The Annals of Applied Probability
2003, Vol. 13, No. 3, 1119–1150
© Institute of Mathematical Statistics, 2003

A BROADER VIEW OF BROWNIAN NETWORKS

BY J. MICHAEL HARRISON

Stanford University

This paper describes a general type of stochastic system model that involves three basic elements: activities, resources, and stocks of material. A system manager chooses activity levels dynamically based on state observations, consuming some materials as inputs and producing other materials as outputs, subject to resource capacity constraints. A generalized notion of heavy traffic is described, in which exogenous input and output rates are approximately balanced with nominal activity rates derived from a static planning problem. A Brownian network model is then proposed as a formal approximation in the heavy traffic parameter regime. The current formulation is novel, relative to models analyzed in previous work, in that its definition of heavy traffic takes explicit account of the system manager's economic objective.

1. Introduction. Brownian networks are a class of stochastic system models that provide crude but relatively tractable representations for problems of dynamic resource allocation. Such dynamic control problems arise in a wide range of economic and technological settings, from telecommunications and computing to manufacturing and service operations. Roughly speaking, Brownian networks are appropriate as approximate models of systems where the ambient mode of operation is characterized by balanced, high-volume flow of work or material, with inventories and backlogs fluctuating over time as a result of stochastic variability. In addition to their generality and relative tractability, Brownian networks have the virtue of mathematical elegance, requiring a minimum of data and having a compact mathematical description.

On the other hand, optimal control policies for Brownian networks often require subtle interpretations. To develop intuition in that regard, it is customary at this stage in the development of the subject to speak of Brownian networks as approximations for models of more conventional type, rather than just accepting them as system models in their own right. To be specific, Brownian networks arise as "heavy traffic" approximations for conventional stochastic system models, and Williams [2, 10] has described (in slightly different words) the following five levels of formulation and analysis to guide potential users of Brownian networks:

(a) Formulate a conventional stochastic system model, with an associated dynamic control problem.

Received November 2000; revised August 2002.

AMS 2000 subject classifications. Primary 90B15, 60K25; secondary 60J70.

Key words and phrases. Stochastic processing networks, Brownian networks, heavy traffic, stochastic control.

(b) Identify a limiting parameter regime that formalizes the notion of heavy traffic.

(c) Formulate a Brownian network model, with an associated Brownian control problem that plausibly represents the "heavy traffic limit" of the original control problem.

(d) Solve the Brownian control problem and "interpret" that solution, translating it into a proposed control policy for the original system.

(e) Show that the proposed policy is "asymptotically optimal" in the heavy traffic limit, its limiting performance being that associated with the optimal solution of the Brownian control problem.

Contributions of this paper. Focusing exclusively on levels (a)–(c) of the analytical hierarchy just described, but addressing them in roughly reverse order, this paper extends the general theory of Brownian networks, or Brownian network models, that was initiated in [4], then developed further in [6] and [8]. To be more precise, this paper continues to develop one of the two themes in [6], as follows.

In [6] it was argued that Brownian networks may serve as approximations for a broad class of stochastic processing networks, which include as a special case the multiclass queuing networks that were emphasized in [4]. Here we generalize that argument by removing the restriction to "open" processing networks that was imposed in [6]. That is, the theory developed in this paper includes processing systems with exogenous *outputs* as well as exogenous inputs and systems where all flows are endogenous, including "closed" processing networks. This broadening of the application domain for Brownian networks is not only important from a practical standpoint but also aesthetically pleasing. A related contribution of this paper is to generalize somewhat the economic structure that has been considered previously in conjunction with Brownian networks [4, 6, 8].

The most important contribution of this paper is to generalize the notion of heavy traffic that was developed in [6] as the setting for Brownian network approximations. The key element of this generalized treatment is the explicit consideration of costs and revenues, in addition to physical data, in the analysis. An illustrative parallel-server system will be described (see Section 3) that *is* in heavy traffic according to the definition proposed here, but *not* according to the definition used in earlier works.

The contributions described above are modeling contributions, rather than hard mathematical results, because the central conclusions are supported only by formal limits of the kind used in [4] and [6]. That is, this paper provides nonrigorous arguments intended to elucidate the proper formulation and proper interpretation of Brownian network models. A second major theme in [6], which will *not* be dealt with here, concerned the reduction of a Brownian control problem to its equivalent workload formulation. The generalized definition of a Brownian network proposed in this paper requires a corresponding generalization of that theory, but after the deficiencies of extant theory have been explained, the task of formulating a remedy will be reserved for future research.

As stated above, the theory of Brownian networks developed here is more general in several important ways than what was described in [4] and [6], but it is also more restrictive in one regard: we assume a bounded state space throughout the main development, explaining afterward the potential complications that may arise if that restriction is lifted. The bounded state space assumption is by no means essential, but if it is relaxed, other restrictions on the model data must be introduced (see Section 9), and associated technicalities tend to distract attention from the main ideas.

Structure of the paper. First, the generalized definition of a Brownian network is laid out in Section 2, with relatively little in the way of justification or interpretation. The generic interpretation of the Brownian network as a limit of conventional system models is developed in Sections 3–6, following an expositional path similar to the one in [6]. Notation introduced originally in Section 2 will be reused in Sections 3 and 4, in order to establish correspondences between elements of the Brownian network model and elements of the conventional model that it approximates. The exposition begins with consideration of a deterministic planning problem in Section 3, which provides the means of articulating a *balanced loading assumption* that is one essential ingredient for justifying a Brownian network model. The static planning problem considered here has profit maximization, or value maximization, as its objective, whereas the corresponding static optimization problem considered in [6] had as its objective the minimization of a uniform upper bound on resource utilization.

The static planning model of Section 3 is expanded into a full-blown stochastic processing network in Section 4, and then Section 5 explains how the Brownian model laid out in Section 2 can be viewed as a formal limit of that more conventional stochastic network model. To be more precise, Section 5 explains how a Brownian network *with zero drift* can be viewed as the limit of conventional stochastic processing networks that satisfy *exactly* the balanced loading assumption articulated in Section 3. That argument is extended in Section 6, which explains how a Brownian network with nonzero drift arises as the formal limit of conventional models that satisfy our balanced loading assumption in a suitable approximate sense. Sections 4–6 are written with two distinct but complementary goals in mind: the first is to explain in concrete terms how a Brownian network approximation can be formulated for a given conventional model in the relevant parameter regime; the other is to support such approximations by means of formal limiting arguments.

Because virtually all of the arguments developed in Sections 3–6 have precise analogs in [6], most of that paper's lengthy introduction serves equally well to frame the issues addressed here. Also, [6] contains a number of concrete examples, some of them quite elaborate, which illustrate the problems and special structures that have motivated the development of Brownian network theory up to now. Section 7 of this paper analyzes a parallel-server example, introduced originally in Section 3, that has a somewhat different character: as noted earlier, its data do not conform to the notion of heavy traffic advanced in [6] but *do* satisfy the assumptions imposed in this paper.

In Section 8, certain observations made in conjunction with the parallel-server example are recast in a more general setting. There we focus on the class of stochastic processing networks in which each activity consumes the capacity of at most one resource (most network models that one encounters in the published literature have this property), showing how that special structure simplifies both heavy traffic analysis and its interpretation.

Section 9 contains a brief discussion of the generalized Brownian network model laid out in Section 2. Given the restrictions we impose on its data, the model is shown to pass a first and most obvious test of internal consistency. Other foundational questions are also posed, but their resolution is left to later work. Additional issues that would arise with an unbounded state space are described, and those issues are connected with the limiting arguments presented in Section 5.

As suggested earlier, the expositional sequence just described is not what one might naturally expect: in particular, Sections 2–4 of this paper address the first three levels of the analytical hierarchy (a)–(e) in roughly reverse order. Readers who are new to the subject matter may find it easiest to just skim Section 2 initially and then refer back to it as necessary.

Notation and terminology. All vectors should be envisioned as column vectors; the transpose of a vector v is denoted v'. The scalar product of two vectors u and v is denoted $u \cdot v$ as usual. When we say that a multidimensional stochastic process $X = \{X(t), t \geq 0\}$ is a (θ, Σ) Brownian motion, this means that the associated drift vector is θ, the associated covariance matrix is Σ, and the initial state is $X(0) = 0$ almost surely. At some points the more compact notation $BM(\theta, \Sigma)$ is used for that same process. All continuous-time stochastic processes used in this paper will be assumed to have paths that are right-continuous with finite left limits (RCLL). Let $D^n[0, \infty)$ denote the space of RCLL functions from $[0, \infty)$ into $\mathbb{R}^n$ (here n is a positive integer), endowed with the usual Lindvall–Skorohod J_1 topology (cf. Section 16 of Billingsley [1]). The symbol "$\Rightarrow$" is used to denote convergence in distribution for stochastic processes whose paths lie in $D^n[0, \infty)$. Given a real-valued function $f(\cdot)$ defined on $[0, \infty)$ and a real constant a, the statement "$f(t) \sim at$ as $t \to \infty$" is understood to mean that $t^{-1} f(t) \to a$ as $t \to \infty$; when $f(\cdot)$ and a are vector valued or matrix valued, such statements are understood componentwise.

2. Generalized description of a Brownian network.

Let m, n, and p be positive integers. Let $X = \{X(t), t \geq 0\}$ be an m-dimensional (θ, Σ) Brownian motion with respect to a given filtration on a fixed probability space (cf. page 47 of Karatzas and Shreve [9]). (In general, the filtration can be bigger than that generated by the Brownian motion X, but $\{X(t)-\theta t,\ t\geq 0\}$ must be a martingale relative to the filtration and thus the filtration does not contain information about future increments of X.) All stochastic processes discussed in this section are understood to be defined on that same filtered probability space. In addition to θ and Σ, the data of our Brownian network model include an $m\times n$ matrix R, a $p\times n$ matrix K, an m-vector z, a compact and convex subset S of $\mathbb{R}^m$ that has a nonempty interior, a continuous function $h:S\to\mathbb{R}$, and an n-vector v. Later in the paper, various additional assumptions will be made about these data (see Sections 3, 4 and 6).

An *admissible control* for the Brownian network model is an n-dimensional process $Y=\{Y(t),t\geq 0\}$ that is adapted to the given filtration and also satisfies the additional restrictions specified in the next paragraph. The five relationships that define the Brownian network model, or Brownian system model, are as follows:

$$Z(t)=z+X(t)+RY(t)\qquad\text{for all } t\geq 0, \tag{1}$$

$$U(t)=KY(t)\qquad\text{for all } t\geq 0, \tag{2}$$

$$Z(t)\in S\qquad\text{for all } t\geq 0, \tag{3}$$

$$U(\cdot)\text{ is nondecreasing with } U(0)\geq 0 \tag{4}$$

and

$$\xi(t)=\int_0^t h(Z(s))\,ds+v\cdot Y(t)\qquad\text{for all } t\geq 0. \tag{5}$$

General interpretations of the model data, and of the processes Z, U and ξ defined in terms of Y by means of (1), (2) and (5) will be developed in the sections that follow. For the time being we simply call $Z(t)$ the "state of the system at time t." Thus (1) describes a system where state dynamics are linear in the chosen control Y and are subject to Brownian noise.

From the perspective of conventional stochastic control theory, a notable feature of our Brownian network model is that components of the control Y are not required to be monotone or even of bounded variation. However, in addition to being adapted, an admissible control Y must have RCLL sample paths and must satisfy (3) and (4).

In most Brownian network models studied to date, the state space S appearing in (3) has been the nonnegative m-dimensional orthant, but that case is ruled out by our restriction to compact S. Of course, a compact state space is intrinsic to closed network models of the kind discussed in Section 6 of [8], and there are also many natural problems where an optimal policy confines Z to a compact region, even though no a priori bounds on Z are specified. Also, one might argue that the restriction to bounded state space is of no practical importance, but the unbounded case is of interest theoretically, because it includes the Brownian analogs of most classical queuing control problems; the possible extension of our theory to unbounded S will be discussed briefly in Section 9.

The quantity $\xi(t)$ defined by (5) is interpreted as the cumulative "cost" incurred by the system manager up to time t. This wording is intended to convey the notion that smaller values of $\xi(\cdot)$ are always preferable, but to obtain a complete problem formulation one obviously must specify a concrete objective. For example, one might strive to minimize expected cost over a particular finite time horizon, or minimize expected discounted cost over an infinite time horizon, but the specific objective will be largely irrelevant for our purposes. The economic structure reflected in (5) is more general than what has been considered in previous treatments of Brownian networks [4, 6, 8]. In particular, previous formulations have assumed, either implicitly or explicitly, that the second term $v \cdot Y(t)$ actually depends on the control Y only through the monotone process U defined by (2). Here we argue that the more general case described by (5) is well motivated by applications, and equivalent workload formulations are more subtle and complex in the general case (see Section 9).

3. A static planning problem.

Proceeding exactly as in Section 2 of [6], but modifying notation slightly, we consider a processing system with ℓ different resources that consume and produce m distinct materials (or stocks, or job classes) by means of n different processing activities. To be more precise, we consider here a static planning problem associated with a deterministic fluid model of such a system (cf. Sections 2 and 3 of [7]). The underlying fluid model will not be spelled out, because it is not actually needed in the mathematical development here, but it is implicitly referred to at several points in the text.

Let us denote by R_{ij} the average amount of material i consumed per unit of activity j, with a negative value interpreted to mean that activity j is a net producer of material i. The $m \times n$ input–output matrix R will eventually appear in the fundamental system equation (1) of our Brownian network model.

Next, let A_{kj} be the amount of resource k capacity consumed per unit of activity j, and let q_k be the quantity of resource k capacity available per unit of time. It might be, for example, that resource k is a group of interchangeable machines, that capacity of resource k is expressed in machine hours, that time is measured in weeks, and that one unit of activity j corresponds to producing one output ton of a given product. Then A_{kj} would be the number of machine hours consumed per output ton produced, and q_k would be the number of machine hours available per week. The $\ell \times n$ capacity consumption matrix $A = (A_{kj})$ is nonnegative, and it will ultimately be incorporated in the matrix K that appears in (2) of our Brownian network model. All components of q are assumed to be strictly positive.

The next element of our static planning problem is an m-vector λ, each component of which may be either positive, negative or zero. If λ_i is positive, it represents the average rate at which material i is automatically supplied by external sources (i.e., λ_i represents an exogenous input rate), and if λ_i is negative, then $|\lambda_i|$ represents the average rate at which the system manager is committed to provide material i as an output. More will be said about the vector λ of exogenous flow rates in the next section, where a more detailed model of dynamic system control is introduced.

The last of the data for our static planning problem is an n-vector v of net "value rates" associated with the various processing activities. That is, we assume that a unit of activity j generates v_j units of "value" on average, with a negative value interpreted as a net "cost." The case where $v = 0$ is of considerable interest (in fact, most previous work on Brownian networks has focused on this case), and it is *not* excluded in the general development to follow.

Ignoring stochastic variability associated with processing activities and with exogenous inflows and outflows, and thus ignoring all congestion-related and backlog-related costs, a system manager might plausibly seek a solution to the following *static planning problem*: find an n-vector x of average activity rates (each component x_j is expressed in units of activity per unit of time) so as to

$$\text{maximize } v \cdot x \tag{6}$$

subject to the constraints

$$Rx = \lambda, \qquad Ax \le q \quad \text{and} \quad x \ge 0. \tag{7}$$

In the static planning problem (6) and (7), one seeks to maximize the net rate at which value is generated by processing activities, subject to three sets of constraints. The constraints embodied in $Rx = \lambda$ require that exogenous inputs be processed to completion, and that exogenous output requirements be satisfied, without inventories or deficits of any materials developing. The constraints embodied in $Ax \le q$ require that the capacity limitations of all resources be respected, and finally, all activity levels must be nonnegative. The balanced loading assumption referred to in Section 1 is the following.

ASSUMPTION 1. The static planning problem (6)–(7) has a unique optimal solution x^*, and moreover $Ax^* = q$.

Hereafter x^* will be called the system manager's *nominal processing plan*. This name reflects the fact that x^* is derived from a naive or idealized planning model in which stochastic variability is suppressed. In the presence of such variability, it may be desirable for actual activity rates to vary around the nominal rates x_j^* (see Section 4), depending on system status.

Assumption 1, which plays a central role in our formulation of a Brownian network approximation, says that all resource capacities are exhausted under the nominal processing plan x^*. When $v = 0$, Assumption 1 amounts to the following: there exists a unique vector x^* satisfying all of the constraints in (7),

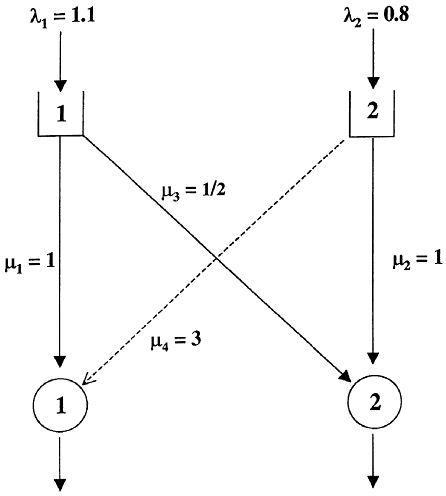

FIG. 1. *A parallel-server example.*

and moreover, that vector satisfies $Ax^* = q$. That is, there exists only one program of average activity levels which processes all exogenous inputs and meets all exogenous output requirements, and that program uses all available capacity. For open processing networks of the kind treated in [6], Bramson and Williams [3] have shown that this is equivalent to the heavy traffic assumption imposed in Section 2 of that paper.

As an example, consider the static planning problem associated with the queuing system portrayed in Figure 1, which is closely related to one studied earlier in [5]. As usual in queuing theory, we imagine that units of flow are discrete; those units will be called "jobs" and processing resources will be called "servers." Here we have $\ell = 2$ servers (represented by the circles in Figure 1) and $m = 2$ job classes that are stored in separate buffers (represented by the open-ended rectangles in Figure 1) as they await processing.

For each job class $i = 1, 2$ the average arrival rate λ_i, expressed in jobs per hour, is as shown in Figure 1. There are a total of $n = 6$ processing activities in our parallel-server example, the first four of which are portrayed in Figure 1. (The numbering of activities is arbitrary, of course.) Each activity $j = 1, \dots, 4$ consists of a particular server processing jobs from a particular buffer, the associated average service rate being μ_j jobs per hour (see Figure 1). With activity levels expressed in server hours, one may alternatively say that $\mu_1, \dots, \mu_4$ each represent an average rate of material flow per unit of activity. For each server $k = 1, 2$ the capacity available per time unit is $q_k = 1$, which means that there is a full server hour available per clock hour. The decision variables $x_1, \dots, x_4$ in our static planning problem (6)–(7) each represent the average number of hours that a particular server devotes to processing jobs from a particular buffer per hour, or equivalently, the fraction of that server's capacity devoted to that buffer.

In addition to the processing activities described above, there are two activities that we use to represent input control capabilities: activities 5 and 6 correspond to the system manager ejecting jobs from buffers 1 and 2, respectively, which we assume can be done at any time without penalty. However, such disposal is irreversible, and thus it deprives the system manager of whatever value might have been derived from processing the jobs ejected. We express activity levels for the two disposal activities directly in number of jobs ejected, so the average rate of material consumption per unit of activity is 1 in both cases, simply as a matter of definition. On the other hand, no capacity constraints will be associated with the disposal activities (i.e., no upper bounds are imposed on the instantaneous activity rates), and so any number of ejections can be enforced in any given time interval, provided that state space constraints are respected; see Section 7 for the details of model specification.

Input control capabilities of the kind just discussed are realistic in many contexts, and they are easily accommodated within the modeling framework developed in this paper, as in the framework developed earlier in [6]. Brownian approximations for network models with input control capabilities have been discussed explicitly in Section 9 of [4] and Section 5 of [8].

Using the format prescribed at the beginning of this section, we summarize the data for the static planning problem associated with our parallel-server example as follows:

$$R = \begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 3 & 0 & 1 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}, \qquad q = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad \lambda = \begin{bmatrix} 1.1 \\ 0.8 \end{bmatrix}. \tag{8}$$

If activities 4–6 were deleted from this parallel-server model (i.e., if server 1 were unable to process class 2 jobs and neither job class could be ejected), and if we were to further take $v = 0$, then the definition of "heavy traffic" advanced in [6] would be satisfied as follows: by devoting all server 1 capacity to class 1, twenty percent of server 2 capacity to class 1, and eighty percent of server 2 capacity to class 2, the system manager can achieve average processing rates that match the average input rates, but there is no way to process these inputs using strictly less than all available capacity. With activity 4 available, it *is* possible to process the exogenous inputs and still have capacity left over, so the system pictured in Figure 1 does not satisfy the heavy traffic assumptions set out in [6].

With regard to system economics, let us suppose that each activity $j = 1, \dots, 4$ generates value at an average rate of y_j hundred dollars *per service completed*, where $y_1 = 1$, $y_2 = 1$, $y_3 = 2$ and $0 < y_4 < \frac{1}{2}$. For each of these activities, then, the average value generated *per unit of activity* (i.e., per server hour devoted to the activity) is $v_j = y_j \mu_j$ hundred dollars. Assuming that there is neither direct cost nor direct benefit associated with activities 5 and 6, we then have the value rate vector

$$v = (1, 1, 1, v_4, 0, 0)' \qquad \text{where } 0 < v_4 < 3/2. \tag{9}$$

The situation described here with regard to value rates is commonplace: economic benefits are often most naturally associated with *output* quantities, but for the purpose of reckoning capacity consumption, one wants to express activity levels as *input* quantities like server hours; thus the value rate v_j per unit of activity is best viewed as a derived or computed quantity, not as a primitive data element.

Given the inequality $0 < v_4 < 3/2$ in (9), readers may verify that the unique optimal solution of our static planning problem (6)–(7) is

$$x^* = \left(1,\ \tfrac{8}{10},\ \tfrac{2}{10},\ 0, 0, 0\right)'. \tag{10}$$

Moreover, $Ax^* = q$, so Assumption 1 is satisfied. That is, the value-maximizing nominal processing plan x^* in (10) uses all available capacity. There *do* exist alternative plans that process all inputs without using all available capacity, but they generate less value. For example, readers may verify that $x = (\frac{6}{10},\ 0,\ 1,\ \frac{4}{15}, 0, 0)'$ is feasible, satisfying all the constraints in (7), and it gives server 1 a utilization rate of $\frac{13}{15}$. (This becomes the value-maximizing feasible solution if one takes $v_4 > \frac{3}{2}$.)

Let us return now to the general setting. Preparing the way for later developments, we denote by b the number of activities j such that $x_j^* > 0$, calling these *basic activities*, and we number activities so that the basic ones are $1, \dots, b$. As in [6], activities $b+1, \dots, n$ will be called nonbasic, and the matrices R and A will be partitioned as follows:

$$R = [H\,J] \quad \text{and} \quad A = [B\,N], \tag{11}$$

where H and B both have b columns. Thus H and B are the submatrices of R and A, respectively, that correspond to basic activities.

4. A balanced stochastic processing network.

In this section we describe a general stochastic processing network (SPN), or stochastic network model. It can be viewed as an enriched version of the deterministic fluid model that implicitly underlies our static planning problem (6)–(7); in particular, the SPN also has ℓ resources, m distinct materials, and n processing activities. Further discussion of this modeling framework can be found in the companion paper [7], which emphasizes connections with other model classes that one encounters in applied probability, economics and operations research.

In describing the SPN, we take as given matrices and vectors (R, A, λ, q, v) that satisfy the various assumptions imposed in Section 3. (As readers will see shortly, each of these model elements has essentially the same interpretation as before.) We enrich the previous model by associating stochastic variability with both the exogenous flows and the endogenous processing activities described in Section 3. Such variability causes surpluses and deficits of materials to develop over moderate time spans, which motivates the system manager to vary activity levels dynamically, depending on observed system status.

We shall generalize in certain ways the model formulation developed in Section 5 of [6], which was itself a generalization of the multiclass queuing network model developed in [4]. For the most part, notation agrees with that used in [6]. The caveats expressed in Section 5 of [6], concerning the stylized nature of that paper's model formulation, all apply equally well here. In particular, readers should recognize that more complex variations of the model described below would plausibly yield the same Brownian network (1)–(5) as their natural diffusion approximation.

Altering slightly the notation used in [4] and [6], we take as given a collection of mutually independent, m-dimensional *elemental flow processes* $E = \{E(t),\ t \geq 0\}$ and $F_j = \{F_j(t),\ t \geq 0\}$ for $j = 1, \dots, n$. The ith component of the vector $F_j(t)$ is denoted $F_{ij}(t)$. We interpret $E_i(t)$ as the cumulative *exogenous* input of material i up to time t, with a negative value indicating a net removal of material i by exogenous processes. For $i = 1, \dots, m$ and $j = 1, \dots, n$ we interpret $F_{ij}(t)$ as the cumulative amount of material i consumed by the first t units of activity j undertaken, with a negative value indicating net production rather than net consumption.

We denote by $Q_i(t)$ the inventory of material i (or *quantity* of material i) on hand at time t. Imagining that each material i is stored in a dedicated buffer, the m-dimensional process $Q = \{Q(t), t \geq 0\}$ will be called both an *inventory process* and a *buffer contents process* at various points in the text below. Depending on the application context, negative inventories may be allowed in order to represent net deficit conditions.

The static activity rates x_j encountered in Section 3 must now be generalized to dynamic control policies, or dynamic resource allocation policies, and following the practice established in [4, 6], we express such policies in terms of cumulative activity levels. That is, a dynamic control policy takes the form of a nondecreasing, n-dimensional stochastic process $T = \{T(t),\ t \geq 0\}$ with components $T_1, \dots, T_n$. Further requiring that $T(0) = 0$, we interpret $T_j(t)$ as the cumulative amount of activity j undertaken up to time t, so the m-dimensional inventory process Q corresponding to policy T is given by

$$Q(t) = Q(0) + E(t) - \sum_{j=1}^{n} F_j(T_j(t)), \qquad t \geq 0, \tag{12}$$

where $Q(0)$ is a given initial inventory vector, assumed deterministic for simplicity.

Given a nonnegative $\ell \times n$ capacity consumption matrix A as in Section 3, we require that the dynamic control policy T satisfy

$$A\bigl(T(t) - T(s)\bigr) \leq q(t - s) \qquad \text{for } 0 \leq s < t < \infty, \tag{13}$$

which means simply that the total amount of resource k capacity allocated during any time interval $(s, t]$ must be less than or equal to the amount that is available during that interval $(k = 1, \dots, l)$.

To give the vector λ and matrix R precise meaning in our stochastic network context, we define m-dimensional centered processes $\hat{E}, \hat{F}_1, \dots, \hat{F}_n$ by setting

$$\hat{E}_i(t) = E_i(t) - \lambda_i t \qquad \text{for } i = 1, \dots, m \text{ and } t \ge 0, \tag{14}$$

and

$$\hat{F}_{ij}(t) = F_{ij}(t) - R_{ij} t \qquad \text{for } i = 1, \dots, m,\ j = 1, \dots, n \text{ and } t \ge 0, \tag{15}$$

and then assume that $\hat{E}, \hat{F}_1, \dots, \hat{F}_n$ each satisfy a functional central limit theorem (FCLT); this is Assumption 2, which appears in Section 5. To repeat, the defining role of λ and R for our purposes is as centering constants in the aforementioned FCLTs. As in Section 3, we denote by x^* the optimal solution of the static planning problem (6)–(7), calling x^* our nominal processing plan.

Next, we take as given a large parameter $r > 0$ that serves to define performance relevant units of measurement, in a sense to be explained shortly. This is used to define an m-dimensional scaled buffer contents process Z via

$$Z(t) = r^{-1} Q(r^2 t), \qquad t \ge 0. \tag{16}$$

The last two elements of our stochastic processing network description are a compact and convex subset S of $\mathbb{R}^m$ with a nonempty interior and a continuous function h mapping S into $\mathbb{R}$. The *state space* S enters our formulation as follows: the dynamic control policy T must be chosen to ensure that (almost surely)

$$Z(t) \in S \qquad \text{for all } t \ge 0. \tag{17}$$

Of course, (17) is a restriction on the chosen control policy $T(\cdot)$; the activities available to the system manager are assumed to be such that controls satisfying (17) *do* exist. The scaled process Z expresses buffer contents as multiples of r, and it is implicit in (17) that these units of measurement are the relevant ones for purposes of describing system status. Also, the scaling of time embodied in (16) anticipates our eventual focus on time spans of order r^2 for purposes of evaluating system performance.

The *holding cost function* h and value rate vector v enter our model through the following definition: the *cumulative net value* realized by the system manager up to time t is

$$V(t) = v \cdot T(t) - r^{-1} \int_0^t h\big(r^{-1} Q(s)\big)\, ds, \qquad t \ge 0. \tag{18}$$

The first term on the right-hand side of (18) is the total value generated by processing activities up to time t (see Section 3), and the second term is a quantification of inventory holding costs (i.e., congestion-related costs) over the same time span. The factor of r^{-1} appearing in the argument of $h(\cdot)$ reflects again the notion that buffer contents are most naturally expressed as multiples of r, and the multiplicative factor of r^{-1} appearing outside the integral reflects an assumption that congestion-related costs are of lower order than the value derived from processing activities (except when $v = 0$). Readers will see that this order-of-magnitude separation in the model's two economic elements is crucial for our theory.

It is natural to focus on the difference between $V(t)$ and the maximum cumulative value $(v \cdot x^*)t$ that would be achievable in the deterministic model of Section 3. Accordingly, let

$$\hat{V}(t) = (v \cdot x^*)t - V(t), \qquad t \geq 0. \tag{19}$$

Assuming that time spans of order r^2 are the ones of interest, we shall hereafter express system performance by means of the scaled process

$$\xi(t) = r^{-1}\hat{V}(r^2 t), \qquad t \geq 0, \tag{20}$$

calling ξ the system manager's cumulative *cost process*. That name is potentially misleading, of course, because $\xi(t)$ includes both congestion-related costs and value degradation relative to a deterministic ideal, but it does communicate effectively the notion that smaller values of $\xi(\cdot)$ are desirable.

Proceeding as in Section 5 of [6], but with some small changes in notation, we define an n-dimensional process Y of scaled deviation controls via

$$Y(t) = r^{-1}\big(x^* r^2 t - T(r^2 t)\big), \qquad t \geq 0. \tag{21}$$

The basic idea behind this definition is to express the system manager's chosen activity levels $T_j(t)$ for the time interval $[0, t]$ as decrements from the nominal activity levels $x_j^* t$ for that same interval, but then we apply the same scaling of time and "space" that appears in (16).

Of course, T completely determines Y and vice versa, but Y is the more convenient representation of the system manager's control policy for our purposes. Now set

$$p = \ell + n - b, \tag{22}$$

and define a $p \times n$ matrix K via

$$K = \begin{bmatrix} B & N \\ 0 & -I \end{bmatrix}. \tag{23}$$

Comparing (23) with (11), one sees that the first ℓ rows of K are the capacity consumption matrix A, and the negative identity matrix $-I$ appearing in (23) has dimension $n - b$, which is the number of nonbasic activities in our static planning problem. Now let

$$U(t) = KY(t), \qquad t \geq 0. \tag{24}$$

Of course, a control policy T must meet certain restrictions if it is to be deemed "admissible." The next paragraph lists several admissibility requirements that are obvious in light of our network model's intended interpretation. Later, toward the end of Section 5, it will be argued that other "obvious" restrictions on control policies do not actually need to be expressed in a Brownian network approximation; for that reason, they will not be incorporated in the mathematical development below. Similarly, in describing the SPN to be approximated by a Brownian network, we make no attempt to explain in precise mathematical terms what is meant by a "nonanticipating" control policy. Instead, we leap directly to the obvious definition of nonanticipating controls when writing out the Brownian network's definition.

Our first admissibility restriction on a control policy T is that the p-dimensional process U derived from it by means of (21) and (24) satisfies the following requirement:

$$U(\cdot) \text{ is nondecreasing with } U(0) = 0. \tag{25}$$

To understand the content of (25), first recall that $Ax^* = q$ by Assumption 1. Combining this with (21), (11), (23) and (24), we see that the first ℓ components of the vector $U(t)$ equal

$$r^{-1}[Ax^* r^2 t - AT(r^2 t)] = r^{-1}[r^2 qt - AT(r^2 t)].$$

Thus the requirement that $U_1(\cdot), \dots, U_\ell(\cdot)$ be nondecreasing is equivalent to (13), and one has $U_1(0) = \cdots = U_\ell(0) = 0$ because $T(0) = 0$. Next, recall that $x_{b+1}^* = \cdots = x_n^* = 0$ (i.e., activities $b+1, \dots, n$ are nonbasic in our static planning problem), so the last $n - b$ components of $U(\cdot)$ are $r^{-1}T_{b+1}(r^2\cdot), \dots, r^{-1}T_n(r^2\cdot)$. Thus, the last $n - b$ components of (25) simply articulate the requirement that cumulative activity levels for nonbasic activities be nondecreasing.

Of course, (24) is identical to the definition (2) that appears in our specification of the Brownian network model. However, for reasons explained in the next section, the analog of (25) in our Brownian network model is (4), where the requirement $U(0) = 0$ is weakened to $U(0) \geq 0$. In the next section we also explain why the Brownian model need *not* have any element which expresses the requirement that cumulative activity levels for *basic* activities be nondecreasing.

Again proceeding as in Section 5 of [6], we express our basic system equation (12) in a form suggesting its Brownian analog (1). Given an admissible control policy T, let

$$X(t) = r^{-1}\left[\hat{E}(r^2 t) - \sum_{j=1}^{n} \hat{F}_j(T_j(r^2 t))\right], \qquad t \geq 0, \tag{26}$$

where the centered flow processes $\hat{E}$ and $\hat{F}_j$ are defined by (14) and (15). Let us now define the m-vector

$$z = r^{-1}Q(0). \tag{27}$$

Using the definition (21) of Y, and recalling that $Rx^* = \lambda$ (see Section 3), readers can verify that

$$r^{-1}\bigl[r^2\lambda t - RT(r^2t)\bigr] = RY(t), \tag{28}$$

and hence that (12) is equivalently expressed as

$$Z(t) = z + X(t) + RY(t), \qquad t \geq 0. \tag{29}$$

Also, by combining (16) with (18)–(21), readers can verify that

$$\xi(t) = \int_0^t h(Z(s))\,ds + v \cdot Y(t), \qquad t \geq 0. \tag{30}$$

This section has described a stochastic system model of conventional type, reusing notation that appeared earlier in Section 2 in order to establish correspondences between the conventional and Brownian models. That is, the processes X, Y, U and ξ appearing in the Brownian model (1)–(5) correspond to the processes denoted by those same letters in this section, and to form a Brownian approximation for a given stochastic processing network, the data R, K, v, z, S and h of the Brownian model are set equal to the elements of the conventional model denoted by those same letters in this section. What has *not* been explained thus far is how one determines the drift vector θ and covariance matrix Σ for the vector Brownian motion X appearing in (1). For that purpose, let us define

$$\tilde{X}(t) = r^{-1}\left[\hat{E}(r^2t) - \sum_{j=1}^{n} \hat{F}_j(x_j^* r^2 t)\right], \qquad t \geq 0, \tag{31}$$

observing that $\tilde{X}$ is identical to the process X defined above via (26), except that the actual activity levels $T_j(r^2t)$ appearing in (26) are replaced by the corresponding nominal activity levels $x_j^* r^2 t$ in (31). Readers will see shortly that, in the parameter regime where a Brownian approximation is appropriate, the only interesting control policies are those whose chosen activity levels deviate little from the nominal choices, in a certain sense; see (46) in the next section. The process X in (26) and (29) is then well approximated by $\tilde{X}$, and assuming that each of our elemental flow processes E, $F_1, \ldots, F_n$ satisfies a functional central limit theorem, it will be argued immediately below that $\tilde{X}$ is well approximated by a Brownian motion with drift vector $\theta = 0$ and the covariance matrix Σ specified in (35); these are the parameters to be used in our Brownian network approximation.

5. The driftless Brownian network as a formal limit. To support or motivate the Brownian approximation described immediately above, we introduce a thought experiment in which $r \to \infty$ but all other model elements remain fixed. In particular, the holding cost function h, the state space S, the scaled initial state z and the data (R, A, λ, q, v) for our static planning problem (6) and (7) do not depend on r. However, readers should bear in mind that the scale parameter r enters the economic structure of our model through (18). The balanced loading condition articulated as Assumption 1 in Section 3 remains in force.

Throughout this section, to indicate a process or quantity that depends on r we attach a superscript r to notation established earlier in Section 4. To justify the approximation of $\tilde{X}$ by a Brownian motion, it is assumed that each elemental flow process satisfies a functional central limit theorem (FCLT), as follows. Let

$$(32)\qquad \tilde{E}^r(t) = r^{-1}\hat{E}(r^2 t) \quad \text{and} \quad \tilde{F}_j^r(t) = r^{-1}\hat{F}_j(r^2 t) \qquad \text{for } j = 1, \dots, n.$$

ASSUMPTION 2. There exist covariance matrices $\Gamma_0, \Gamma_1, \dots, \Gamma_n$ such that, as $r \to \infty$,

$$(33)\qquad \tilde{E}^r \Rightarrow BM(0, \Gamma_0) \quad \text{and} \quad \tilde{F}_j^r \Rightarrow BM(0, \Gamma_j) \qquad \text{for each } j = 1, \dots, n.$$

Recall that $E, F_1, \dots, F_n$ were assumed to be mutually independent in Section 4. Moreover, from the definition (31) one has that

$$(34)\qquad \tilde{X}^r(t) = \tilde{E}^r(t) - \sum_{j=1}^{n} \tilde{F}_j^r(x_j^* t), \qquad t \geq 0.$$

Thus it follows directly from (33) that

$$(35)\qquad \tilde{X}^r \Rightarrow BM(0, \Sigma) \qquad \text{as } r \to \infty, \text{ where } \Sigma = \Gamma_0 + x_1^*\Gamma_1 + \cdots + x_n^*\Gamma_n.$$

Our formal argument to support the Brownian approximation described in Section 3 begins by considering system behavior under "fluid" scaling; we want to establish certain first-order conclusions about effective control strategies before addressing system behavior under Brownian scaling. Let there be given an admissible control strategy T^r for each $r > 0$, and then define a fluid-scaled version of T^r as follows:

$$(36)\qquad \tau^r(t) = r^{-2}T^r(r^2 t), \qquad t \geq 0.$$

To simplify argumentation, let us assume that there exists *some* process τ (necessarily nondecreasing and right-continuous) such that

$$(37)\qquad \tau^r \Rightarrow \tau \qquad \text{as } r \to \infty.$$

(This restriction to families of policies that are convergent under fluid scaling simplifies discussion substantially. A complete and rigorous limit theory to justify the proposed Brownian approximation would have to deal with nonconvergent families as well, but that and other potential complexities are simply ignored in the current treatment.) We now define a fluid-scaled version of the cumulative value process V^r as follows:

$$\nu^r(t) = r^{-2}V^r(r^2 t), \qquad t \geq 0.$$

From the definitions (18) and (16) of V and Z, respectively, one has that

$$\nu^r(t) = v \cdot \tau^r(t) - r^{-1} \int_0^t h\big(Z^r(s)\big)\, ds, \qquad t \geq 0. \tag{38}$$

The holding cost function h is bounded over the state space S of Z^r, so we have the following limit theorem for the fluid-scaled cumulative value process ν^r:

$$\nu^r \Rightarrow v \cdot \tau \qquad \text{as } r \to \infty. \tag{39}$$

We define fluid-scaled versions of the elemental flow processes E, $F_1, \dots, F_n$ as follows:

$$e^r(t) = r^{-2} E(r^2 t) \quad \text{and} \quad f_j^r(t) = r^{-2} F_j(r^2 t) \tag{40}$$

for $r > 0$, $t \geq 0$ and $j = 1, \dots, n$. The FCLTs assumed in (33) imply that $r^{-1}\tilde{E}^r \Rightarrow 0$ and $r^{-1}\tilde{F}_j^r \Rightarrow 0$ $(j = 1, \dots, n)$ as $r \to \infty$. Denoting by R_j the jth column of R, one can restate this conclusion as follows:

$$e^r \Rightarrow e \quad \text{and} \quad f_j^r \Rightarrow f_j \qquad \text{for } j = 1, \dots, n \tag{41}$$

as $r \to \infty$, where

$$e(t) = \lambda t \quad \text{and} \quad f_j(t) = R_j t \qquad \text{for } j = 1, \dots, n. \tag{42}$$

From the fundamental system equation (12) and the definitions (16) and (27), one has the following relationships among fluid-scaled processes for the rth system model,

$$r^{-1} Z^r(t) = r^{-1} z + e^r(t) - \sum_{j=1}^{n} f_j^r\big(\tau_j^r(t)\big). \tag{43}$$

Recall from (17) that any admissible control strategy for the rth model must keep Z^r within the bounded state space S. Neither S nor z depends on r, so the left-hand side of (43) and the first term on the right-hand side both converge weakly to the zero process as $r \to \infty$. Combining this with (37) and (41)–(43), then rearranging terms and recalling that R_j is by definition the jth column of our input–output matrix R, we use the continuity of e, $f_1, \dots, f_n$ and the random change of time theorem (see [1], page 151) to arrive at the following identity:

$$R\tau(t) = \lambda t \qquad \text{for all } t \geq 0. \tag{44}$$

The boundedness of our state space S is crucial for this identity, because it insures that $r^{-1}Z^r \Rightarrow 0$; see Section 9 for further discussion of the boundedness assumption. As a companion to (44), it follows directly from the capacity constraint (13) that

$$A\tau(t) \leq qt \qquad \text{for all } t \geq 0. \tag{45}$$

Moreover, $\tau(\cdot) \geq 0$, so for any fixed $t > 0$ the vector $x_t := t^{-1}\tau(t)$ satisfies all the constraints (7) of the static planning problem discussed in Section 3. In the case

$v = 0$, where Assumption 1 says that x^* is the *only* vector satisfying (7), this means that $x_t = x^*$ and hence $\tau(t) = x^* t$ for arbitrary $t > 0$; then by the right-continuity of τ one has $\tau(0) = 0$ as well. If $v \neq 0$, then Assumption 1 says that $v \cdot x^* \geq v \cdot x_t$, or equivalently, $v \cdot \tau(t) \leq (v \cdot x^*)t$, with equality holding only if $\tau(t) = x^* t$.

Thus we find that, among all processes τ which are achievable as weak limits in (37), there is one that uniquely maximizes (in a pathwise sense) the limiting fluid-scaled cumulative value process in (39), namely, $\tau^*(t) := x^* t$ for all $t \geq 0$. Accordingly, attention will hereafter be restricted to families of control policies T^r such that

$$\tau^r \Rightarrow \tau^* \qquad \text{as } r \to \infty. \tag{46}$$

Assuming that (46) holds, it follows from the FCLTs in (33) and the definition (26) of X that X^r converges weakly to the same limit as does $\tilde{X}^r$ in (35). That is, under any family of policies worthy of economic consideration, the process X appearing in our main system equation (29) converges in distribution as $r \to \infty$ to the corresponding process X in (1).

From this point onward, the heuristic justification of our proposed Brownian network approximation proceeds exactly as in [4] or [6]: in Section 4 we have seen that (1)–(5), which define the Brownian network model, simply repeat the corresponding relationships (29), (24), (17), (25) and (30) for our original stochastic processing network, except for one rather minor discrepancy.

The discrepancy is that (25) requires $U(0) = 0$, whereas the corresponding relationship (4) in the Brownian model imposes the weaker restriction $U(0) \geq 0$. As in Section 5 of [6], one can explain or defend this "relaxation" in the following way. Given the scaling that is used in the definition (21) of Y, components of the process U defined by (24) can increase at rate r in the original stochastic processing network, and as $r \to \infty$ that restriction becomes inconsequential. That is, as $r \to \infty$ the system manager is able to approximate ever more closely a nonzero initial value for Y, and hence also for U, at $t = 0$. However, if the system manager chooses to do that, the initial "impulse control" $Y(0)$ must be such that $U(0) = KY(0) \geq 0$.

In a similar fashion, there is no restriction in the Brownian model (1)–(5) corresponding to the requirement that cumulative activity levels $T_j(\cdot)$ be nondecreasing for the *basic* activities $j = 1, \ldots, b$ in our original stochastic processing network. For those j one has $x_j^* > 0$, so the corresponding deviation control Y_j obtained from a nondecreasing choice of T_j via (21) may either increase or decrease as a function of time. The requirement that T_j be nondecreasing corresponds to a bound of order r on the rate of decrease for Y_j, again because of the scaling used in (21), and that restriction on the rate of decrease becomes inconsequentially weak as $r \to \infty$.

Another issue deserving comment concerns our restriction to *adapted* controls Y in the Brownian network model (1)–(5). Of course, this is intended to capture the notion that control policies in the original stochastic processing network must be suitably nonanticipating, meaning that activity levels up to time t depend only on information available at t. That requirement was not given formal expression in Section 4, and as in [4] and [6], no attempt will be made here to justify the obvious way it has been represented in the Brownian model.

6. Relaxing the assumption of perfect balance.

In this section we generalize both the Brownian network approximation proposed in Section 4 and the formal limiting argument advanced in Section 5 to support it, extending the discussion to systems that satisfy Assumption 1 only in an approximate sense. This progression is similar to the standard treatment of heavy traffic theory for a single-server queueing system: denoting by ρ the system's traffic intensity parameter, most authors first assume that $\rho = 1$, and then make extensions to the case where ρ is near 1. Of course, the traffic intensity parameter of a single-server queue *need not* be near 1, and in similar fashion, the assumptions embedded in this section, which make precise the idea of approximate balance in a stochastic processing network, need not be staisfied by an arbitrary model.

The viewpoint adopted here is that we have at the outset a single stochastic processing network, referred to as the original model, and our central concern is to specify a Brownian network approximation for it. Only after a concrete approximation has been specified will we consider formal limits.

Formulating a reference model. Let there be given an original model, not necessarily satisfying Assumption 1 but having all the structure described in Section 4, with exogenous flow rate vector λ and input–output matrix R that enter as centering terms in (14) and (15), respectively, and with resource capacity vector q. To generalize the earlier assumption of perfect balance (Assumption 1 of Section 3), we consider a reference model that is identical to the original one in all regards except that its capacity vector is q^* rather than q. One typically has a certain amount of discretion in choosing or specifying q^* (i.e., the reference model is not unique), and we shall impose four assumptions or restrictions related to that choice. Readers will see that these restrictions concern the original model's parameter values and, in a small way, its structure as well. No method will be proposed here for systematically mapping a given original model into a suitable reference model, although it may be possible to develop such methods.

Our first assumption regarding the choice of a reference model is that all components of q^* are strictly positive. Second, q^* must be close to q (see below for elaboration). Third, q^* must be chosen so that the reference model satisfies Assumption 1 of Section 3. That is, if we substitute q^* for q, then the static planning problem (6)–(7) has a unique solution x^*, and moreover, $Ax^* = q^*$. Exactly as in Section 3, we number activities so that $x_1^*, \dots, x_b^*$ are strictly positive and $x_{b+1}^* = \cdots = x_n^* = 0$, and we partition R and A as in (11), so that H and B each have b columns. In particular, then, B is the $\ell \times b$ submatrix of A whose columns correspond to basic activities. Also, setting $p=\ell+n-b$, we define the $p\times n$ matrix K via (23) as before.

Our fourth assumption or requirement related to the choice of q^* is that the submatrix B has full row dimension ℓ (i.e., the rows of B are linearly independent). This very mild assumption is satisfied by virtually all model structures of practical interest, regardless of how q^* is chosen. In particular, as readers will see in Section 8, it is automatically satisfied by the familiar class of models where each activity consumes the capacity of exactly one resource. Given that B has full row dimension, let us denote by $B^\dagger$ a fixed right inverse (i.e., a $b\times\ell$ matrix satisfying $BB^\dagger=I$); it will be shown later in this section that *which* right inverse one chooses is unimportant, because two different choices of $B^\dagger$ lead to Brownian network approximations that are effectively equivalent. Now define $y\in\mathbb{R}^b$ by setting

$$y=B^\dagger(q-q^*), \tag{47}$$

extend y to an n-vector via

$$y_{b+1}=\cdots=y_n=0, \tag{48}$$

and then define a nominal processing plan x for the original model (as distinct from the nominal processing plan x^* for the reference model) by taking

$$x=x^*+y. \tag{49}$$

Of course, if $q-q^*$ is sufficiently close to 0, then (47)–(49) guarantee that $x-x^*$ is close to 0 as well. We require in particular that $q-q^*$ be small enough to ensure that $x_1,\dots,x_b$ are all strictly positive. From (47)–(49) and the definition of B we have

$$Ax=Ax^*+Ay=q^*+(q-q^*)=q. \tag{50}$$

That is, the vector x of average activity rates has been constructed so as to precisely consume all available capacity in our original model. A modified parallel-server example will be discussed at the end of Section 7 in order to illustrate the new ideas related to reference models that have been introduced in this section.

In choosing a reference model, one wants a balanced stochastic processing network (i.e., one whose first-order data satisfy Assumption 1) that is close to the original model, that has the same basic structure, and differs from the original model in a relatively simple way. With an eye toward the last two of those criteria, we have required here that the reference model differ from the original one only by small changes in the resource capacities $q_1,\dots,q_\ell$. In contrast, the reference models considered in Section 5 of [6] were allowed to differ from the original model only by changes in the vector λ of average exogenous flow rates. (Under this approach, the first-order data R, A, q and v are identical for the original and reference models, but the vector λ in the original model can be replaced by a nearby vector λ^* in the reference model.) Upon careful consideration, however, one sees that the latter approach to reference model formulation is unsatisfactory: to find a reference model that satisfies Assumption 1 and is close to the original one, without any change in the input–output matrix R or the resource capacity vector q, one may need to change the basic structure of the exogenous flows; for example, it may be necessary to create nonzero exogenous flows in the reference model where the original one has zero flow. Such distortions of the original model structure can be avoided in the framework proposed here: that is, having assumed all resource capacities q_k to be strictly positive, one can "perturb" any or all of them without changing the model's character, and having allowed such capacity perturbations, there is no need to allow changes in the exogenous flow rates λ_i as well.

The heavy traffic parameter regime. Proceeding exactly as in Section 4, we assume that there is given in conjunction with the original model a large parameter r that enters the model's state space constraints and economic structure via (16)–(18). Sharpening the requirement that q^* be "close to" q, we define $\gamma \in \mathbb{R}^\ell$ via

$$\gamma = r(q^* - q), \tag{51}$$

and then require that all components of γ have moderate absolute magnitude. For future purposes let

$$\theta = r(\lambda - Rx). \tag{52}$$

Recalling that $Rx^* = \lambda$ (because Assumption 1 is satisfied when q^* is substituted for q), and using (47)–(49) plus the definition (11) of H, readers can verify that

$$\theta = HB^\dagger\gamma. \tag{53}$$

In addition to the heavy traffic assumptions already articulated (r is large and $\gamma_1, \ldots, \gamma_\ell$ are moderate), we continue to impose Assumption 2 of Section 5; that is, the elemental flow processes $E, F_1, \ldots, F_n$ satisfy FCLTs with associated centering vectors $\lambda, R_1, \ldots, R_n$ and associated covariance matrices $\Gamma_0, \Gamma_1, \ldots, \Gamma_n$, respectively.

Brownian network approximation. The system of definitions and representations developed in Section 4 remains exactly as before up to the definition (21) of Y, in which we now substitute for x^* the vector x defined by (47)–(49). That is, in defining the scaled deviation control $Y(t)$ we express the chosen activity levels as deviations from nominal activity levels *for our original model.* Then, defining $U(\cdot)$ in terms of $Y(\cdot)$ by means of (24) as before, the restriction (25) on admissible controls is again appropriate because of (48) and (50). Also, readers can verify that the process X appearing in our basic system equation (29) is now given by

$$X(t) = r^{-1}\left\{\hat{E}(r^2 t) - \sum_{j=1}^{n} \hat{F}_j\bigl(T_j(r^2 t)\bigr)\right\} + \theta t, \tag{54}$$

where θ is defined via (52), or equivalently, via (53). That is, our earlier definition (26) of $X(t)$ is altered only by the addition of a "drift term" θt.

In Sections 4 and 5 it was argued that, in the heavy traffic parameter regime, one can effectively restrict attention to control policies T whose fluid-scaled version τ is well approximated by $\tau^*(t) := x^* t$, $t \geq 0$, for purposes of computing the distribution of X. That argument is essentially unchanged in the current context (see below), so exactly as before, we approximate the first term on the right-hand side of (54) by $BM(0, \Sigma)$, where

$$\Sigma = \Gamma_0 + x_1^* \Gamma_1 + \cdots + x_n^* \Gamma_n \tag{55}$$

as in (35). Equivalently stated, the process X is well approximated by $BM(\theta, \Sigma)$ under any policy worthy of consideration. Thus we arrive at an approximating Brownian network model that is exactly as described in Section 4 except for two factors: first, the Brownian motion X appearing in (1) has a nonzero drift vector θ given by formula (52) or (53), and second, the nominal processing plan x^*, which appears in formula (55) for the Brownian network's covariance matrix, is derived from a reference model whose specification involves some degree of discretion (i.e., x^* is not uniquely determined by data of the original model).

Invariance to choice of a right inverse for B. It is obvious from (53) that the drift vector θ for our Brownian network approximation depends on the choice of a right inverse $B^\dagger$. However, two different choices of the right inverse $B^\dagger$ give rise to equivalent Brownian network approximations, as the next three paragraphs explain. Roughly, the argument is as follows: the only role of the right inverse $B^\dagger$ is in determining the nominal processing plan x for our original model, and by changing the nominal processing plan, one simultaneously changes both the drift vector θ of the approximating Brownian network and the baseline relative to which the scaled deviation control Y is defined; when one considers the ultimate meaning or interpretation of the scaled deviation control, those two changes precisely cancel one another. Working through this argument provides a good review of the Brownian network's construction and the use for which it is intended.

Consider two different choices for the right inverse $B^\dagger$. These give rise to two different values y and $\check{y}$ in (47), and hence to two different choices $x = x^* + y$ and $\check{x} = x^* + \check{y}$ for the nominal processing plan in our original model. From (50) we have that $Ax = A\check{x} = q$, and components $b+1, \ldots, n$ of both x and $\check{x}$ are 0, so the definition (23) of K gives us

$$K(x - \check{x}) = 0. \tag{56}$$

Now consider the approximating Brownian network that results from our first choice x for the nominal processing plan, using the notation in Section 2 to describe it. In particular, the letter Y is used to denote a generic control, and according to (52) the drift vector of X is $\theta = r(\lambda - Rx)$. Now suppose we make the change of variable $\check{Y}(t) = Y(t) - r(x - \check{x})t$, so that $Y(t) = \check{Y}(t) - r(\check{x} - x)t$. Substituting that expression for $Y(t)$ into the main system equation (1) we obtain, after simplification,

$$Z(t) = \check{X}(t) + R\check{Y}(t) \qquad \text{for all } t \geq 0, \tag{57}$$

where $\check{X}(t) = X(t) + (\check{\theta} - \theta)t$ and $\check{\theta} = r(\lambda - R\check{x})$. Moreover, from (56), we have that $K\check{Y}(t) = KY(t)$, so the process U defined by (2) can equally well be written as

$$U(t) = K\check{Y}(t) \qquad \text{for all } t \geq 0. \tag{58}$$

Finally, let us define a new "cumulative cost process" $\check{\xi}$ by means of (5) with $\check{Y}$ in place of Y, observing that the difference between ξ and $\check{\xi}$ is a deterministic process and is therefore uncontrollable.

Thus substitution of the generic control $\check{Y}$ for Y has brought us to precisely the same approximating Brownian network that would have been obtained if we had chosen $\check{x}$ rather than x as our nominal processing plan: a control $Y(t)$ is optimal for the Brownian model based on x if and only if $\check{Y}(t) = Y(t) - r(x - \check{x})t$ is optimal for the Brownian model based on $\check{x}$. That is appropriate, of course, because values for $Y(t)$ represent scaled deviations from the nominal time allocations $T(t) = xt$, whereas values for $\check{Y}(t)$ represent scaled deviations from the nominal time allocations $\check{T}(t) = \check{x}t$.

The Brownian network as a formal limit. For purposes of limit theory it is natural to take as given a fixed reference model whose data satisfy Assumption 1 exactly, as in the development above, and to consider a parametric family of stochastic processing networks indexed by $r \to \infty$ and such that the data of these networks approach those of the reference model in a suitable sense. Accordingly, we fix the reference model described earlier in this section, carrying forward all of the assumptions and all of the notation laid out there. To generate a parametric family in a parsimonious fashion, but consistent with the approximation procedure specified above, we take as given a moderate vector $\gamma \in \mathbb{R}^{\ell}$ and then set

$$q^r = q^* - r^{-1}\gamma \qquad \text{for large } r > 0. \tag{59}$$

In the obvious way, we take q^r to be the resource capacity vector in the model with scale parameter r; in all other respects that "rth model" is identical to the reference model. Of course (59) simply reproduces the relationship (51) that appeared in conjunction with our approximation procedure, so one of the models generated by (59) is our original one. As in Section 5, a superscript r will be used to denote a quantity or process associated with the rth model in our parametric family. Recall from (53) that the drift parameter θ appearing in (54) satisfies $\theta = HB^{\dagger}\gamma$. Thus, because we view γ as fixed (i.e., independent of r), θ does not depend on r; it will therefore be written *without* a superscript r below.

As we let $r \to \infty$ in the parametric family of models just described, we conclude exactly as in Section 5 that only policies T^r satisfying

$$r^{-1}T^r(r^2\cdot) \Rightarrow \tau^*(\cdot) \qquad \text{as } r \to \infty \tag{60}$$

are of interest. As before, (60) together with Assumption 2 of Section 5 implies a FCLT for the process

$$X^r(t) = r^{-1}\left\{\hat{E}(r^2t) - \sum_{j=1}^{n} \hat{F}_j\big(T_j^r(r^2t)\big)\right\} + \theta t \tag{61}$$

that appears in the basic system equation (29) for our rth model. That is, $X^r \Rightarrow BM(\theta, \Sigma)$ as $r \to \infty$, where Σ is given by (55).

7. Parallel-server example.

Consider again the parallel-server model portrayed in Figure 1, where we have $m = 2$ job classes, $n = 6$ processing activities (actually, activities 5 and 6 might better be described as "nonprocessing activities") and $\ell = 2$ servers. The following discussion of that example builds on the first-order analysis presented in Section 3. In particular, the data required for our static planning problem are specified by (8) and (9), and so the nominal processing plan x^* is given by (10). Setting $r = 10$, let us suppose that the holding cost function has the form

$$h(z) = a_1z_1^2 + a_2z_2^2 \qquad \text{where } a_1, a_2 > 0. \tag{62}$$

Recall that in Section 3 we implicitly adopted \$100 as our monetary unit: that is, the value rates v_j in (9) were said to be in units of hundreds of dollars per server hour. Similarly, let $h(z)$ be expressed in hundreds of dollars per hour. Then (62) and (18) together say the following with respect to inventory holding cost: when components of the inventory vector z are expressed in tens of jobs, the associated holding cost rate is $10h(z)$ dollars per hour.

Elemental flow processes and covariance matrix Σ. With regard to stochastic structure, let us assume for simplicity that all interarrival and service time distributions are exponential and that the various arrival and service processes are mutually independent as well. With this assumption, the two components of the exogenous arrival process E are independent Poisson processes, while each of the two-dimensional primitive processes $F_1, \dots, F_4$ has one component that is a Poisson process and one component that is identically zero. Of course, the theory developed here would apply equally well with general interarrival and service time distributions, with arrivals of the two classes correlated, and with other complications as well, but the case considered here makes for particularly simple calculations. In particular, with the assumptions made here it follows from the FCLT for Poisson processes that Assumption 2 (see Section 5) is satisfied, and the asymptotic covariance matrix Σ in (35) is

$$\Sigma = \operatorname{diag}(2\lambda_1, 2\lambda_2). \tag{63}$$

To model the "disposal activities" numbered 5 and 6 in the parallel-server example, one can take F_5 and F_6 to be the following deterministic processes: $F_{15}(t) = F_{26}(t) = t$ and $F_{25}(t) = F_{16}(t) = 0$ for all $t \geq 0$. This specification is consistent with the last two columns of the input–output matrix R in (8). Because the last two columns of A in (8) contain only 0, the capacity constraint in (13) does not impose any limitation on how rapidly $T_5(\cdot)$ and $T_6(\cdot)$ can increase. In fact, those two components of the system manager's control policy $T(\cdot)$ can even have instantaneous (positive) jumps; such jumps correspond to instantaneous ejection of jobs, in whatever numbers the system manager may desire, precisely as intended.

Other data of the approximating Brownian network model. For concreteness we take $z = 0$ to be the initial state. Given that activities 4–6 are nonbasic in the static planning problem, the general definition (23) of K specializes in this example to

$$K = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

The nondecreasing process $U(\cdot)$ defined by (2) in the Brownian network approximation is five dimensional, and its components represent limits of (scaled versions of) the following five processes in our original model: $U_1(\cdot)$ corresponds to cumulative unused capacity for server 1; $U_2(\cdot)$ corresponds to cumulative unused capacity for server 2; $U_3(\cdot)$ corresponds to cumulative time devoted to the nonbasic activity 4 by server 1, and $U_4(\cdot)$ and $U_5(\cdot)$ correspond to the cumulative number of jobs ejected from buffers 1 and 2, respectively.

Finally, we take the bounded state space S for our normalized buffer contents process Z to be $S = [0, u] \times [0, u]$, where u is a large positive number.

A modified example. To make connection with the ideas developed in Section 6, suppose that the parallel-server model has capacity vector $q = (1, 1.1)'$ but all other model elements are as described above. (The meaning or intrepretation of this modified example will be discussed in the next section.) The modified example does *not* satisfy Assumption 1, but we know from Section 3 that by taking $q^* = (1, 1)'$ we obtain a balanced reference model, as that term was used in

Section 6, whose associated nominal processing plan x^* is given by (10). If we set $\gamma = (0, -1)'$ and define a parametric family of models with capacity vectors

$$q^r = q^* - r^{-1}\gamma = \begin{pmatrix} 1 \\ 1 + r^{-1} \end{pmatrix} \qquad \text{for } r > 0, \tag{64}$$

then our modified example is the member of this family with $r = 10$. For the reference model that we have chosen, there are $b = 3$ basic activities, and the submatrix B appearing in the partition (11) of our capacity consumption matrix A is

$$B = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,1 \end{bmatrix}.$$

To formulate a nominal processing plan x for the modified parallel-server example, using the general recipe (47)–(49), it will be convenient to take the following right-inverse for B:

$$B^{\dagger} = \begin{bmatrix} 1 & 0 \\ 0 & 0.80 \\ 0 & 0.20 \end{bmatrix}. \tag{65}$$

With this choice the first three components of x^* are given by $B^{\dagger}q^*$, and so the nominal processing plan x obtained from (47)–(49) is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = B^{\dagger}q = \begin{pmatrix} 1 \\ 0.88 \\ 0.22 \end{pmatrix} \quad \text{and} \quad x_4 = x_5 = x_6 = 0. \tag{66}$$

Thus, according to (52) and (53), the Brownian network approximation for our modified parallel-server example has drift vector

$$\theta = r(\lambda - Rx) = HB^{\dagger}\gamma = -\begin{pmatrix} 0.1 \\ 0.8 \end{pmatrix}, \tag{67}$$

and all other elements of the Brownian approximation are the same as described earlier.

8. A common special structure.

Our construction of a nominal processing plan x for the modified parallel-server example, based on the nominal plan x^* for its reference model, involved what might be termed a “canonical choice” of the right inverse $B^{\dagger}$. The method is broadly applicable, but not universally so. The essential special structure exploited there is that each column of the capacity consumption matrix A has at most one positive element; that is, each activity is conducted by a single resource or else consumes no capacity at all. Most of the stochastic processing networks one encounters in the published literature have this special structure, including the multiclass “queueing networks” that were the focus of study in [4].

For a stochastic processing network that has this structure and further satisfies Assumption 1 of Section 3, the number of basic activities b must be greater than or equal to the number of resources ℓ, because according to Assumption 1, the basic activity levels $x_1^*, \dots, x_b^*$ exhaust the capacity of all resources. It follows that the submatrix B in (11) has full row dimension, and by generalizing (65) in an obvious way one can choose a right inverse $B^\dagger$ so that the vector x defined by (47)–(49) is related to x^* as follows:

$$x_j = (q_k/q_k^*)x_j^* \qquad \text{when } A_{kj} > 0, \tag{68}$$

and $x_j = x_j^*$ for those j such that $A_{kj} = 0$ for all $k = 1, \dots, \ell$. That is, to derive a nominal processing plan x from x^*, given the special structure described above, the activity levels associated with each server k can be scaled up or scaled down by a common factor, that factor being chosen so that the capacity q_k is precisely consumed.

Given the special structure identified in this section, more can be said about what it means to have a family of models parameterized by the capacity vector q. Consider, for example, the modified parallel-server example with $q_2 = 1.1$. The effect of ten percent added capacity for server 2, relative to what one has in the reference model described in Section 3, is that the two activities conducted by server 2 are "speeded up" by a factor of 1.1, and the value rates associated with those activities are increased by a factor of 1.1 as well. That is, the outcomes achievable in our modified parallel-server example are exactly the same as if we kept $q_2 = 1$ but increased μ_2, μ_3, v_2 and v_3 each by a factor of 1.1. (Recall from Section 3 that the imputed average value *per service completed* by means of activity j is v_j/μ_j. In the transformation just described, these ratios are unchanged, and so one might say that the essential economic structure of the model is unchanged.)

More generally, consider any stochastic processing network that has the special structure described at the beginning of this section. If the capacity q_k for any resource k is replaced by αq_k, where $\alpha > 0$, the effect is exactly the same as if q_k were left unchanged but $F_j(\cdot)$ were replaced by $F_j(\alpha\cdot)$ and v_j were replaced by αv_j for every activity j such that $A_{kj} > 0$.

9. Comments on the generalized Brownian network model. Having explained earlier how the data of the Brownian network model (1)–(5) are chosen to approximate a given stochastic processing network, we now consider various properties of the Brownian approximation. Recall from Section 6 that the data (R, A, v, λ) of the original model, along with the capacity vector q^* of an associated "reference model," jointly satisfy Assumption 1 of Section 3. That is, when q^* is substituted for q in the original model, the static planning problem (6)–(7) has a unique optimal solution x^*, and moreover $Ax^* = q^*$. To form the Brownian network model, we need to know b, the number of basic activities in x^*, and activities need to be numbered so that the basic ones are $1, \dots, b$. We then set $p = \ell + n - b$, partition A as in (11), and define K via (23). For future reference, we deduce the optimal solution of the following linear program from Assumption 1: choose $y \in \mathbb{R}^n$ to

$$\text{minimize } v \cdot y \qquad \text{subject to } Ry = 0 \text{ and } Ky \geq 0. \tag{69}$$

PROPOSITION 1. *The unique optimal solution of* (69) *is* $y^* = 0$.

PROOF. The proposed solution $y^* = 0$ is obviously feasible, with objective value $v \cdot y^* = 0$. Now suppose there exists some other $y \in \mathbb{R}^n$ such that $Ry = 0$, $Ky \geq 0$, and $v \cdot y \leq 0$. From the definition (23) of K one has that $y_{b+1}, \dots, y_n \leq 0$. For sufficiently small $\varepsilon > 0$, the vector $x = x^* - \varepsilon y$ would then satisfy $Rx = \lambda$, $Ax \leq q^*$, $x \geq 0$, and $v \cdot x \geq v \cdot x^*$. This contradicts Assumption 1. □

The justification provided in this paper for the generalized Brownian network model is admittedly incomplete. To build confidence in its validity as an approximation, one wants to check that the model passes certain tests of internal consistency. Given that the Brownian model formulation allows controls with unbounded variation, the following question naturally suggests itself. Can the system manager drive the cumulative "cost" process $\xi(\cdot)$ arbitrarily far in the negative direction, in any given time interval, by means of some admissible strategy? Loosely adopting a term from financial theory, one might say in this case that the Brownian network admits "arbitrage opportunities," and if such possibilities do not exist, one might say that the Brownian network is "arbitrage free."

To see the connection between Proposition 1 and arbitrage, suppose that there exists a vector $y \in \mathbb{R}^n$ such that

$$Ry = 0, \qquad Ky \geq 0, \quad \text{and} \quad v \cdot y < 0. \tag{70}$$

Now consider a control Y that has jump $Y(t) - Y(t-) = y$ at some fixed time $t > 0$. From (1) and the equality $Ry = 0$ in (70) we have that $Z(t) - Z(t-) = R[Y(t) - Y(t-)] = 0$, which means that the jump in Y does not violate the state space constraint (3). Also, (2) and the inequality $Ky \geq 0$ in (70) ensure that $U(t) - U(t-) \geq 0$, so the admissibility requirement (4) is not violated by the hypothesized jump in Y. Finally, the inequality $v \cdot y < 0$ in (70) says that the control jump generates a strictly negative jump in cumulative "cost," so by considering arbitrarily large multiples of y as control jumps we generate an arbitrage opportunity. Furthermore, one can extend this reasoning to show that arbitrage opportunities exist if *and only if* there exists a vector y satisfying (70). Of course, Proposition 1 says that no such y exists (the boundedness of the state space S is essential in this regard), so it ensures that the Brownian network is "arbitrage free."

A next concern about the Brownian network model is whether the system manager actually *can* keep Z within a bounded state space S, or equivalently, whether there exist *any* admissible controls Y. To address this issue let

$$\Delta = \{\delta \in \mathbb{R}^m : \delta = Ry,\ y \in \mathbb{R}^n,\ Ky \geq 0\}. \tag{71}$$

Elements of Δ are displacements of the state vector Z that the system manager can effect by means of control increments $y = \Delta Y$ that are feasible in the sense of (4). Thus Δ might be called the set of "feasible displacements." Assuming that the covariance matrix Σ is nondegenerate, it is more or less obvious that admissible controls exist if and only if $\Delta = \mathbb{R}^m$ but one would like to reduce that condition to concrete requirements on the matrices R and K.

The assumptions made in this paper do *not* guarantee the existence of admissible controls, because we have not imposed restrictions on the original stochastic processing network which ensure that our bounded state space constraint can be met in that context. (Presumably, such restrictions would ensure the existence of admissible controls when carried over to the corresponding Brownian model.)

This is a good point to comment about the new issues that arise if one allows the state space S of the Brownian network model to be unbounded. To put the discussion on a concrete footing, consider the parallel-server example described in Sections 3 and 7, specialized by taking $v_4 = 5/4$ (this is within the range $0 < v_4 < 3/2$ specified earlier) and *modified* by taking $\mu_4 = 1$ rather than $\mu_4 = 3$. This change makes activity 4 less attractive, so activity 4 remains nonbasic in the static planning problem (6) and (7). The approximating Brownian network then has

$$R = \begin{bmatrix} 1 & 0 & \frac{1}{2} & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}, \qquad K = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad \text{and} \quad v = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \frac{5}{4} \\ 0 \\ 0 \end{bmatrix}. \tag{72}$$

As a control increment $y = \Delta Y$, consider specifically

$$y = (1, 1, -1, -1, 0, 0)'. \tag{73}$$

Given the definition (21) of the scaled deviation control Y, one has the following interpretation of (73): over an unspecified time interval, server 1 undertakes r more units of activity 4 and r fewer units of activity 1 than called for in the nominal plan, while server 2 undertakes r more units of activity 3 and r fewer units of activity 2. The net effects of this control increment on the processes Z, U, and ξ are

$$Ry = \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}, \qquad Ky = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad v \cdot y = -\tfrac{1}{4}, \tag{74}$$

respectively. That is, the scaled buffer contents process Z_1 increases by one-half unit, the content of buffer 2 is unchanged, neither server experiences any idleness, the cumulative use of the nonbasic activity 4 increases, neither of the two disposal activities are used and the system manager decreases cumulative "cost," relative to what would have happened if the nominal plan were implemented over that same time interval. The economic gain derives from the fact that activity 4 earns higher value per unit of server capacity allocated to it than does activity 1 (activities 2 and 3 have identical value rates).

If we take the state space S of the Brownian network to be all of $\mathbb{R}^2_+$, then according to (74), the system manager can drive ξ arbitrarily far in the negative direction by implementing a control increment that is an arbitrarily large multiple of y, without violating the state space constraint (3). The *disadvantage* to such an action is that it causes a correspondingly large increase in Z, and hence in the holding cost rate $h(Z)$, but one sees that great care must be taken in formulating mathematically the system manager's problem. If, for example, the stated objective were to minimize $\mathbb{E}[\xi(T)]$ for some fixed T, without any penalty associated with ending inventory $Z(T)$, then the problem would be unbounded, because the system manager can implement an arbitrarily large multiple of y as the control increment over an arbitrarily short time span $[T-\varepsilon, T]$. This new and subtle danger in problem formulation arises because of the expanded economic framework employed in this paper, where individual activities may generate value in a linear fashion.

If one allows the state space S to be unbounded, then exactly the same issues come up in conjunction with the limiting argument described in Section 5. Specifically, to prove that policies violating (46) are dominated by policies satisfying (46), in the limit as $r \to \infty$, one must make assumptions about the holding cost function h, and one must consider the precise economic objective to be optimized. In fact, it seems likely that policies satisfying (46) are dominant *if and only if* there exists an optimal policy with a finite objective value in the Brownian control problem (1)–(5) that is mechanically derived from the model data.

Thus far nothing has been said about the "initial state" z appearing in (1). It is natural to think in terms of the case where $z \in S$, but this is not really necessary. If z is *not* in S, then an instantaneous displacement $\delta \in \Delta$ must be effected at $t = 0$, moving the system state from z to $z + \delta \in S$, by choosing a nonzero initial value $Y(0) = y$ for the cumulative control process. Of course, the system manager may choose to enforce such an instantaneous displacement at $t = 0$ even when $z \in S$. In such circumstances it is deceptive to describe z as the "initial state" of the Brownian network model. One might say instead that z represents an "initial condition" that the system manager may or may not convert to a different initial state $Z(0) = z + Ry$ by taking $Y(0) = y$.

Finally, let us consider the topic of "state space collapse," or "equivalent workload formulation," for a Brownian control problem whose cumulative cost process ξ can have the generic form (5). The approach to such model reduction developed in [8] centers on the concept of "reversible displacements," which are defined exactly as in (71) but with $Ky = 0$ in place of $Ky \geq 0$. The ultimate conclusion of the theory developed in [8] is that under an optimal control policy, the process Z lives in a manifold of dimension $d \leq m$. (In applications to multiclass queuing networks, the reduced dimension d is typically much smaller than m.) Points on the manifold correspond to different values of the "workload process" $W(t) = MZ(t)$, where M is a certain $d \times m$ matrix derived from R and K. Under model assumptions imposed in [8], a system manager is indifferent between any two states z and z' that have the same work content (meaning that $Mz = Mz'$), because either of the two states can be instantly and costlessly exchanged for the other by means of a reversible displacement. With the more general cost structure hypothesized in (5), it will be shown in future work that the generalized Brownian network still has dimension d, with the same workload process $W = MZ$ serving to summarize system status, but the process Z lives on a d-dimensional manifold that may differ from the one described in [8].

Acknowledgments. I am indebted to Bariş Ata, Hong Chen and Assaf Zeevi for their helpful comments on an earlier version of this paper, to Linda Bethel for her technical assistance in preparing the manuscript, and to Ruth Williams for her careful reading and insightful, constructive criticism at every stage of revision. The paper has been greatly improved by their contributions.

REFERENCES

[1] BILLINGSLEY, P. (1999). *Convergence of Probability Measures*, 2nd ed. Wiley, New York.

[2] BRAMSON, M. and WILLIAMS, R. J. (2000). On dynamic scheduling of stochastic networks in heavy traffic and some new results for the workload process. In *Proceedings of the 39th IEEE Conference on Decision and Control*. IEEE, New York.

[3] BRAMSON, M. and WILLIAMS, R. J. (2002). Two workload properties for Brownian networks. Unpublished manuscript.

[4] 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.

[5] HARRISON, J. M. (1998). Heavy traffic analysis of a system with parallel servers: Asymptotic analysis of discrete-review policies. *Ann. Appl. Probab.* **8** 822–848.

[6] HARRISON, J. M. (2000). Brownian models of open processing networks: Canonical representation of workload. *Ann. Appl. Probab.* **10** 75–103.

[7] HARRISON, J. M. (2003). Stochastic networks and activity analysis. In *Analytic Methods in Applied Probability. In Memory of Fridrich Karpelevich* (Y. Suhov, ed.). Amer. Math. Soc., Providence, RI.

[8] HARRISON, J. M. and VAN MIEGHEM, J. A. (1997). Dynamic control of Brownian networks: State space collapse and equivalent workload formulations. *Ann. Appl. Probab.* **7** 747–771.

[9] KARATZAS, I. and SHREVE, S. (1991). *Brownian Motion and Stochastic Calculus*, 2nd ed. Springer, New York.

[10] WILLIAMS, R. J. (2000). On dynamic scheduling of stochastic networks in heavy traffic. Presentation to Workshop on Stochastic Networks, Univ. Wisconsin, Madison.

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