We're interested in constructing a bit assignment policy that is *optimal* with respect to a specified measure

of the feedback control system's performance. In this paper we choose the performance measure in equation 2 where the supremum is taken over all $x[0] \in x^q[0] + U[0]$. Note that by definition, $|e_i[k]| \leq L_i[k]$ for $i = 1, 2$ and for any $x[0]$. This inequality becomes equality for the specific $x[0]$, e.g. $x[0] = x^q[0] + [L_1[0], L_2[0]]^T$, that maximizes the sum $\sum_{k=1}^{N} e_1^2[k] + e_2^2[k]$, which means that P_N in equation 2 may be rewritten as

$$P_N = \sum_{k=1}^{N} (L_1^2[k] + L_2^2[k]) \tag{6}$$

For a given number of quantization bits, Q, the objective is to find $b_i[k]$ $(i = 1, 2)$ that minimize the P_N given in equation 6.

This paper proposes a variation on the switching dynamic policy found in [1] that we call *dynamic bit assignment policy* or **DBAP**. DBAP is a recursive algorithm that generates $b_i[k]$ as follows.

Algorithm 3.1: **Dynamic Bit Assignment Policy**

1) *Initialize* $b_1[k] = 0$ and $b_2[k] = 0$,
   and set $L_1 = \lambda_1 L_1[k]$ and $L_2 = \lambda_2 L_2[k]$.
2) *For* $q = 1$ to Q
   $I = argmax_{i \in \{1,2\}} L_i$.
   $b_I[k] := b_I[k] + 1$ and $L_I = L_I/2$.

The following lemma provides a closed form characterization of $b_2[k]$ generated by DBAP. The other bit assigment is $b_1[k] = Q - b_2[k]$ under our constant bit rate constraint.

Lemma 3.1: Under DBAP,

$$b_2[k] = \begin{cases} 0, & \frac{1}{2^{Q+1}} \lambda_1 L_1[k] \geq \lambda_2 L_2[k] \\ Q, & \frac{1}{2^{-Q-1}} \lambda_1 L_1[k] \leq \lambda_2 L_2[k] \\ \left\lceil \frac{1}{2} \left(Q - \log_2 \left(\frac{\lambda_1 L_1[k]}{\lambda_2 L_2[k]} \right) \right) \right\rceil, & otherwise \end{cases} \tag{7}$$

where $\lceil \cdot \rceil$ is defined as $\lceil x \rceil = \lceil x - 0.5 \rceil$.

IV. Optimal Dynamic Bit Assignment

This section characterizes the bit assignment policy that minimizes the performance index, P_N, in equation 6. Our optimization problem is formally stated as follows,

$$\begin{array}{ll} \min_{\{b_1[k], b_2[k]\}_{k=0}^{N-1}} & \sum_{k=1}^{N} (L_1^2[k] + L_2^2[k]) \\ \text{subject to} & b_1[k] + b_2[k] = Q, \end{array} \tag{8}$$

where $b_1[k], b_2[k] \in \mathcal{N}$. Let $\mathbf{b} = \{b_1[j], b_2[j]\}_{j=0}^{N-1}$ denote the optimal solution to this problem. We will determine this solution by first considering a sequence of simpler problems and then show that the solutions to these simpler problems also solve the original problem and furthermore that they are generated by the proposed DBAP.

Consider the following sequence of minimization problems indexed by k for $k = 1, \ldots, N$.

$$\begin{array}{ll} \min_{\{b_1[j], b_2[j]\}_{j=0}^{k-1}} & (L_1^2[k] + L_2^2[k]) \\ \text{subject to} & b_1[j] + b_2[j] = Q, \end{array} \tag{9}$$

where $b_1[j], b_2[j] \in \mathcal{N}$. The solution to the kth subproblem will be denoted as $\mathbf{b}^{(k)} = \{b_1^{(k)}[j], b_2^{(k)}[j]\}_{j=0}^{k-1}$. The following lemma establishes the basic relationship between subproblems 9 and the original problem 8. In the following lemma, we say $\mathbf{b}^{(k-1)} \subset \mathbf{b}^{(k)}$ if and only if $b_i^{(k-1)}[j] = b_i^{(k)}[j]$ for $j < k - 1$. Essentially this means that $\mathbf{b}^{(k-1)}$ is a prefix of $\mathbf{b}^{(k)}$.

Lemma 4.1: If $\{\mathbf{b}^{(k)}\}_{k=1}^{N}$ solves the sequence of subproblems 9 such that $\mathbf{b}^{(k-1)} \subset \mathbf{b}^{(k)}$ for $k = 2, \ldots, N$, then $\mathbf{b}^{(N)}$ solves the original problem 8.

Rather than directly solving subproblem 9, we consider a *relaxed* problem of the form

$$\begin{array}{ll} \min_{s_1[k], s_2[k]} & \left(\frac{\lambda_1^k}{2^{s_1[k]}} L_1[0] \right)^2 + \left(\frac{\lambda_2^k}{2^{s_2[k]}} L_2[0] \right)^2 \\ \text{subject to} & s_1[k] + s_2[k] = kQ \end{array} \tag{10}$$

where $s_1[k], s_2[k] \in \mathcal{N}$. In these relaxed problems, we interpret $s_i[k]$ as the number of bits used to represent the ith component of the state up to time k. In other words, we let $s_i[k] = \sum_{j=0}^{k-1} b_i[j]$. Let $\mathbf{s}^{(k)} = \{s_1[k], s_2[k]\}$ denote the solution to the kth relaxed subproblem. Note that $L_i[k] = \frac{\lambda_i^k}{2^{s_i[k]}} L_i[0]$ $(i = 1, 2)$ by eq. 5. So subproblems 9 and 10 have the same performance index. In subproblems 10, the constant bit rate constraint (equation 4) implies that the summed numbers of bits satisfy,

$$s_1[k] + s_2[k] \quad = \quad kQ \tag{11}$$

So this problem relaxes problem 9 by only minimizing the cost index with respect to the bit sum, rather than the individual history of assigned bits. The following lemma states the solution for problem 10.

Lemma 4.2: The solution to the kth problem in equation 10 is

$$s_1[k] \quad = \quad kQ - s_2[k] \tag{12}$$

$$s_2[k] \quad = \quad \begin{cases} 0, & \frac{\lambda_1^k}{2^{kQ+1}} L_1[0] \geq \lambda_2^k L_2[0] \\ kQ, & \frac{\lambda_1^k}{2^{-kQ-1}} L_1[0] \leq \lambda_2^k L_2[0] \\ \left\lceil \frac{1}{2} \left(kQ - \log_2 \left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]} \right) \right) \right\rceil, & otherwise \end{cases} \tag{13}$$

It is important to note a similarity between equation 13 in lemma 4.2 and the characterization of the bit assignment generated by DBAP in equation 7 in lemma 3.1. The following theorem formalizes this relationship by asserting that the sequence of summed bits, $\mathbf{s}^{(k)}$, generated by DBAP indeed solve the relaxed problem 10 while enforcing the additional reqirements that $b_1[k] + b_2[k] = Q$ and $s_i[k] = \sum_{i=1}^{k-1} b_i[k]$. These additional constraints are precisely those that were relaxed in going from problem 9 to 10, so DBAP also solves the original sequence of subproblems in equation 9.

Lemma 4.3: Let $b_i[k]$ denote the bit sequence generated by the proposed DBAP. If we let

$$s_i[k] = \sum_{j=0}^{k-1} b_i[j], \ i = 1, 2$$

then $\mathbf{s}^{(k)} = \{s_1[k], s_2[k]\}$ also solves the kth relaxed minimization problem in equation 10.

Based on Lemmas 4.1, 4.2 and 4.3, we establish the optimality of our proposed DBAP for noise-free quantized linear systems.

Theorem 4.4: Dynamic bit assignment policy (DBAP) generates a bit assignment sequence that solves optimization 8.

V. Conclusions

This paper studies the optimal bit assignment policy for a second-order linear system over a finite horizon. It is an extension of the study of scalar quantized system in [9]. The 2-dimension and diagonalizablity assumptions in this paper, however, limit its generality. Further research to relax these assumptions is under investigation.

VI. Appendix

This section uses the following notation to represent the ratio of $L_1[k]$ and $L_2[k]$.

$$\gamma[k] = \frac{L_1[k]}{L_2[k]} \tag{14}$$

A. Proof of Lemma 3.1

We prove this lemma by using mathematical induction on Q. When $Q = 1$, Lemma 3.1 trivially holds.

Suppose Lemma 3.1 holds for $Q = Q_1$. We try to prove it also holds for $Q = Q_1 + 1$. In order to emphasize the dependence of $b_2[k]$ on Q_1, $L_1[k]$ and $L_2[k]$, we denote $b_2[k]$ as $b_2[k](Q_1, L_1[k], L_2[k])$.

Now we compute $b_2[k](Q_1 + 1, L_1[k], L_2[k])$. Based on $\gamma[k]$, there are 3 kinds of decisions on $b_2[k]$.

a) $\gamma[k] \geq \frac{\lambda_2}{\lambda_1} 2^{(Q_1+1)+1}$*:* Following the procedure in algorithm 3.1, we find out $b_2[k] = 0$, which satisfies eq. 7, i.e. Lemma 3.1 holds for that case.

b) $\gamma[k] \leq \frac{\lambda_2}{\lambda_1} 2^{-(Q_1+1)-1}$*:* Following the procedure in algorithm 3.1, we find out $b_2[k] = Q_1 + 1$, which satisfies eq. 7, i.e. Lemma 3.1 holds for that case.

c) $\frac{\lambda_2}{\lambda_1} 2^{-(Q_1+1)-1} < \gamma[k] < \frac{\lambda_2}{\lambda_1} 2^{(Q_1+1)+1}$*:* The case can be further categorized into two sub-cases, $\lambda_1 L_1[k] \geq \lambda_2 L_2[k]$ and $\lambda_1 L_1[k] < \lambda_2 L_2[k]$.

If $\lambda_1 L_1[k] \geq \lambda_2 L_2[k]$, the first bit will be assigned to $L_1[k]$ by algorithm 3.1. So

$$\begin{aligned} & b_2[k]\left(Q_1 + 1, L_1[k], L_2[k]\right) \\ = \;& b_2[k]\left(Q_1, \frac{L_1[k]}{2}, L_2[k]\right) \end{aligned} \tag{15}$$

By $\gamma[k] < \frac{\lambda_2}{\lambda_1} 2^{(Q_1+1)+1}$ and $\lambda_1 L_1[k] \geq \lambda_2 L_2[k]$, we get

$$2^{-Q_1-1} < 2^{-1} < \frac{\lambda_1 \frac{L_1[k]}{2}}{\lambda_2 L_2[k]} < 2^{Q_1+1} \tag{16}$$

By the assumption that Lemma 3.1 holds for $Q = Q_1$, we get

$$\begin{aligned} & b_2[k](Q_1, \frac{L_1[k]}{2}, L_2[k]) \\ = \;& \left[\frac{1}{2}\left(Q_1 - \log_2\left(\frac{\lambda_1 \frac{L_1[k]}{2}}{\lambda_2 L_2[k]}\right)\right)\right] \\ = \;& \left[\frac{1}{2}\left((Q_1+1) - \log_2\left(\frac{\lambda_1 L_1[k]}{\lambda_2 L_2[k]}\right)\right)\right] \end{aligned} \tag{17}$$

Substituting eq. 17 into eq. 15 yields

$$\begin{aligned} & b_2[k](Q_1 + 1, L_1[k], L_2[k]) \\ = \;& \left[\frac{1}{2}\left((Q_1+1) - \log_2\left(\frac{\lambda_1 L_1[k]}{\lambda_2 L_2[k]}\right)\right)\right] \end{aligned}$$

The above expression on $b_2[k]$ agrees with eq. 7. So Lemma 3.1 holds for that sub-case.

If $\lambda_1 L_1[k] < \lambda_2 L_2[k]$, the first bit will be assigned to $L_2[k]$ by algorithm 3.1. So

$$\begin{aligned} & b_2[k]\left(Q_1 + 1, L_1[k], L_2[k]\right) \\ = \;& 1 + b_2[k]\left(Q_1, L_1[k], \frac{L_2[k]}{2}\right) \end{aligned}$$

We can compute $b_2[k]\left(Q_1, L_1[k], \frac{L_2[k]}{2}\right)$ in a similar manner to show that the achieved expression on $b_2[k](Q_1 + 1, L_1[k], L_2[k])$ satisfies eq. 7.

Because Lemma 3.1 holds for both sub-cases, it holds for $\frac{\lambda_2}{\lambda_1} 2^{-(Q_1+1)-1} < \gamma[k] < \frac{\lambda_2}{\lambda_1} 2^{(Q_1+1)+1}$.

Because Lemma 3.1 holds for all three cases on $\gamma[k]$, Lemma 3.1 holds for $Q = Q_1 + 1$. Together with the assumption that Lemma 3.1 holds for $Q = Q_1$ and the fact that Lemma holds for $Q = 1$, we know Lemma 3.1 holds for all $Q \geq 1$. $\diamondsuit$

B. Proof of Lemma 4.1

We use P^* and $P^{(k)*}$ to denote the optimal performance of problem 8 and the kth subproblem in equation 9 respectively.

It is straightforward to see that

$$\begin{aligned} & \min_{\{b_1[k], b_2[k]\}_{k=0}^{N-1}} \sum_{k=1}^{N} (L_1^2[k] + L_2^2[k]) \\ \geq \;& \sum_{k=1}^{N} \min_{\{b_1[j], b_2[j]\}_{j=0}^{N-1}} (L_1^2[k] + L_2^2[k]) \qquad (18) \\ = \;& \sum_{k=1}^{N} \min_{\{b_1[j], b_2[j]\}_{j=0}^{k-1}} (L_1^2[k] + L_2^2[k]) \qquad (19) \end{aligned}$$

The equality in eq. 19 comes from the fact that $L_1[k]$ and $L_2[k]$ are independent of $\{L_1[j], L_2[j]\}_{j=k}^{N-1}$ due to the causal updating rule in eq. 5. Note that all $\min$ operations in the above equations are performed under the constraint of $b_1[j] + b_2[j] = Q$ $(j = 0, \cdots, N-1)$. Considering the definitions of P^* and $P^{(k)*}$, eq. 18 and 19 can be rewritten into

$$P^* \geq \sum_{k=1}^{N} P^{(k)*} \tag{20}$$

As stated in Lemma 4.1, $\mathbf{b}^{(k-1)} \subset \mathbf{b}^{(k)}$ $(k = 2, \cdots, N)$. So the performance of the kth problem in eq. 9 under $\mathbf{b}^{(N)}$ is

$$L_1^2[k] + L_2^2[k] = P^{(k)*} \tag{21}$$

Summing eq. 21 for $k = 1, \cdots, N$ yields

$$\sum_{k=1}^{N} L_1^2[k] + L_2^2[k] = \sum_{k=1}^{N} P^{(k)*} \tag{22}$$

Because $\mathbf{b}^{(N)}$ satisfies the constraint of problem 8, i.e. $b_1^{(N)}[k]+b_2^{(N)}[k]=Q$ $(k=0,\cdots,N-1)$, $\mathbf{b}^{(N)}$ is a feasible solution to problem 8. By eq. 22, the performance of problem 8 under $\mathbf{b}^{(N)}$ is $\sum_{k=1}^{N} P^{(k)*}$. By the optimality of P^*, we obtain

$$P^* \leq \sum_{k=1}^{N} P^{(k)*} \tag{23}$$

Combining eq. 20 and 23 yields

$$P^* = \sum_{k=1}^{N} P^{(k)*} \tag{24}$$

By the feasibility of $\mathbf{b}^{(N)}$ and eq. 22 and 24, we know $\mathbf{b}^{(N)}$ solves the original problem 8. $\diamondsuit$

C. Proof of Lemma 4.2

The performance index in problem 10 is the summation of two terms, $\left(\frac{\lambda_1^k}{2^{s_1[k]}}L_1[0]\right)^2$ $(=L_1^2[k])$ and $\left(\frac{\lambda_2^k}{2^{s_2[k]}}L_2[0]\right)^2$ $(=L_2^2[k])$. We know the product of the two terms is independent of $s_1[k], s_2[k]$ due to the constraint $s_1[k]+s_2[k]=kQ$.

$$L_1^2[k]L_2^2[k] = \left(\frac{\lambda_1^k\lambda_2^k}{2^{kQ}}L_1[0]L_2[0]\right)^2 \tag{25}$$

This structure reminds us the following lemma.

Lemma 6.1: If $x, y > 0$ and $xy=\beta$, then

$$x+y = 2\sqrt{\beta}g(|\log_2(x/y)|) \tag{26}$$

where $g(\alpha)=0.5\left(\sqrt{2^\alpha}+\frac{1}{\sqrt{2^\alpha}}\right)$.

The proof of Lemma 6.1 is straightforward and omitted here.

By its definition, we know $g(\alpha)$ is strictly increasing for $\alpha \geq 0$. Apply this lemma to $L_1^2[k]+L_2^2[k]$ with eq. 25 considered, we get

$$L_1^2[k]+L_2^2[k] \quad = \quad 2Cg(2|\log_2(L_1[k]/L_2[k])|) \tag{27}$$

where $C=\frac{\lambda_1^k\lambda_2^k}{2^{kQ}}L_1[0]L_2[0]$. In order to minimize $L_1^2[k]+L_2^2[k]$, we have to minimize $|\log_2(L_1[k]/L_2[k])|$, i.e. keeping $L_1[k]$ and $L_2[k]$ as balanced as possible. By the expression of $L_i[k]=\frac{\lambda_i^k}{2^{s_i[k]}}L_i[0]$ $(i=1,2)$, we know

$$\begin{aligned} & \log_2(L_1[k]/L_2[k]) \\ = \quad & \log_2\left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right) - (s_1[k]-s_2[k]) \\ = \quad & \log_2\left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right) - kQ + 2s_2[k] \end{aligned}$$

The second equality shown above comes from the constraint $s_1[k]+s_2[k]=kQ$. $s_2[k]$ is an integer between 0 and Qk. The minimization of $|\log_2(L_1[k]/L_2[k])|$ may be formally expressed as

$$\begin{aligned} & \min_{s_2[k]} \left|\log_2\left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right) - kQ + 2s_2[k])\right| \\ & s.t. \qquad s_2[k] \in \{0,1,\cdots,kQ\} \end{aligned} \tag{28}$$

It is straightforward to show that the solution to optimization 28 is exactly eq. 13. By the strictly increasing property of $g(\alpha)$ $(\alpha \geq 0)$ and eq. 27, we know $s_2[k]$ in eq. 13, together with $s_1[k]$ in eq. 12, solves problem 10. $\diamondsuit$

D. Proof of Lemma 4.3

$\{b_1[k], b_2[k]\}_{k=0}^{N-1}$ is generated by DBAP and $s_i[k]$ is defined as

$$s_i[k] = \sum_{j=0}^{k-1} b_i[j], i=1,2 \tag{29}$$

We will prove Lemma 4.3 by showing that $s_2[k]$ defined in eq. 29 satisfies eq. 13. This result will be established by using mathematial induction on k.

When $k=1$, $s_2[k]=b_2[k-1]$ by the definition of $s_2[k]$. Eq. 13 (for $s_2[k]$) and 7 (for $b_2[k-1]$) are really the same. So Lemma 4.3 holds for $k=1$.

Suppose $s_2[k-1]$ satisfies eq. 13. We will prove $s_2[k]$ also satisfies eq. 13.

By eq. 13, the decision on $s_2[k]$ is categorized into three cases based on $\gamma[0]=\frac{L_1[0]}{L_2[0]}$.

1) $\gamma[0] \geq \frac{\lambda_2^k}{\lambda_1^k}2^{kQ+1}$*:* Under this situation, we get

$$\begin{aligned} \frac{\lambda_1^{k-1}}{2^{(k-1)Q+1}}L_1[0] \quad &\geq \quad \lambda_2^{k-1}L_2[0]\frac{2^Q\lambda_2}{\lambda_1} \\ &> \quad \lambda_2^{k-1}L_2[0] \end{aligned}$$

where the last inequality comes from $2^Q > \lambda_1\lambda_2$ and $\lambda_2 > 1$. By assumption, $s[k-1]$ satisfies eq. 13. So

$$s_2[k-1]=0 \tag{30}$$

Then we obtain

$$L_1[k-1] \quad = \quad \frac{\lambda_1^{k-1}}{2^{(k-1)Q}}L_1[0] \tag{31}$$

$$L_2[k-1] \quad = \quad \lambda_2^{k-1}L_2[0] \tag{32}$$

We can verify that $\frac{\lambda_1}{2^{Q+1}}L_1[k-1] \geq \lambda_2 L_2[k-1]$. Therefore DBAP yields $b_2[k-1]=0$ and

$$s_2[k]=s_2[k-1]+b_2[k-1]=0 \tag{33}$$

The above result on $s_2[k]$ satisfies eq. 13.

2) $\gamma[0] \leq \frac{\lambda_2^k}{\lambda_1^k}2^{-kQ-1}$*:* We can similarly prove $s[k]$ satisfies eq. 13 as we did for the case $\gamma[0] \geq \frac{\lambda_2^k}{\lambda_1^k}2^{kQ+1}$.

3) $\frac{\lambda_2^k}{\lambda_1^k}2^{-kQ-1} < \gamma[0] < \frac{\lambda_2^k}{\lambda_1^k}2^{kQ+1}$*:* First we prove it is **impossible** that

$$\frac{\lambda_1}{2^{Q+1}}L_1[k-1] \geq \lambda_2 L_2[k-1] \tag{34}$$

Suppose eq. 34 holds. Substituting the expressions of $L_1[k-1]$ $(L_1[k-1]=\frac{\lambda_1^{k-1}}{2^{s_1[k-1]}}L_1[0])$ and $L_2[k-1]$ $(L_2[k-1]=\frac{\lambda_2^{k-1}}{2^{s_2[k-1]}}L_2[0])$ into eq. 34 yields

$$\gamma[0]=\frac{L_1[0]}{L_2[0]} \geq \frac{\lambda_2^k}{\lambda_1^k}2^{Q+1+s_1[k-1]-s_2[k-1]} \tag{35}$$

Combining the requirement $\gamma[0] < \frac{\lambda_2^k}{\lambda_1^k}2^{kQ+1}$ with the above bound produces

$$Q+1+s_1[k-1]-s_2[k-1] < kQ+1 \tag{36}$$

Considering $s_1[k-1]+s_2[k-1]=(k-1)Q$, we get

$$s_2[k-1]>0 \tag{37}$$

i.e. side L_2 gets at least one bit among the total of $(k-1)Q$ ones. Suppose side L_2 gets the first bit at $k=k_1$ ($k_1 \leq k-1$). By algorithm 3.1, the decision on $b_1[j]$ and $b_2[j]$ aims to balance $L_1[j+1]$ and $L_2[j+1]$, which guarantees that

$$\frac{L_1[j]}{L_2[j]} \leq 2, \forall j \geq k_1 \tag{38}$$

The above equation certainly holds for $j=k-1$, i.e.

$$\frac{L_1[k-1]}{L_2[k-1]} \leq 2 \tag{39}$$

Thus $\frac{\lambda_1 L_1[k-1]}{\lambda_2 L_2[k-1]} \leq 2\frac{\lambda_1}{\lambda_2} < 2^{Q+1}$, which contradicts eq. 34 ! So eq. 34 is **impossible**.

Second we can similarly prove it is also **impossible** that

$$\frac{\lambda_1}{2^{-Q-1}} L_1[k-1] \leq \lambda_2 L_2[k-1] \tag{40}$$

Based on the impossibility of eq. 34 and 40 and the decision rule in eq. 7, we get

$$b_2[k-1]=\left[\frac{1}{2}\left(Q-\log_2\left(\frac{\lambda_1 L_1[k-1]}{\lambda_2 L_2[k-1]}\right)\right)\right] \tag{41}$$

Substituting the expressions of $L_1[k-1]$ and $L_2[k-1]$ into the above equation yields

$$\begin{aligned} & b_2[k-1] \\ = & \left[\frac{1}{2}\left(Q+s_1[k-1]-s_2[k-1]-\log_2\left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right)\right)\right] \end{aligned}$$

By the identity $s_1[k-1]=(k-1)Q-s_2[k-1]$, the above result can be simplified into

$$\begin{aligned} & b_2[k-1] \\ & = \left[0.5\left(kQ-\log_2\left(\tfrac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right)\right)\right]-s_2[k-1] \end{aligned}$$

Considering the definition of $s_2[k]$ in eq. 29, we obtain

$$\begin{aligned} s_2[k] &= s_2[k-1]+b_2[k-1] \\ &= \left[\frac{1}{2}\left(kQ-\log_2\left(\frac{\lambda_1^k L_1[0]}{\lambda_2^k L_2[0]}\right)\right)\right] \end{aligned}$$

Therefore $s_2[k]$ satisfies eq. 13.

In summary, $s_2[0]$ satisfies eq. 13. If $s_2[k-1]$ satisfies eq. 13, then $s_2[k]$ also satisfies equation 13. So by mathematical induction method, we can guarantee that $s_2[k]$ satisfies eq. 13 for all k and the proof is complete. $\diamondsuit$

E. Proof of Theorem 4.4

Denote the optimal performance of problems 8, 9 and 10 as P^*, $P^{(k)*}$ and $P_s^{(k)*}$ respectively. By the relaxation relationship among them, P^*, $P^{(k)*}$ and $P_s^{(k)*}$ satisfy the following equations.

$$P^* \geq \sum_{k=1}^{N} P^{(k)*} \tag{42}$$

$$P^{(k)*} \geq P_s^{(k)*} \tag{43}$$

Implementing DBAP, we obtain a bit assignment sequence $\mathbf{b}=\{b_1[j],b_2[j]\}_{j=0}^{N-1}$. By Lemma 4.3, we know the generated $\mathbf{b}^{(k)}=\{b_1[j],b_2[j]\}_{j=0}^{k-1}$ solves the kth problem in eq. 10, i.e. the optimal performance $P_s^{(k)*}$ is achieved by $\mathbf{b}^{(k)}$. It is obvious that $\mathbf{b}^{(k)}$ satisfies the constant bit rate constraint in eq. 4. So $\mathbf{b}^{(k)}$ is also a feasible solution to the kth problem in eq. 9. The two optimization problems in eq. 9 and 10 have the same performance index. By the optimality assumption of $P^{(k)*}$, we know $P_s^{(k)*} \geq P^{(k)*}$. Therefore the equality in eq. 43 holds.

The DBAP algorithm guarantees that $\mathbf{b}$ satisfies $\mathbf{b}^{(k-1)} \subset \mathbf{b}^{(k)}$ for $k=2,\cdots,N$. By Lemma 4.1, we know the equality in eq. 42 holds. Therefore the optimal performance P^* is equal to $\sum_{k=1}^{N} P_s^{(k)*}$ which is exactly the performance under DBAP and the proof is complete. $\diamondsuit$

REFERENCES

[1] Q. Ling and M. Lemmon, "Stability of quantized control systems under dynamicc bit assignment," *IEEE Trans. on Automatic Control*, vol. 50(5), pp. 734–740, 2005.

[2] D. Delchamps, "Stabilizing a linear system with quantized state feedback," *IEEE Transactions on Automatic Control*, vol. 35(8), pp. 916–924, 1990.

[3] N. Elia and S. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46(9), pp. 1384 –1400, 2001.

[4] W. Wong and R. Brockett, "Systems with finite communication bandwidth constraints- part ii: stabilization with limited information feedback," *IEEE Transactions on Automatic Control*, vol. 44(5), pp. 1049–1053, 1999.

[5] J. Baillieul, "Feedback designs in information-based control," in *Stochastic Theory and Control, Lecture Notes in Control and Information Sciences, B. Pasik-Duncan (ed.),Springer-Veralg LNCIS 280*, pp. 35–37.

[6] F. Fagnani and S. Zampieri, "Stability analysis and synthesis for scalar linear systems with a quantized feedback," *IEEE Transactions on Automatic Control*, vol. 48(9), pp. 1569–1584, 2003.

[7] R. Brockett and D. Liberzon, "Quantized feedback stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 45(7), pp. 1279 –1289, 2000.

[8] D. Liberzon, "On stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 48(2), pp. 304–307, 2003.

[9] S. Tatikonda, "Control under communication constraints," Ph.D. dissertation, M.I.T., 2000.

[10] S. Tatikonda and S. Mitter, "Control under communication constraints," *IEEE Transactions on Automatic Control*, vol. 49(7), pp. 1056–1068, 2004.

[11] J. Hespanha, A. Ortega, and L. Vasudevan, "Towards the control of linear systems with minimum bit-rate," in *Proc. of the Int. Symp. on the Mathematical Theory of Networks and Systems*, 2002.

[12] G. Nair and R. Evans, "Exponential stabilisability of finite-dimensional linear systems with limited data rates," *Automatica*, vol. 39, pp. 585–593, 2003.

[13] S. Haykin and M. Moher, *Modern wireless communications*. Pearson Prentice Hall, Upper Saddle Reiver, NJ 07458, 2003.

[14] S. Sarma, M. Dahleh, and S. Salapaka, "On time-varying bit-allocation maintaining input-output stability: a convex parameterization," in *IEEE Conference on Decision and Control*, 2004, pp. 1430–1435.