The Annals of Applied Probability
1997, Vol. 7, No. 3, 747–771

DYNAMIC CONTROL OF BROWNIAN NETWORKS: STATE SPACE COLLAPSE AND EQUIVALENT WORKLOAD FORMULATIONS

BY J. MICHAEL HARRISON AND JAN A. VAN MIEGHEM

Stanford University and Northwestern University

Brownian networks are a class of linear stochastic control systems that arise as heavy traffic approximations in queueing theory. Such Brownian system models have been used to approximate problems of dynamic routing, dynamic sequencing and dynamic input control for queueing networks. A number of specific examples have been analyzed in recent years, and in each case the Brownian network has been successfully reduced to an "equivalent workload formulation" of lower dimension. In this article we explain that reduction in general terms, using an orthogonal decomposition that distinguishes between reversible and irreversible controls.

1. Introduction and summary. This article is concerned with a class of stochastic system models, originally defined in [2] and there called "Brownian networks," that arise as heavy traffic approximations in queueing theory. To be more specific, these Brownian system models arise as approximations of queueing networks where system managers can exercise various types of dynamic control. In recent years such Brownian approximations have been employed successfully to study problems of dynamic sequencing [1, 4, 5], dynamic routing [6, 8, 9, 10, 16] and dynamic input control [15, 17].

Brownian networks have proved to be much more tractable than the conventional queueing models they replace. In particular, a key feature of the Brownian networks studied thus far is what Reiman [12] called "state space collapse." That is, a stochastic control problem associated with a Brownian network typically can be reduced to another problem which has a state space of lower dimension but is "equivalent" in an appropriate sense. The derivation of this "equivalent workload formulation" has been carried out on an ad hoc basis in the earlier work previously cited, but here we seek to explain it in general terms.

The next few paragraphs are devoted to a mathematical description of a Brownian network and a statement of our main result. In later sections we discuss examples and special cases that serve to explicate the relationship between Brownian networks and conventional queueing models. Readers will find that our notation differs somewhat from that used in [2] and other antecendent work, simply because more letters are needed here.

Throughout the paper, let $(\Omega, \mathscr{F}, P)$ be a fixed probability space on which is defined a filtration $\{\mathscr{F}_t,\ t \geq 0\}$. All stochastic processes to be considered are defined on this probability space, have time domain $[0, \infty)$ and have RCLL

Received August 1996; revised March 1997.
AMS 1991 *subject classifications*. 60K25, 60J70, 90B15.
Key words and phrases. Brownian networks, queueing networks, state space collapse, dynamic scheduling.

sample paths. In the usual way, a process $\eta = \{\eta(t),\ t \geq 0\}$ is said to be *adapted* if $\eta(t)$ is measurable with respect to $\mathcal{F}_t$ for each fixed $t \geq 0$. Let $X = \{X(t),\ t \geq 0\}$ be an m-dimensional Brownian motion with respect to $\{\mathcal{F}_t,\ t \geq 0\}$. We denote by μ and Σ the drift vector and covariance matrix of X, respectively, and assume that $X(0) = 0$ almost surely.

The data of a Brownian network are an $m \times n$ input–output matrix R, a $p \times n$ capacity consumption matrix K and an m-dimensional initial inventory vector $z \geq 0$. (Here m, n and p are all positive integers.) An *admissible control* is an n-dimensional process $Y = \{Y(t), t \geq 0\}$ such that

(1) Y is adapted,

(2) $U(\cdot)$ is nondecreasing with $U(0) \geq 0$, and

(3) $Z(t) \geq 0$ for all $t \geq 0$, where

(4) $Z(t) = z + X(t) + R\,Y(t)$ for all $t \geq 0$ and

(5) $U(t) = K\,Y(t)$ for all $t \geq 0$.

In the applications described later, m represents the number of *stocks* or *inventories* maintained by a system manager, n represents the number of *activities* available to the system manager, and p represents the number of *servers* or *processing resources* whose capacities are consumed by those activities. Components of the control Y represent cumulative time allocations to the various activities, usually expressed as deviations (which can be either positive or negative) from some nominal allocation. Components of Z represent inventory levels, and components of U represent cumulative idleness, or cumulative unused capacity, for the various servers. A pair of processes (Z, U) is said to be *achievable* if there exists an admissible control Y such that (4) and (5) hold.

The preceding paragraph specifies what we call a Brownian network, or Brownian system model, and to associate with this model a clear-cut stochastic control problem one must further specify an objective function. Strictly speaking, the objective is irrelevant to the mathematical results developed in this paper. However, our theory is motivated by problems in which cumulative cost incurred up to time t has the form

$$C(t) = \int_0^t h(Z(s))\,ds + cU(t), \tag{6}$$

where $h\colon \mathbb{R}^m \to \mathbb{R}$ is a measurable inventory holding cost function and c is a p-vector of linear cost rates associated with idleness of the p different processing resources. The important point here is that cost depends on the chosen control Y only through its associated inventory process Z and cumulative idleness process U.

In queueing network applications it is often natural to express congestion costs in terms of the total delays (also called throughput times or sojourn

times) experienced by jobs entering the network. This would seem to be inconsistent with (6), where congestion costs are expressed as a function of the inventory or queue length process Z. However, under heavy traffic conditions that justify a Brownian network formulation, Reiman's "snapshot principle" [11, 12, 13] suggests that one can represent the total delay experienced by an arriving job as a linear combination of the queue lengths seen by that job upon arrival. This allows cumulative cost to be expressed in the form (6), as Van Mieghem [14] has shown in the case of single-server systems.

For some applications it is desirable to replace (3) by a more general restriction of the following form: $Z(t) \in S$ for all $t \geq 0$, where S is a specified convex polyhedral state space. In the Appendix it is shown that such a model can always be reduced to a Brownian network of the standard form considered here.

The statement of our main result involves two new matrices, M and G, that are defined in terms of K and R. To motivate these definitions, consider the Brownian network model (1)–(5) with arbitrary initial state z, and suppose that an immediate impulse control $Y(0) = y$ is applied at time $t = 0$. According to (4) and (5), the corresponding initial values of Z and U are $Z(0) = z + \delta$ and $U(0) = u$, where $\delta = Ry$ and $u = Ky$. From (2) and (3) one sees that the impulse control is admissible only if $z + \delta \geq 0$ and $u \geq 0$. Hereafter we shall refer to y as a *control increment* and to δ as a *displacement*. If $u = 0$ (that is, $Ky = 0$), then the system manager can immediately apply another control increment of $-y$, which causes a displacement of $-\delta$ and thus returns the system to state z. Thus a control increment y is said to be *reversible* if $Ky = 0$. Let us define the linear spaces

$$\mathscr{B} = \{y \in \mathbb{R}^n \colon Ky = 0\} \quad \text{and} \tag{7}$$

$$\mathscr{N} = R\mathscr{B} = \{\delta \in \mathbb{R}^m \colon \delta = Ry,\ Ky = 0\}, \tag{8}$$

calling $\mathscr{B}$ the space of reversible control increments and $\mathscr{N}$ the space of reversible displacements. Also, let $\mathscr{M}$ be the orthogonal complement of $\mathscr{N}$. We denote by q the dimension of $\mathscr{N}$ $(0 \leq q \leq m)$, so $\mathscr{M}$ is of dimension

$$d = m - q. \tag{9}$$

Finally, let N be any $q \times m$ matrix whose rows span $\mathscr{N}$ and M be any $d \times m$ matrix whose rows span $\mathscr{M}$. Such matrices can be computed mechanically from K and R, of course, and more will be said about that in Section 2.

Given an arbitrary state vector $z \in \mathbb{R}^m_+$, let $w = Mz$ and $\widehat{w} = Nz$; these vectors are effectively the projections of z onto the spaces $\mathscr{M}$ and $\mathscr{N}$, respectively. The essential insight embodied in Theorem 1 below is that any two state vectors whose difference is a reversible displacement are equivalent, because a system manager can instantaneously exchange either of those state vectors for the other without affecting the cumulative idleness process U, and hence without incurring any cost. Thus, to describe more compactly the state of the system at any given time, we can simply discard the reversible component

$\widehat{w}$ of z, retaining its orthogonal complement w as an adequate summary of system status for purposes of future decision making.

This observation leads to an alternative model formulation, described precisely below, in which system status at time t is represented by the d-dimensional process $W(t) = M\,Z(t)$. By definition, the rows of M are orthogonal to all reversible displacements, meaning that $MRy = 0$ if $Ky = 0$; one can choose the rows of M to be any maximal set of linearly independent m-vectors having that property. In Section 2 it is shown that

$$MR = GK \tag{10}$$

for at least one $d \times p$ matrix G (in general, G is not unique), and that conversely, one can choose M to be any $d \times m$ matrix of rank d such that (10) holds. There we also show how, given a matrix M whose rows span $\mathscr{M}$, one can mechanically compute a matrix G satisfying (10).

Recalling that the m-dimensional Brownian motion X and the initial inventory vector $z \in \mathbb{R}_+^m$ are taken as primitive in our original Brownian network model (1)–(5), let us now define

$$w = M\,z \quad \text{and} \tag{11}$$

$$\xi(t) = M\,X(t) \quad \text{for all } t \geq 0. \tag{12}$$

Obviously, w is a d-vector and $\xi = \{\xi(t)\colon t \geq 0\}$ is a d-dimensional Brownian motion with drift vector $M\mu$ and covariance matrix $M\Sigma M'$. One can now left-multiply both sides of (4) by M, then use (5) and (10)–(12) to obtain the basic system equation $W(t) = w + \xi(t) + G\,U(t)$ for all $t \geq 0$. This tells us that our reduced system descriptor W evolves in the absence of control as the d-dimensional Brownian motion ξ, and that it depends on the chosen control Y only through its associated cumulative idleness process U.

With that motivation and preliminary development, our main result can be stated. A pair of processes (Z, U) is said to be *achievable in the reduced Brownian network* if

$$Z \text{ and } U \text{ are adapted,} \tag{13}$$

$$U(\cdot) \text{ is nondecreasing with } U(0) \geq 0, \tag{14}$$

$$U(t) \in \text{column space of } K \text{ for all } t \geq 0, \tag{15}$$

$$Z(t) \geq 0 \text{ for all } t \geq 0, \text{ and} \tag{16}$$

$$W(t) = M\,Z(t) \text{ for all } t \geq 0, \text{ where} \tag{17}$$

$$W(t) = w + \xi(t) + G\,U(t) \text{ for all } t \geq 0. \tag{18}$$

For comparison, recall that the pair (Z, U) is said to be achievable in our original Brownian network if there exists a process Y satisfying (1)–(5).

THEOREM 1. *A pair* (Z, U) *is achievable in the original Brownian network if and only if it is achievable in the reduced Brownian network.*

REMARK 1. In all of the applications described in later sections, and in all other interesting applications of which we are aware, the capacity consumption matrix K has rank p (i.e., its rows are linearly independent). In that case, the column space of K is all of $\mathbb{R}^p$, and condition (15) can simply be deleted when describing the reduced Brownian network.

REMARK 2. The equivalence expressed by Theorem 1 is valid for any choice of the basis matrix M (that is, for any $d \times m$ matrix M whose rows span $\mathcal{M}$) and for any G that is consistent with that choice in the sense that (10) holds.

Theorem 1 will be proved in the next section, where some additional notation is introduced for that purpose. Sections 4–6 show how the theorem applies to various special cases of interest in queueing theory, and in particular, how it generalizes various results that have been proved in the literature. In the course of that discussion we explain why in certain contexts it is natural to describe the process W in (17) as a *workload process*, and to describe (13)–(18) as an "equivalent workload formulation" of the original Brownian network. In Section 7 two important open problems are described briefly.

Theorem 1 does not explicitly address problems of optimal system control. Rather, it characterizes the set of achievable trajectories (Z, U) from which a system manager may choose. In Section 3 we prove another theorem which shows in a general setting the implications of that equivalence for optimal control, making no reference to the specially structured matrices K and R that arise in queueing network models. (Readers will see that many variants of this theorem are possible.) The next few paragraphs are devoted to an informal statement of the result to be proved in Section 3, with enough connective logic to make the result at least plausible. To ease the exposition, several technical assumptions are suppressed for the time being.

Let us assume a cost structure of the form (6), with the vector c of idleness cost rates and the holding cost function $h(\cdot)$ both nonnegative. For the sake of concreteness, let us further suppose that the objective is to minimize expected discounted costs over an infinite planning horizon. (One could equally well consider a finite planning horizon, either with or without discounting.) Thus, denoting by $\gamma > 0$ the interest rate for discounting, the system manager seeks to

$$\text{minimize } E\left\{\int_0^\infty e^{-\gamma t}\big[h(Z(t))\,dt + c\,dU(t)\big]\right\}. \tag{19}$$

One should think of m and n as large integers (on the order of hundreds, say) while p is relatively small (less than ten, say). The applications described later should make it clear why such a cost structure and such parameter ranges are natural. Recall that the control Y and inventory process Z in our original Brownian network are of dimension n and m, respectively, while the processes

U and W appearing in the reduced Brownian network are of dimension p and d, respectively, where $d \leq p$ by definition.

Using Theorem 1, the high-dimensional state descriptor $Z(t)$ can be eliminated from our reduced Brownian network, at least in a formal sense, as follows. First, conditions (16) and (17) are completely equivalent to the requirement that

$$W(t) \in S \quad \text{for all } t \geq 0, \tag{20}$$

where S is a convex cone in $\mathbb{R}^d$ defined by

$$S = \{w\colon w = Mz,\ z \geq 0\}. \tag{21}$$

To maintain feasibility, the system manager need only observe the low-dimensional process W, driven by a low-dimensional Brownian motion ξ, and continuously choose increments of the low-dimensional, nondecreasing control U so as to assure that the process W remains within the cone S. Given a feasible choice of U, one can immediately identify the best associated choice of the process Z: at each time t, given the value of $W(t) \in S$, the system manager should choose

$$Z(t) = \arg\min\{h(z)\colon Mz = W(t),\ z \geq 0\}. \tag{22}$$

Thus, defining $f\colon S \to \mathbb{R}$ via

$$f(w) = \min\{h(z)\colon Mz = w,\ z \geq 0\},\ w \in S, \tag{23}$$

it follows from Theorem 1 that our original control problem reduces to this: choose a p-dimensional control U to

$$\text{minimize } E\left\{\int_0^\infty e^{-\gamma t}\left[f(W(t))\,dt + c\,dU(t)\right]\right\}, \tag{24}$$

subject to

$$U \text{ is adapted,} \tag{25}$$

$$U(\cdot) \text{ is nondecreasing with } U(0) \geq 0, \tag{26}$$

$$U(t) \in \text{column space of } K \text{ for all } t \geq 0, \text{ and} \tag{27}$$

$$W(t) \in S \text{ for all } t \geq 0, \text{ where} \tag{28}$$

$$W(t) = w + \xi(t) + GU(t) \text{ for all } t \geq 0. \tag{29}$$

Paraphrasing, one might say that the reduced Brownian network model gives rise to a *hierarchical* view of system control: first the system manager must choose an irreversible control component U so as to keep W within the "feasible region" S defined by (21), and then a reversible control component can be chosen so as to realize any process Z which is consistent with W in the sense that $W(t) = MZ(t)$. To repeat, section 3 is devoted to a proof that our

original control problem, specified by (1)–(5) and (19), is equivalent to the reduced problem (24)–(29) under additional regularity assumptions on the cost function h.

Readers will find that no property of Brownian motion is ever invoked in Section 2 or 3. The proof of Theorem 1 involves basic linear algebra applied on a path-by-path basis, and the control problem equivalence proved in Section 3 derives similarly from sample path relationships. Thus all of our results actually hold when X is an arbitrary stochastic process, but in our view the system models and control problems studied here are most interesting in the Brownian case. Also, as Sections 4–6 suggest, it is applications to Brownian models of queueing networks that motivate the general theory developed in this paper, and we have chosen a title to emphasize that connection.

Primes are used throughout the paper to denote transposes, and vector inequalities should be interpreted componentwise as usual. Also, the letters e and I will be used to denote a column vector of ones and an identity matrix, respectively, of any dimension. The appropriate dimension will usually be clear from context, but parenthetical remarks about the dimension of e or I are inserted at several points to eliminate potential confusion.

2. Proof of Theorem 1. Before stating the proof, we need some additional notation and preliminary propositions. Let r be the rank of the capacity consumption matrix K (thus $r \leq p$), and let $\mathscr{A}$ be the row space of K. In Section 1 we defined $\mathscr{B}$ as the null space of K, so $\mathscr{B}$ is of dimension $n-r$, and the linear spaces $\mathscr{A}$ and $\mathscr{B}$ are orthogonal. Hereafter, let A be an $r \times n$ orthonormal matrix whose rows span $\mathscr{A}$, and let B be an $(n-r) \times n$ orthonormal matrix whose rows span $\mathscr{B}$. Thus (A', B') is an orthonormal basis matrix for $\mathbb{R}^m$ and we have $AA' = I$, $BB' = I$, $A'A + B'B = I$ and $AB' = 0$. Our definition (8) of the linear space $\mathscr{N}$ can be equivalently stated as follows: $\mathscr{N}$ is the column space of RB'. In Section 1 we denoted by q the dimension of $\mathscr{N}$ (or equivalently, the rank of RB'), and by N a $q \times n$ matrix whose rows span $\mathscr{N}$. To be concrete, one can take the rows of N to be any q linearly independent rows of BR'. The basis matrices A, B and N are considered to be fixed throughout the following discussion.

Because A and K have the same row space $\mathscr{A}$, we know that $A = LK$ for some $r \times p$ matrix L (not necessarily unique). For future purposes it will be convenient to fix a choice of L and then define the $d \times m$ matrix

$$H = L'A. \tag{30}$$

The usefulness of this matrix H derives from the following observation: we have $y = A'Ay + B'By$ for any m-vector y; now making the substitution $A = LK$, that identity can be rewritten as

$$y = H'u + B'v \quad \text{where } u = Ky \text{ and } v = By. \tag{31}$$

The component $H'u$ in (31) is the projection of y onto $\mathscr{A}$, while $B'v$ is the projection of y onto $\mathscr{B}$. Because $KB' = 0$ as a matter of definition, the following proposition is immediate from (31).

PROPOSITION 1. *$KH'u = u$ for all $u \in$ column space of K.*

Let us consider now the matrices M and G that appear in the reduced Brownian network model (13)–(18). By definition, M can be chosen as any $d \times m$ matrix satisfying $MRB' = 0$; given numerical values for B and R, one can use that relationship and Gaussian elimination to mechanically compute M. The following proposition provides an alternative and very useful characterization.

PROPOSITION 2. *Let M be a $d \times m$ matrix of rank d. The rows of M span $\mathscr{M}$ if and only if*

$$MR = GK \tag{32}$$

for some $d \times p$ matrix G (not necessarily unique).

REMARK. Given a $d \times m$ matrix M whose rows span $\mathscr{M}$, the following proof shows that one G satisfying (32) is

$$G = MRH'. \tag{33}$$

PROOF. Suppose first that M satisfies (32) for some G. Then $MRB' = 0$, because $KB' = 0$ by the definition of B. Thus each row of M is orthogonal to $\mathscr{N}$, because $\mathscr{N}$ is the column space of RB'. Moreover, the d rows of M are linearly independent, so they span the d-dimensional orthogonal space $\mathscr{M}$. Conversely, suppose that the rows of M span $\mathscr{M}$, implying that $MRB' = 0$. Now (31) can be equivalently expressed as $I = H'K + B'B$, and multiplying both sides of that identity by MR gives $MR = MRH'K$, so (32) holds with $G = MRH'$. □

Turning now to the proof of Theorem 1, let M and G be an arbitrary but fixed pair of matrices, where M is full rank $d \times m$, G is $d \times p$ and both jointly satisfy (32). Thus the rows of M span $\mathscr{M}$ by Proposition 2. Let us first suppose that (Z, U) is an achievable pair in the original Brownian network. Thus there exists a control Y such that Y, X, U, Z and z jointly satisfy (1)–(5). From (1), (4) and (5) if follows that Z and U are both adapted to X, which is condition (13) of the reduced Brownian model. Moreover, condition (14) is identical to (2), condition (15) is immediate from (5) and condition (16) is identical to (3). We define $W(t) = MZ(t)$ for all $t \geq 0$, which is condition (17) of the reduced Brownian network. Finally, defining w and ξ by means of (11) and (12), we premultiply both sides of (4) by M and use the fundamental identity (32) to obtain (18). Thus Z, U, W, ξ and w jointly satisfy (13)–(18), so the pair (Z, U) is achievable in our reduced Brownian network model.

To prove the converse, let (Z, U) be a pair of processes that is achievable in the reduced Brownian network. Thus Z, U, W, ξ and w jointly satisfy (13)–(18). By the definition of N, the $q \times (n - r)$ matrix NRB' is of rank q (recall that $q \leq n - r$), so there exists at least one $(n - r) \times q$ matrix Q satisfying

$$(NRB')Q = I \quad \text{(the } q \times q \text{ identity matrix).} \tag{34}$$

Let us now define a control Y via

$$Y(t) = H'U(t) + B'V(t) \quad \text{for all } t \geq 0, \tag{35}$$

where

$$V(t) = QN[Z(t) - z - X(t) - RH'U(t)] \quad \text{for all } t \geq 0. \tag{36}$$

Our goal is to show that this control Y is admissible in the original Brownian network and that it yields the pair (Z, U). That is, we seek to show that Y, X, U, Z and z jointly satisfy (1)–(5). Condition (1) is immediate from (13), (35) and (36), while conditions (2) and (3) are identical to (14) and (16), respectively. Moreover, premultiplying both sides of (35) by K gives $KY(t) = KH'U(t)$, because $KB' = 0$ by the definition of B. Combining this with (15) and Proposition 1 gives $KY(t) = U(t)$ for all $t \geq 0$, which is (5), and thus it remains only to show that the control Y defined by (35) and (36) satisfies (4).

Recall that $MRB' = 0$ by the definition of M. Thus, premultiplying both sides of (35) by MR and then using (32), we have $MRY(t) = GKH'U(t)$. It has already been noted that $KH'U(t) = U(t)$ by (15) and Proposition 1, so we have $MRY(t) = GU(t)$. Combining that with (11), (12), (17) and (18) yields

$$MZ(t) = Mz + MX(t) + MRY(t) \quad \text{for all } t \geq 0. \tag{37}$$

Next, premultiplying both sides of (36) by NRB', substituting (34) on the right-hand side, then rearranging terms and substituting (35), we have

$$NZ(t) = Nz + NX(t) + NRY(t) \quad \text{for all } t \geq 0. \tag{38}$$

By definition, the $m \times m$ matrix formed by combining rows of M and N is nonsingular, so (37) and (38) together imply $Z(t) = z + X(t) + RY(t)$, which is (4). Thus the pair (Z, U) is achievable in the original Brownian network, which completes the proof of Theorem 1.

3. Reducing the dimension of optimal control problems. Let us consider now the two problems of optimal system control discussed at the end of Section 1: the *original problem* is to choose a control Y satisfying (1)–(5) so as to minimize the discounted expected cost functional appearing in (19); and the *reduced problem* is to choose a control U satisfying (25)–(29) so as to minimize the expected discounted cost in (24). The original problem involves an m-dimensional state descriptor Z, while the reduced problem involves a d-dimensional state descriptor W.

In addition to the measurability of h and the nonnegativity of c and h assumed in Section 1, which assure that expected discounted costs are always well defined (but possibly infinite), we impose the following regularity assumptions on h. First, for each w in the "feasible region" S defined by (21), there exists a z^* in the set $\{Mz = w,\ z \geq 0\}$ which minimizes h over that set. Moreover, one can choose the minimizer so that $z^* = g(w)$, where $g\colon S \to \mathbb{R}^m$ is continuous. Given a process $W = \{W(t),\ t \geq 0\}$ taking values in S (recall that the word "process" automatically implies RCLL paths), we write $g(W)$ to

mean the process $Z = \{Z(t),\ t \geq 0\}$ with $Z(t) = g(W(t))$ for all $t \geq 0$; the continuity of g assures that $g(W)$ also has RCLL paths. For each $w \in S$ let

$$f(w) = h(g(w)) = \min\{h(z)\colon Mz = w,\ z \geq 0\},$$

as in (23). For each control Y satisfying (1)–(5) we define the objective value

$$\Phi(Y) = E\left\{\int_0^\infty e^{-\gamma t}[h(Z(t))\,dt + c\,dU(t)]\right\},$$

and similarly, for each control U satisfying (25)–(29) let

$$\Psi(U) = E\left\{\int_0^\infty e^{-\gamma t}[f(W(t))\,dt + c\,dU(t)]\right\}.$$

Throughout this section, an "admissible control in the original control problem" is a process Y satisfying (1)–(5). Similarly an "admissible control in the reduced control problem" is a process U satisfying (25)–(29). Let $\Phi^* = \inf \Phi(Y)$ and $\Psi^* = \inf \Psi(U)$, where the infima are taken over admissible controls in the original and reduced control problems, respectively. An admissible control Y for the original problem is said to be "optimal" if $\Phi(Y) = \Phi^*$, and similarly for the reduced problem.

PROPOSITION 3. *Suppose that U is an admissible control in the reduced problem, with corresponding workload process $W = w + \xi + GU$, and define $Z^* = g(W)$. Then there exits an admissible control Y^* in the original problem that achieves the pair (Z^*, U), and $\Phi(Y^*) = \Psi(U)$.*

PROOF. By hypothesis, U satisfies the constraints (25)–(29) of the reduced control problem, and it follows that U and Z^* jointly satisfy (13)–(18). That is, the pair (Z^*, U) is achievable in our reduced Brownian network model, so Theorem 1 shows that there exists an admissible control Y^* in the original Brownian network model which achieves (Z^*, U). Finally, from the definitions of f and g, one has that

$$\Psi(U) = E\left\{\int_0^\infty e^{-\gamma t}[h(Z^*(t))\,dt + c\,dU(t)]\right\} = \Phi(Y^*). \qquad \square$$

PROPOSITION 4. *Suppose that (U, Z) is a pair of processes achieved by an admissible control Y in the original control problem. Then U is an admissible control in the reduced problem, and $\Psi(U) \leq \Phi(Y)$.*

PROOF. By hypothesis, the pair (U, Z) is achievable in our original Brownian network model, so Theorem 1 says that (U, Z) is also achievable in the reduced network, which means that U and Z jointly satisfy (13)–(18). From (16) and (17) it follows that the process $W = w + \xi + GU$ satisfies $W(t) \in S$ for all $t \geq 0$, and hence that U satisfies (25)–(29). That is, U is an admissible control in our reduced Brownian control problem. Defining Z^* and Y^* exactly as in Proposition 3, we have that $\Psi(U) = \Phi(Y^*)$. From the definition of g it

is immediate that $h(Z^*(t)) \leq h(Z(t))$ for all $t \geq 0$ and hence $\Phi(Y^*) \leq \Phi(Y)$. Thus $\Psi(U) = \Phi(Y^*) \leq \Phi(Y)$. □

THEOREM 2. *The original and reduced control problems are equivalent in the following sense.*

(i) *Suppose that Y is an optimal control in the original problem, with associated cumulative idleness process $U = KY$. Then U is an optimal control in the reduced problem, and $\Phi(Y) = \Psi(U)$.*

(ii) *Suppose that U is an optimal control in the reduced problem, with associated workload process $W = w+\xi+GU$, and define $Z^* = g(W)$. There exists an admissible control Y^* in the original problem which achieves the pair (Z^*, U), that control Y^* is optimal in the original problem, and $\Psi(U) = \Phi(Y^*)$.*

PROOF. (i) Because Y is admissible in the original problem, Proposition 4 says that U is admissible in the reduced problem and $\Psi(U) \leq \Phi(Y)$. Suppose U is not optimal in the reduced problem, meaning that there exists another admissible control U^* with $\Psi(U^*) < \Psi(U)$. Then by Proposition 3 there exists an admissible control Y^* in the original problem with $\Phi(Y^*) = \Psi(U^*) < \Psi(U) = \Phi(Y)$, which contradicts the assumed optimality of Y.

(ii) Proposition 3 assures the existence of an admissible control Y^* in the original problem which achieves (Z^*, U) and satisfies $\Phi(Y^*) = \Psi(U)$. Suppose Y^* is not optimal in the original problem, meaning that there exists another admissible control Y' with $\Phi(Y') < \Phi(Y^*)$. Then by Proposition 4 there is an admissible control U' in the reduced problem with $\Psi(U') \leq \Phi(Y') < \Phi(Y^*) = \Psi(U)$, but this contradicts the assumed optimality of U. □

4. Dynamic sequencing in open queueing networks. Brownian networks were introduced in [2] as heavy traffic approximations of complex queueing networks in which system managers exercise dynamic control in various forms. The theory was originally motivated by problems of dynamic sequencing (or dynamic scheduling) in open multiclass networks. Let us briefly summarize these sequencing problems in the context of a concrete example, adopting the framework proposed in [2] but using somewhat different notation and terminology. Consider the system pictured in Figure 1, which was introduced and discussed at some length in [3]. We have external arrivals of three different types (called A, B and C) and there are three servers (represented by circles). As shown in the figure, type A jobs may follow either of two different routes, and for purposes of this section let us assume that the routes of type A jobs are determined by a sequence of independent coin flips as they arrive. We define a different "job class" for each combination of job type and stage of route completion, including one class for type A jobs seeking a first service at station 1 when following the upper route and another class for A jobs following the lower route. There are eight such classes in total, and their numbering is indicated in the open-ended rectangles in Figure 1. (These rectangles represent infinite-capacity buffers in which the various classes reside.) For concreteness, let us assume that the mean service times for the different classes

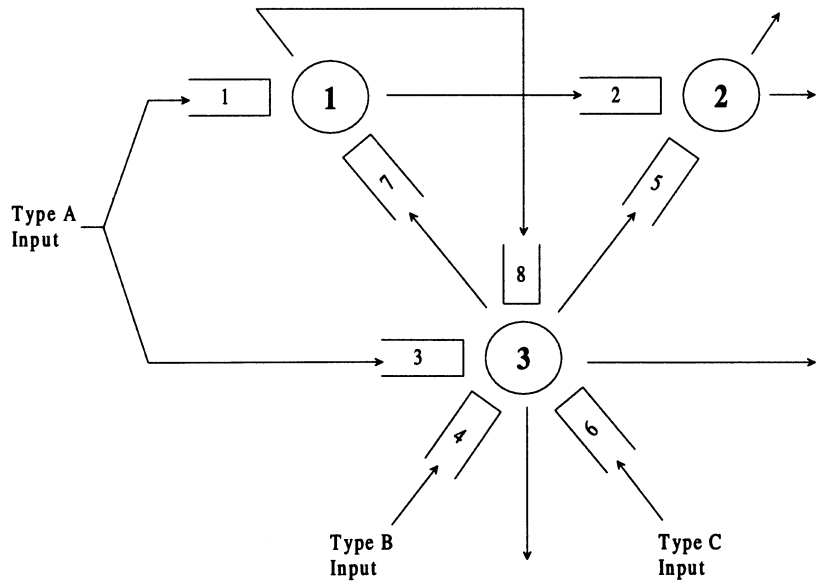

FIG. 1. *An open three-station network.*

are given by

$$(\tau_1, \ldots, \tau_8) = (2,\ 3,\ 1,\ 1,\ 1,\ 1,\ 2,\ 1), \tag{39}$$

and that jobs of types A, B and C arrive according to independent Poisson processes at average rates of $1/2, 1/4$ and $1/4$, respectively. With the random routing of type A jobs described above, we then have the following vector α of external arrival rates into the various classes:

$$(\alpha_1, \ldots, \alpha_8) = (1/4,\ 0,\ 1/4,\ 1/4,\ 0,\ 1/4,\ 0,\ 0). \tag{40}$$

Each job class $i = 1, \ldots, 8$ is handled by a unique server $\sigma(i)$ and we have in this example that

$$\begin{gathered} \sigma(1) = \sigma(7) = 1, \qquad \sigma(2) = \sigma(5) = 2 \quad \text{and} \\ \sigma(3) = \sigma(4) = \sigma(6) = \sigma(8) = 3. \end{gathered} \tag{41}$$

Finally, routing in a multiclass network is summarized by a Markov class-transition matrix P (also called the *switching matrix* or *routing matrix*) as follows: when a job of class i finishes service at station $\sigma(i)$, it immediately turns into a job of class j with probability P_{ij}, independent of all previous history. For our example P is 8×8 with $P_{12} = P_{45} = P_{67} = P_{78} = 1$ and $P_{ij} = 0$ otherwise. For a general open network P is transient and the probability that a class i job leaves the network after completing service is $1 - \sum_j P_{ij}$. Thus we can define (recall that primes denote transposes)

$$Q = (I - P')^{-1} = (I + P + P^2 + \cdots)' \quad \text{and} \quad \lambda = Q\alpha, \tag{42}$$

so that λ is the vector of total arrival rates into the various job classes, including both external arrivals and internal transitions.

Assuming that external arrivals and job routes are uncontrollable, the only decision-making capability that the system manager has involves the sequence in which jobs of various classes will be served at each station. That is, each time server k completes the service of a job, the system manager can choose a job from any of the classes $i \in \mathscr{C}(k)$ whose queue is nonempty at that moment. In general, systems of the type under discussion have p different processing resources, m different stocks of material, and m different processing activities, where resource k refers to server k, stock i refers to jobs of class i, and activity j corresponds to the processing of class i jobs by server $\sigma(i)$. Later we shall consider systems in which the number of activities n is strictly larger than the number of job classes (stocks) m. Proceeding as in [2], we define an $m \times m$ diagonal matrix D, a $p \times m$ *resource consumption matrix* K and an $m \times m$ *input-output matrix* R as follows:

$$D = \operatorname{diag}(\tau_1, \ldots, \tau_m), \tag{43}$$

$$K_{ij} = 1 \quad \text{if } \sigma(j) = i \text{ and } K_{ij} = 0 \text{ otherwise, and} \tag{44}$$

$$R = (I - P')D^{-1}. \tag{45}$$

Thus $K_{ij} = 1$ if server i is responsible for activity j (the processing of class j jobs) and $K_{ij} = 0$ otherwise. R_{ij} represents the average rate at which activity j depletes stock i (the queue of class i jobs) when server $\sigma(j)$ is devoted exclusively to that activity. For the example pictured in Figure 1, we have

$$K = \begin{bmatrix} 1 & & & & & 1 & \\ & 1 & & & 1 & & \\ & & 1 & 1 & & 1 & & 1 \end{bmatrix} \tag{46}$$

and

$$R = \begin{bmatrix} 1/2 & & & & & & & \\ -1/2 & 1/3 & & & & & & \\ & & 1 & & & & & \\ & & & & 1 & & & \\ & & & & -1 & 1 & & \\ & & & & & & 1 & \\ & & & & & & -1 & 1/2 \\ & & & & & & & -1/2 & 1 \end{bmatrix}. \tag{47}$$

Sections 2–5 of [2] explain how a dynamic sequencing problem of the general type described above can be approximated by a dynamic control problem for a properly chosen Brownian network. The approximating Brownian network is described by (1)–(5), with the capacity consumption matrix K and input–output matrix R derived from data of the original queueing system via

(44) and (45). The drift vector μ and covariance matrix Σ for the approximating Brownian network are derived from data of the queueing model through straightforward calculations that need not be repeated here. The controlled stochastic process Z appearing in (1)–(5) approximates a scaled queue length process, or scaled inventory process, with one component for each job class, and the nondecreasing process U approximates a scaled cumulative idleness process, with one component for each server. Finally, the m-dimensional control Y in (1)–(5) approximates a vector of scaled deviation controls, each component of which expresses the cumulative time allocation to jobs of a particular class as a deviation from the long-run average allocation required by that class.

Let us consider now the application of Theorem 1 to a Brownian network whose data K and R are derived from queueing network data via (44) and (45). From (44) it follows that the rank of K equals its row dimension p (the number of servers). Also, the rows of K are orthogonal, because the constituencies of the various servers are disjoint, so the orthonormal matrix A in Section 2 can be formed by simply rescaling the rows of K. Recall that $\mathscr{B}$ consists of all $y \in \mathbb{R}^m$ such that $Ky = 0$. By definition, $\mathscr{N}$ is the space of vectors $z \in \mathbb{R}^m$ such that $z = Ry$ for some $y \in \mathscr{B}$. Now R is invertible in the current context by (42) and (45), with $R^{-1} = DQ$, so $\mathscr{N} = \{z \in \mathbb{R}^m\colon R^{-1}z \in \mathscr{B}\}$, or equivalently,

$$\mathscr{N} = \{z \in \mathbb{R}^m\colon KR^{-1}z = 0\}. \tag{48}$$

Next, $\mathscr{M}$ is defined as the orthogonal complement of $\mathscr{N}$ and M can be chosen as any matrix whose row span is $\mathscr{M}$, so by (48) we can take

$$M = KR^{-1} = KDQ = KD(I - P')^{-1}. \tag{49}$$

Reviewing the definitions of q and d given in Section 1, readers will see that $q = m - p$ and $d = p$ in the current context. Finally, from (49) we have $MR = GK$ where

$$G = I \quad (\text{the } p \times p \text{ identity matrix}). \tag{50}$$

Equations (49) and (50) provide the essential data for the reduced Brownian network model (13)–(18) that was introduced in Section 1. As observed in Section 6 of [2], one interprets the (k, j)th element of M as the expected total time that server k must devote to a class j job before it ultimately exits the network. Thus, recalling that $Z(t)$ represents a vector of (scaled) queue lengths for the various classes at time t, one may interpret the p-dimensional process $W(t) = MZ(t)$ in (17) as follows: $W_k(t)$ represents the (scaled) expected total time required from server k to complete the processing of all jobs present anywhere in the network at time t. In brief, $W_k(t)$ is the total workload for server k embodied in jobs present at time t, and we call W the *workload process* for our Brownian network model. Theorem 1 says that the original Brownian network model (1)–(5) is equivalent to the reduced Brownian network (13)–(18), although the latter formulation appears to be much simpler.

For a general interpretation of the reduced Brownian network, let us assume for concreteness an objective of the form (19), so the system manager's

dynamic control problem is ultimately expressed in reduced form by (24)–(29). As mentioned at the end of Section 1, the reduced formulation involves a *hierarchical* control structure. At the higher level, the manager chooses a (scaled) cumulative idleness process U, thus reflecting a policy as to when the various servers will and will not be working. This resource utilization plan must be one that keeps the (scaled) workload process W within the feasible region S defined by (21), which means that $W(t)$ must at all times be consistent with *some* choice of nonnegative (scaled) queue length vector $Z(t)$. In the context under discussion here, the "reversible control increments" described in Section 1 correspond to redistributions of server effort, relative to a set of nominal allocations, that do not create idleness. Given a choice of U, the system manager can allocate or distribute the busy time of each server so as to realize *any* vector queue length process Z which is consistent with the workload process W achieved by U. This means that, in the idealized limiting case represented by the Brownian network model, any queue length vector can be swapped instantaneously for any other queue length vector that has the same expected total work content for each server. Further discussion of this "equivalent workload formulation" for open networks with dynamic sequencing can be found in Section 6 of [2].

5. Effect of dynamic routing in open queueing networks. In our previous discussion of the three-station example pictured in Figure 1, matters were simplified by the assumption that type A jobs were randomly routed via independent coin flips. In this section we consider the more interesting problem in which type A jobs can be dynamically routed to either the upper route pictured in Figure 1 or the lower one, depending on system status. Moreover, any of the three input processes can be turned off (or equivalently, jobs of any type can be rejected upon arrival) at any time. For simplicity, we assume that the routing of each type A job must be decided at the moment of its arrival. The Brownian approximation for this network control problem was derived in Sections 2 and 3 of [3], and we shall recapitulate only the essential aspects of that derivation in the following paragraph.

With dynamic routing and dynamic input control eliminated from consideration, we previously identified eight "processing activities" in the three-station example, corresponding to service of the eight job classes defined in Figure 1, each conducted by one of three servers. To extend that formulation it is conceptually easiest to speak in terms of three fictional "input servers" numbered 4, 5 and 6, who generate arrivals of types A, B, and C, respectively. We associate with the input servers four new "activities," as follows:

$$\begin{aligned}
\text{activity } 9 &= \text{creation of class 1 jobs} \quad \text{(by server 4)},\\
\text{activity } 10 &= \text{creation of class 3 jobs} \quad \text{(by server 4)},\\
\text{activity } 11 &= \text{creation of class 4 jobs} \quad \text{(by server 5)},\\
\text{activity } 12 &= \text{creation of class 6 jobs} \quad \text{(by server 6)}.
\end{aligned}$$

Hereafter when we say that "server 4 is engaged in activity 9" this is understood to mean that the type A input process is turned on and any resulting arrival will be routed to buffer 1, so the instantaneous Poisson arrival rates to buffers 1 and 3 are $\lambda_1 = 1/2$ and $\lambda_3 = 0$, and similarly for activities 10, 11 and 12. For concreteness, we allow the input servers to work at less than full capacity and server 4 to divide its time between activities 9 and 10 if doing so is deemed desirable. Such actions correspond to randomized acceptance of new arrivals and randomized routing to type A arrivals, respectively. Similarly, servers 1, 2 and 3 are allowed to divide their time among activities available to them, processing several job classes simultaneously, provided that the total rate of effort allocation does not exceed 100 percent.

In the problem conceptualization outlined above, one takes the view that all job flows are the consequence of some "activity" undertaken by one of the system's six "servers." In the obvious way, the 8×8 input–output matrix R defined by (47) is extended to an 8×12 matrix with one column for each activity as follows (nonzero elements in the last four columns reflect the average arrival rates assumed for jobs of the three types):

$$R = \begin{bmatrix} 1/2 & & & & & & -1/2 & & & & & \\ -1/2 & 1/3 & & & & & & & & & & \\ & & 1 & & & & & -1/2 & & & & \\ & & & 1 & & & & & -1/4 & & & \\ & & & -1 & 1 & & & & & & & \\ & & & & & 1 & & & & & -1/4 & \\ & & & & & -1 & 1/2 & & & & & \\ & & & & & & -1/2 & 1 & & & & \end{bmatrix}. \tag{51}$$

Similarly, the 3×8 capacity consumption matrix K defined by (46) is extended to the following 6×12 matrix, whose last three rows correspond to the new input servers and last four columns correspond to the new input activities:

$$K = \begin{bmatrix} 1 & & & & & 1 & & & & & & \\ & 1 & & & 1 & & & & & & & \\ & & 1 & 1 & & 1 & & 1 & & & & \\ & & & & & & & & 1 & 1 & & \\ & & & & & & & & & & 1 & \\ & & & & & & & & & & & 1 \end{bmatrix}. \tag{52}$$

Let us consider now the application of Theorem 1 to a Brownian network whose data R and K are given by (51) and (52). Beginning with this numerical data, one can mechanically compute the matrices M and G for a reduced Brownian network as follows: first normalize the rows of K to obtain an orthonormal matrix A; then determine a complementary matrix B such that (A', B') is a basis for $\mathbb{R}^{12}$, using Gaussian elimination to solve the system of

linear equations $BA' = 0$; finally, again using Gaussian elimination, determine M by solving $MRB' = 0$, and then set $G = MRH'$ in accordance with the remark following Proposition 2, so that $MR = GK$. One possible outcome of this computation is

$$M = \begin{bmatrix} 2 & 0 & 2 & 2 & 0 & 6 & 4 & 2 \\ 3 & 3 & 3 & 4 & 1 & 6 & 3 & 3 \end{bmatrix} \tag{53}$$

and

$$G = \begin{bmatrix} 1 & 0 & 2 & -1 & -1/2 & -3/2 \\ 0 & 1 & 3 & -3/2 & -1 & -3/2 \end{bmatrix}. \tag{54}$$

Of course, any other 2×8 matrix having the same row space as (53) may be obtained for M, depending on how the computations are structured, and then G must be chosen so that $MR = GK$.

In the preceding paragraph we have emphasized the fact that one can mechanically compute the data of a reduced Brownian network, given the framework laid out in Sections 1 and 2. However, to get more insight into the effects of dynamic routing, we now describe a more conceptual approach to analysis of the three-station example. It will be helpful to partition the matrices R and K specified by (51) and (52) as follows:

$$R = \begin{bmatrix} R_1 & R_2 \end{bmatrix} \quad \text{and} \quad K = \begin{bmatrix} K_1 & 0 \\ 0 & K_2 \end{bmatrix}, \tag{55}$$

where R_1 and R_2 are 8×8 and 8×4, respectively, and K_1 and K_2 are 3×8 and 3×4, respectively. Readers will note that $R_1 = (I - P')D^{-1}$, where P and D are defined as in the previous section, implying that

$$R_1^{-1} = DQ = D(I + P + P^2 + \cdots)'.$$

A general vector $z \in \mathscr{N}$ has the form $z = R_1 y_1 + R_2 y_2$ where $K_1 y_1 = 0$ and $K_2 y_2 = 0$. That is, $\mathscr{N} = \mathscr{N}_1 \oplus \mathscr{N}_2$ where

$$\mathscr{N}_1 = \{z \in \mathbb{R}^8 \colon z = R_1 y_1,\ K_1 y_1 = 0,\ y_1 \in \mathbb{R}^8\}$$

and

$$\mathscr{N}_2 = \{z \in \mathbb{R}^8 \colon z = R_2 y_2,\ K_2 y_2 = 0,\ y_2 \in \mathbb{R}^4\}.$$

In the obvious way, let us denote by $\mathscr{M}_1$ the orthogonal complement of $\mathscr{N}_1$. Proceeding exactly as in the previous section, one finds that $\mathscr{M}_1$ is the row space of

$$M_1 = K_1 R_1^{-1} = K_1 DQ = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 2 & 2 & 0 \\ 3 & 3 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 2 & 1 & 1 \end{bmatrix}, \tag{56}$$

and M_1 is the "workload content matrix" for our multiclass network; that is, its (k, j)th element is the expected total work for server k embodied in a class

j job ($k = 1, 2, 3$ and $j = 1, \ldots, 8$). On the other hand, readers can verify that $\mathcal{N}_2$ is the one-dimensional vector space spanned by

$$N_2 = [1, 0, -1, 0, 0, 0, 0, 0]. \tag{57}$$

By definition, $\mathcal{M}$ is the orthogonal complement of $\mathcal{N}$, so it consists of all eight-vectors z in the row space of M_1 that are also orthogonal to N_2. That is, one can take $M = \Lambda M_1$ where the rows of Λ are a maximal set of linearly independent three-vectors such that $\Lambda M_1 N_2' = 0$. From (56) and (57) one finds that $M_1 N_2'$ is the 3×1 matrix $(2, 3, -1)'$, so we can take

$$\Lambda = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \end{bmatrix}, \tag{58}$$

and then $M = \Lambda M_1$ is the same 2×8 matrix displayed earlier in (53). From (58) and the previous interpretation of M_1 we have the following interpretation of the "workload process" $W(t) = M\,Z(t)$ that appears in our reduced Brownian network model. Its first component $W_1(t)$ represents the total work for server 1 embodied in all customers that are present in the network at time t, plus two times the total work for server 3 embodied in those customers, and its second component $W_2(t)$ similarly represents total work for server 2 plus three times the total work for server 3. Thus one might say that the two components of $W(t)$ represent *weighted total workloads* for two *overlapping resource pools*, the first composed of servers 1 and 3 and the second composed of servers 2 and 3.

The state space S for our workload process $W(t) = MZ(t)$ consists of all two-vectors w representable as positive linear combinations of the columns of the matrix M displayed in (53). The extreme rays of this positive cone S are the vectors $(0, 3)$ and $(4, 3)$, so it is the shaded region pictured in Figure 2. In the reduced Brownian network for our three-station example with dynamic routing, a system manager must first choose a six-vector U of cumulative idleness controls so as to keep W within the feasible region S. The main system

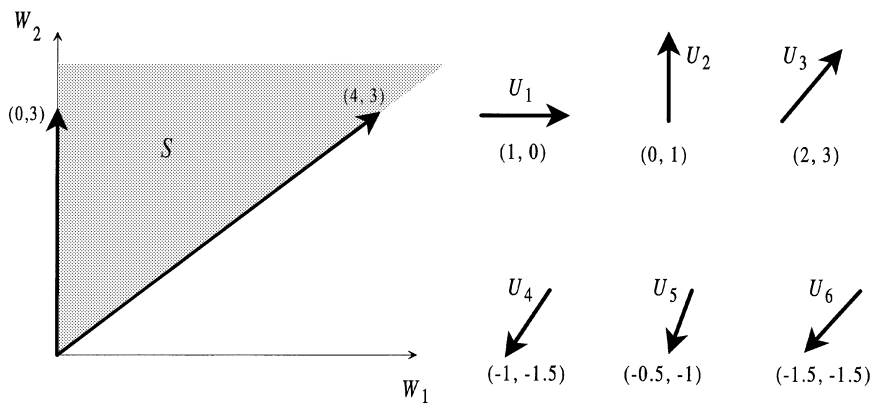

FIG. 2. *State space S and the directions of available control for the three-station open network.*

equation $W(t) = w + \xi(t) + GU(t)$ says that W evolves as a Brownian motion ξ in the absence of server idleness, and by idling server k the system manager can instantaneously displace W in a direction given by the kth column of the matrix G displayed in (54). Those six directions of available control are also pictured in Figure 2, each being labeled with its associated control component. Having chosen an idleness control U in the reduced Brownian network, and thus determined the workload process W, the system manager can at each instant t choose any desired queue length vector $Z(t)$ which is consistent with $W(t)$, meaning that $W(t) = MZ(t)$.

Kelly and Laws [6] provide an excellent survey of research to date on Brownian models of dynamic alternate routing. In each of the studies that they summarize, as in our three-station example, alternate routing capabilities lead to a reduced Brownian network of lower dimension than the one which would have obtained without such capabilities. That is, if dynamic routing capabilities are intelligently exercised in heavily loaded networks, subtle and surprising types of *resource pooling* generally occur. Readers will find that all of the examples discussed by Kelly and Laws lead to Brownian models having the special structure portrayed in (55): R_1 and K_1 are the input–output matrix and capacity consumption matrix, respectively, that would have obtained without dynamic routing, while R_2 and K_2 describe the dynamic routing capabilities that are available. Thus one ultimately expresses system status by means of a "workload process" $W(t) = MZ(t)$ where $M = \Lambda M_1$ and $\Lambda M_1 N_2' = 0$, exactly as in our three-station example. One might describe Λ as a *pooling matrix*: its row dimension d is less than or equal to its column dimension p, and $d - p$ is the reduction in effective system dimension due to dynamic routing capabilities. The generality of this representation will be discussed further in Section 7.

In each of the examples discussed by Kelly and Laws [6], the reduction in effective system dimension referred to above is equal to the number of "degrees of freedom" in routing new arrivals, but that effect is not general: adding an element of routing discretion to a queueing network model *may* reduce by one the effective system dimension, but only if the new capability is in some sense "linearly independent" of other model elements. To make this statement precise, one must connect the row dimension of our pooling matrix Λ with structural features of the matrices M_1 and N_2. Development of that general idea would seem to be a promising area for future research.

6. Dynamic sequencing in closed queueing networks. Consider a multiclass queueing network model like the one described in Section 3, except that there are no external arrivals (i.e., $\lambda_j = 0$ for each class j) and the switching matrix P is stochastic (i.e., each row of P sums to one). Thus any jobs initially present at time 0 circulate perpetually among stations of the network, with no arrivals and no departures, and we assume that the number of jobs in the network is relatively large. Also, P is assumed to be irreducible, which means that a job of any given class eventually visits all other classes with probability one. All of the notation established in Section 3 carries over to the closed network case except as noted later.

Because the $m \times m$ matrix P is irreducible, the rank of $I - P$ is $m - 1$. Thus the input–output matrix $R = (I - P')D^{-1}$ is also of rank $m - 1$, so there exists an m-vector β satisfying $R\beta = 0$, and β is unique up to a scale constant. One obvious solution of this identity is $\beta = D\pi$, where π is the stationary distribution of P (note that all components of β are then strictly positive), but the following alternative scaling will prove to be convenient: defining a p-vector ρ of relative traffic intensities via $\rho = K\beta$, we scale β so that $\max(\rho_1, \ldots, \rho_p) = 1$. This same scaling convention was used in Section 8 of [2], where β was interpreted as a set of *nominal activity rates*. That is, ρ_k represents the largest possible long-run utilization rate for server k, and β_j is the fraction of time that server $k = \sigma(j)$ would devote to service of class j if it were to attain utilization rate ρ_k.

The problem of interest is that of dynamic sequencing in the multiclass closed network. The associated Brownian network model was developed in Section 8 of [2], and here we shall repeat only its salient features. First, one can choose a state space scaling such that the initial queue length vector z in the fundamental system equation (4) satisfies $e'z = 1$. That is, in forming the Brownian system model we express the queue length for each job class as a fraction of the fixed population size. Second, the Brownian motion X appearing in (4) also satisfies $e'X(t) = 0$: that is, its drift vector μ satisfies $e'\mu = 0$, and its covariance matrix Σ satisfies $e'\Sigma e = 0$. Finally, $e'R = 0$ because $e'P' = e'$, so premultiplying both sides of (4) by e' gives $e'Z(t) = e'z = 1$, which is consistent with the physical model under consideration.

To develop an equivalent workload formulation for this closed Brownian network, let us first define the $m \times m$ matrix (recall that π is the stationary distribution of P, represented as a column vector)

$$\Pi = e\pi' = \lim_{i \to \infty} \frac{1}{i} \sum_{j=1}^{i} P^j. \tag{59}$$

In our earlier treatment of open networks, the matrix Q defined in (42) played a central role. For a closed network model with irreducible switching matrix, the appropriate analog is

$$Q = (I - P' + \Pi')^{-1}. \tag{60}$$

Kemeny and Snell [7] call the transpose of Q the *fundamental matrix* of the Markov chain with transition matrix P; in addition to proving existence of the inverse that defines Q, they derive a number of useful identities, including

$$Q\pi = \pi. \tag{61}$$

PROPOSITION 5. *For the closed Brownian network described above, the space $\mathscr{M}$ has dimension p.*

PROOF. The p rows of K are linearly independent, so its null space $\mathscr{B}$ has dimension $m - p$. Recall that B denotes an $(m - p) \times m$ orthonormal matrix whose rows span $\mathscr{B}$, and that $\mathscr{N}$ is the column space of RB'. Suppose that the

$m - p$ columns of RB' are linearly dependent. Then there exists a nontrivial vector $y \in \mathscr{B}$ satisfying $Ry = 0$. As noted above, this implies $y = c\beta$ for some constant c. But $K\beta = \rho$, so β is not an element of $\mathscr{B}$, and we have arrived at a contradiction. That is, the $m - p$ rows of RB' are linearly independent, implying that $\mathscr{N}$ has dimension $m - p$, and hence its orthogonal complement $\mathscr{M}$ has dimension p. □

PROPOSITION 6. *Let $\widehat{G}$ be any $(p-1) \times p$ matrix of rank $p-1$ such that $\widehat{G}\rho = 0$, and let $\widehat{M} = \widehat{G}KDQ$. Define a $p \times m$ matrix M by adding to $\widehat{M}$ a final row of ones, and a $p \times p$ matrix G by adding to $\widehat{G}$ a final row of zeros. Then M is of rank p, and $MR = GK$.*

COROLLARY. *The rows of M span $\mathscr{M}$.*

PROOF. First observe that $DQR = DQ(I - P')D^{-1} = DQ(I - P' + \Pi' - \Pi')D^{-1} = D(I - Q\Pi')D^{-1} = (I - DQ\Pi'D^{-1})$. Now $\Pi' = \pi e'$, so $Q\Pi' = \Pi'$ by (61). Moreover, $D\pi = c\beta$ where c is a positive normalization constant, so $DQ\Pi'D^{-1} = c\beta e'D^{-1}$. Recalling that $\rho = K\beta$, we then have $KDQR = K - c\rho e'D^{-1}$. Since $\widehat{G}\rho = 0$ by assumption, this implies that

$$\widehat{M}R = \widehat{G}KDQR = \widehat{G}K. \tag{62}$$

Recall that $e'R = 0$ because $e'P' = e'$. The final row of M is e', so the final row of MR is zero. The final row of $\widehat{G}$ is also zero by definition, and combining this with (62) gives $MR = GK$.

To complete the proof of Proposition 6, it remains only to show that M has rank p. Given the full row rank of $\widehat{G}$ and the special structure of K, it is easy to show by contradiction that $\widehat{G}K$ has rank $p-1$. Since the $m \times m$ matrix DQ is nonsingular, it follows that $\widehat{M} = \widehat{G}KDQ$ has full row rank. Finally, to prove by contradiction that e' (the last row of M) does not lie in the row span of $\widehat{M}$, suppose that $v'\widehat{M} = e'$ for some nontrivial vector v. Substituting the definition of $\widehat{M}$, this is equivalently stated as $v'\widehat{G}KDQ = e'$. Postmultiplying both sides of that equation by $(I - P' + \Pi')D^{-1}$, and noting again that $e'P' = e'\Pi' = e'$, we then obtain $v'\widehat{G}K = e'D^{-1}$. Now postmultiplying by the vector β and recalling that $K\beta = \rho$, we have $v'\widehat{G}\rho = e'D^{-1}\rho$. Since $\widehat{G}\rho = 0$ by assumption while $e'D^{-1}\rho > 0$, this is a contradiction. The proof of Proposition 6 is thus complete, and the corollary is immediate from Proposition 2. □

Any pair of matrices M and G defined as in Proposition 6 will suffice for the purposes of the reduced Brownian network model (13)–(18). The last element of the workload process $W(t) = MZ(t)$ is then $e'Z(t)$, and it has already been noted that $e'Z(t) = 1$ for all $t \geq 0$ in a closed network model, regardless of what control Y is chosen. Thus one can simply delete the last component of W. That is, one ultimately arrives at a reduced Brownian network model where

the state of the system at time t is summarized by a $p-1$ dimensional process $\widehat{W}(t) = \widehat{M}Z(t)$ satisfying

$$\widehat{W}(t) = \widehat{w} + \widehat{\xi}(t) + \widehat{G}U(t) \quad \text{for all } t \geq 0, \tag{63}$$

where $\widehat{w} = \widehat{M}z$ and $\widehat{\xi}(t) = \widehat{M}X(t)$. To develop an interpretation of this process we begin with the identity

$$Q = I + \sum_{i=1}^{\infty} (P^i - \Pi)', \tag{64}$$

which is a standard result in Markov chain theory (cf. [7]). Let us now define the infinite series of $p \times m$ matrices

$$H(n) = KD\left(I + \sum_{i=1}^{n} P^i\right)' \quad \text{for } n = 1, 2, \ldots . \tag{65}$$

(The letters H and n were used with other meanings in previous sections, but they have not been needed in the discussion of closed network models, so the temporary reuse of notation should cause no confusion.) One may interpret $H_{kj}(n)$ as the expected total work required from server k in completing the first n services for a job that begins in class j. Combining (64) and (65) with the definition of $\widehat{M}$ gives $\widehat{M} = \widehat{G}KDQ = \lim[\widehat{G}H(n) - n\widehat{G}KD\Pi']$. It was noted in the proof of Proposition 6 that $\widehat{G}KD\Pi' = 0$ because $\widehat{G}\rho = 0$ by definition, and thus we have the following:

$$\widehat{M} = \lim_{n\to\infty} \widehat{G}H(n). \tag{66}$$

Assuming, without loss of generality, that $\rho_p = 1$ (recall that $\max \rho_k = 1$ as a matter of convention), one feasible choice of $\widehat{G}$ is

$$\widehat{G} = \begin{pmatrix} 1 & & & -\rho_1 \\ & 1 & & -\rho_2 \\ & & \ddots & -\rho_2 \\ & & 1 & -\rho_{p-1} \end{pmatrix}, \tag{67}$$

and with this choice (66) becomes

$$\widehat{M}_{kj} = \lim_{n\to\infty} \left[H_{kj}(n) - \rho_k H_{pj}(n)\right] \tag{68}$$

for all $k = 1, \ldots, p-1$ and $j = 1, \ldots, m$. In the terminology of Harrison and Wein [5] and Chevalier and Wein [1], (68) is a measure of *workload imbalance*. That is, $\widehat{M}_{kj}$ represents in a certain sense the difference between the expected future work for server k embodied in a job of class j and the expected future work for server p embodied in that same job. Thus each component $k = 1, \ldots, p-1$ of the process $\widehat{W}(t) = \widehat{M}Z(t)$ expresses the imbalance between work for server k and work for server p embodied in the queue length vector $Z(t)$.

In their treatment of multiclass closed network models, both Harrison and Wein [5] and Chevalier and Wein [1] develop a reduced Brownian network with a $(p-1)$-dimensional state descriptor $\widehat{W}(t) = \widehat{M}Z(t)$ as above, calling $\widehat{W}(t)$ the *workload imbalance process* and $\widehat{M}$ the *workload imbalance profile matrix.* In fact, several alternative definitions of the workload imbalance process are advanced in those papers and shown to be equivalent. Our definition of $\widehat{W}(t)$ is also equivalent to any one of those definitions in the following sense. Denoting by Δ the difference of our workload imbalance profile matrix $\widehat{M} = \widehat{G}KDQ$ and any of the corresponding matrices used by Harrison and Wein [5] or Chevalier and Wein [1], it can be shown that $\Delta = ue'$, where u is some $(p-1)$-vector and e is the p-vector of ones. Thus our workload imbalance process $\widehat{W}(t)$ differs from that in the earlier papers by a vector of the form $ue'Z(t)$, but $e'Z(t) = 1$ for all $t \geq 0$ as noted earlier, so the difference is a constant vector u which is uncontrollable and can therefore be ignored. Two appealing features of the definition $\widehat{M} = \widehat{G}KDQ$ in Proposition 6 are its symmetry with respect to both servers and job classes, and the connection it makes with the fundamental matrix Q of classical Markov chain theory.

7. Two open problems. In Section 5 we analyzed a three-station open network model with dynamic alternate routing, and derived an attractive representation for the matrix M appearing in its equivalent workload formulation. To be specific, it was found that $M = \Lambda M_1$, where M_1 is the nonnegative workload profile matrix for a corresponding open network with random routing, and Λ is a nonnegative pooling matrix. One naturally asks whether this result is general, and if so, exactly how M_1 and Λ are defined in terms of the original network data. Of course, one further wants a characterization of Λ that gives insight as to which alternate routing capabilities are most effective in reducing system dimension. Kelly and Laws [6] argue that such insights are among the most important ones to be gained from heavy traffic analysis of queueing networks, and sharp general results may in fact be attainable.

In traditional queueing network models, each processing activity involves a single server acting on a single job of a particular class. In some applications, however, two or more servers may be required simultaneously for some processing activities, as when a machine and an operator work together in manufacturing. When a stochastic processing network involves such simultaneous resource requirements, the capacity consumption matrix K for its natural Brownian analog may have more than one nonzero element per column. It would be interesting to know how that structural change manifests itself in the Brownian network's equivalent workload formulation, and in particular, whether any novel types of resource pooling may result.

APPENDIX

Our standard formulation (1)–(5) of a Brownian network model includes the state-space constraint $Z(t) \geq 0$. However, more general state-space con-

straints can actually be accommodated within the standard formulation as follows. Let S be a convex polyhedral subset of $\mathbb{R}^m$ and suppose that (3) is replaced by the requirement that $Z(t) \in S$ for all $t \geq 0$. Of course, S can be represented as $S = \{x \in \mathbb{R}^m\colon A\,x \leq b\}$ for some matrix A and vector b. Defining $z^* = b - Az$, $R^* = -AR$ and $X^*(t) = -AX(t)$, we then replace conditions (3) and (4) by

$$Z^*(t) \geq 0 \quad \text{for all } t \geq 0, \text{ where} \tag{69}$$

$$Z^*(t) = z^* + X^*(t) + R^*\,Y(t) \quad \text{for all } t \geq 0. \tag{70}$$

The definition of an admissable control Y is completed by adding conditions (1), (2) and (5) to (69) and (70). This brings us to a Brownian network model having the standard form (1)–(5).

Acknowledgments. We are grateful to Jim Dai and Ruth Williams for helpful comments on an earlier draft and to an anonymous referee for mentioning the second open problem described in Section 7.

REFERENCES

[1] CHEVALIER, P. B. and WEIN, L. M. (1993). Scheduling networks of queues: heavy traffic analysis of a multistation closed network. *Oper. Res.* **41** 743–758.

[2] HARRISON, J. M. (1988). Brownian models of queueing networks with heterogeneous customer populations. In *Stochastic Differential Systems, Stochastic Control Theory and Applications* (W. Fleming and P. L. Lions, eds.) 147–186. Springer, New York.

[3] HARRISON, J. M. (1996). The BIGSTEP approach to flow management in stochastic processing networks. In *Stochastic Networks: Theory and Applications* (F. Kelly, S. Zachary and I. Ziendins, eds.) 57–90. Oxford Univ. Press.

[4] HARRISON, J. M. and WEIN, L. M. (1989). Scheduling networks of queues: heavy traffic analysis of a simple open network. *Queueing Systems* **5** 265–280.

[5] HARRISON, J. M. and WEIN, L. M. (1990). Scheduling networks of queues: heavy traffic analysis of a two-station closed network. *Oper. Res.* **38** 1052–1064.

[6] KELLY, F. P. and LAWS, C. N. (1993). Dynamic routing in open queueing networks: Brownian models, cut constraints and resource pooling. *Queueing Systems* **13** 47–86.

[7] KEMENY, J. G. and SNELL, J. L. (1976). *Finite Markov Chains*. Springer, New York.

[8] KUSHNER, H. J. and MARTINS, L. F. (1990). Routing and singular control for queueing networks in heavy traffic. *SIAM J. Control Optim.* **28** 1209–1233.

[9] LAWS, C. N. (1992). Resource pooling in queueing networks with dynamic routing. *Adv. in Appl. Probab.* **24** 699–726.

[10] LAWS, C. N. and LOUTH, G. M. (1990). Dynamic scheduling of a four-station queueing network. *Probab. Engrg. Inform Sci.* **4** 131–156.

[11] REIMAN, M. I. (1982). The heavy traffic diffusion approximation for sojourn times in Jackson networks. In *Applied Probability–Computer Science: The Interface* (R. L. Disney and T. J. Ott, eds.) **2** 409–422. Birkhäuser, Boston.

[12] REIMAN, M. I. (1983). Some diffusion approximations with state space collapse. *Proceedings of International Seminar on Modelling and Performance Evaluation Methodology, Berlin, 1983. Lecture Notes in Control and Inform. Sci.* **60** 209–240. Springer, Berlin.

[13] REIMAN, M. I. (1984). Open queueing networks in heavy traffic. *Math. Oper. Res.* **9** 441–458.

[14] VAN MIEGHEM, J. A. (1995). Dynamic scheduling with convex delay costs: the generalized $c\mu$ rule. *Ann. Appl. Probab.* **5** 809–833.

[15] WEIN, L. M. (1990). Scheduling networks of queues: heavy traffic analysis of a two-station network with controllable inputs. *Oper. Res.* **38** 1065–1078.

[16] WEIN, L. M. (1991). Brownian networks with discretionary routing. *Oper. Res.* **39** 322–340.
[17] WEIN, L. M. (1992). Scheduling networks of queues: heavy traffic analysis of a multistation network with controllable inputs. *Oper. Res.* **40** S312–S334.

GRADUATE SCHOOL OF BUSINESS
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-5015
E-MAIL: fharrisonj@gsb.stanford.edu

J.L. KELLOGG GRADUATE SCHOOL
OF MANAGEMENT
NORTHWESTERN UNIVERSITY
EVANSTON, ILLINOIS 60208-2009
E-MAIL: vanmieghem@nwu.edu