# Two Topics in Multistage Manufacturing Systems

by

Philippe B. Chevalier

Ingénieur Civil en Mathématiques Appliquées
Université Catholique de Louvain (1988)

Submitted to the Sloan School of Management
in Partial Fulfillment of
the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
IN OPERATIONS RESEARCH

at the

Massachusetts Institute of Technology

June 1992

© Massachusetts Institute of Technology 1992

Signature of Author__________

Sloan School of Management
May 1992

Certified by__________

Lawrence M. Wein
Associate Professor of Management Science
Thesis Supervisor

Accepted by__________

Richard C. Larson
Co-Director, Operations Research Center

ARCHIVES
MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

MAY 18 1992

## Abstract

This thesis includes two related but independent subjects. In Part I we consider the problem of finding an optimal dynamic priority sequencing policy to maximize the mean throughput rate in a multistation, multiclass closed queueing network with general service time distributions and a general routing structure. Under balanced heavy loading conditions, this scheduling problem can be approximated by a control problem involving Brownian motion. Although a unique, closed form solution to the Brownian control problem is not derived, an analysis of the problem leads to an effective static sequencing policy, and to an approximate means of comparing the relative performance of arbitrary static policies. Several examples are provided that illustrate the effectiveness of our procedure.

In part II we consider two interrelated decisions concerning inspection in a circuit board assembly plant. The inspection allocation policy determines at which stage(s) to inspect a board. At stages where inspection is performed, the testing policy decides how to accept or reject a board based on the test measurements that are subject to some noise. The objective is to minimize the total expected cost of quality, which includes the inspection cost and the cost of defectives shipped to the customer. A case study is presented that reveals various implementation and data gathering issues. This study shows that our proposed policies offer significant cost savings over current industrial practice.

## Aknowledgements

I dedicate this thesis to my wife, Claire. W for embarking with me in this adventure in the new continent. She certainly could be credited for at least half of my work for she was such a source of inspiration.

I want to thank my advisor, Lawrence Wein, from whom I learned so much about Brownian Motions and many other less esoteric subjects. I also want to thank the members of my committee for their suggestions and enthousiasm for this project.

The second part of this thesis would certainly not have been possible without the enthousiasm and time from Bert Anderson, Steve Kalenik, Chuck Nordingen, Tom McKearn, Joan Hawkes and Rick Schul form Hewlett Packard. I really want to thank them very much for having provided me not only with a thesis subject, but with megabytes of data to try to apply my ideas.

The Leaders for Manufacturing program at MIT is also aknowledged for their financial support.

There is research, and then there is 'doing research'. Surely, nothing can beat the Operations Research Center for 'doing research'. So many friendships were made while 'doing research', it will always be a pleasure remembering the times spent at the OR Center, thank you to all who gave me their friendship.

Finally, I cannot forget Lloyd Cole and the Commotions, Steely Dan, Paolo Conte, Enya, Stan Getz, Mozart, Vivaldi, and all the others who made it bearable to spend hours writing and rewriting this thesis.

# Table of Contents

## Part I : Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Closed Network

## Part II : Inspection for Circuit Board Assembly

# Part I : Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Closed Network

## 1. Introduction

Multiclass closed queueing networks are important models for computer, communication, and manufacturing systems, and the descriptive theory of these networks is well developed (see Baskett et al. 1975 and Kelly 1979). However, no exact results exist for optimal priority sequencing in such systems, and the only approximate analysis is Harrison and Wein (1990), who obtain an effective priority sequencing policy for maximizing the throughput of a two-station, well balanced, heavily loaded network. This policy, called a *workload balancing* sequencing policy, is a static (that is, not state-dependent) policy that outperformed conventional sequencing policies in a simulation study in Harrison and Wein (1990). This result was obtained by analyzing a Brownian system model (developed by Harrison 1988) that approximates a multiclass queueing network with dynamic scheduling capability. Under balanced heavy loading conditions, this model allows a queueing network scheduling problem to be approximated by a control problem involving Brownian motion. The workload balancing sequencing policy was derived by reformulating the Brownian control problem in terms of workload imbalances, solving the workload imbalance formulation, and interpreting the solution in terms of the original queueing system.

In this paper, we attempt to generalize the results of Harrison and Wein (1990) from the setting of a two-station network to a network with any finite number of stations. In order to describe our results, it is easiest to first review the results of

Harrison and Wein (1990). They define a one-dimensional *workload imbalance* process, which measures how imbalanced the total network workload is between the two stations at each point in time, and discover an intricate relationship between workload imbalance, server idleness, and the lowest priority customer classes. In particular, in the idealized Brownian limit, server idleness is only incurred at times when the workload imbalance process is on the boundary of a *workload imbalance polytope*, which is a closed interval on the real line, and the two extreme points of the polytope correspond to the two customer classes, one from each station, that are awarded lowest priority at their respective stations. These two bottom priority classes, which are referred to as *extremal classes*, lead directly to the workload balancing sequencing policy. Furthermore, this relationship allows for an analytic comparison between the workload balancing policy and any other static policy, such as the shortest expected processing time rule (SEPT), where priority is given to the class with the shortest expected processing time for its upcoming operation, and the shortest expected remaining processing time rule (SERPT), where priority is given to the class with the least expected amount of work remaining before exiting the network.

For the general multistation problem considered here, the Brownian control problem can again be reformulated in terms of workload imbalances, but a unique, closed form solution to the workload imbalance formulation is not obtained. However, the corresponding relationship between workload imbalance, server idleness, and the lowest priority classes is retained in the multistation setting. In particular, when there are $I$ stations in the network, an $(I-1)$−dimensional workload imbalance process is defined that stays in a workload imbalance polytope in $R^{I-1}$. Also, server idleness is incurred only when the workload imbalance process is at the boundary of the workload imbalance polytope. Each extreme point of the polytope corresponds to a particular customer class, and these extremal classes are the only classes in the network that are ever given bottom priority at their respective stations. Unlike the two-station case, there will in general be more extremal classes than stations.

The insight gained from the previous paragraph allows us to identify an effective static sequencing policy for maximizing the throughput of a multistation, multiclass closed queueing network under balanced heavy loading conditions, and to approximately compare the performance of this policy to conventional static policies, such as the SEPT and SERPT rules. A simulation study analyzing five examples (two three-station networks and three four-station networks) is carried out that demonstrates the power of the simple procedure of identifying the workload imbalance polytope and the corresponding extremal classes. In particular, for each example, the proposed policy outperforms (at times, dramatically) four conventional policies, and the analysis roughly predicts the relative performance of the proposed policy, the SEPT rule, and the SERPT rule. Also, system performance is greatly influenced by the particular static policy in use.

Perhaps the most interesting conclusion of our study is the effectiveness of *static* policies for maximizing the throughput in multistation closed queueing networks. In contrast, when analyzing perhaps the simplest interesting open queueing network scheduling problem, Harrison and Wein (1989) found that no static policy was effective, and a dynamic (that is, state-dependent) policy was required to offer significant improvement over the first-come first-served (FCFS) policy. We believe this is due to the fundamental tradeoff that exists in all open queueing networks. This tradeoff is between the short run aim of reducing the number of customers in the system, and the longer run aim of avoiding server idleness. On the other hand, in a single-station queue, no such tradeoff exists, and the only concern is with reducing the number of customers in the system. Therefore, it is not surprising that a simple static policy (the so-called $c\mu$−rule; see Klimov 1974, for example) is able to achieve this goal.

Similarly, no tradeoff exists in a closed network setting, where server utilization is the sole concern, and so a static policy again appears to be effective, although obviously not optimal. In summary, it appears that the basic tradeoff that exists in sequencing open networks makes these systems more difficult to analyze and to sequence than closed networks or single-station systems.

The balanced heavy loading conditions imply that any stations in the original network that are not among the most heavily loaded will vanish in our idealized Brownian model. Thus, the proposed sequencing policy can be applied to any closed queueing network by restricting attention to the subnetwork of bottleneck stations. Although our procedure works very well on the bottleneck subnetwork and appears to be quite robust with respect to the magnitude and balance of the network load (see, in particular, example 5 of Section 7), further study is required to assess the effectiveness of this procedure for scheduling an entire network consisting of bottleneck and nonbottleneck stations. However, the bottleneck stations are precisely where most of the congestion occurs, and where scheduling will have its biggest impact. Thus, we believe these results have the potential to enhance system performance in actual closed network settings.

This paper is organized as follows. In Section 2, the queueing network scheduling problem is described, and the workload imbalance formulation of the approximating Brownian control problem is given in Section 3. The workload imbalance polytope is defined in Section 4, where the relationship between server idleness, workload imbalance, and extremal classes is described. A static sequencing policy is proposed in Section 5, which also contains an approximate analytic comparison between the proposed policy and any other static policy. An alternative workload imbalance formulation is described in Section 6, which can be omitted by readers interested primarily in the nonmathematical aspects of the paper. Five examples are contained in Section

7, along with simulation results.

## 2. The Queueing Network Scheduling Problem

Consider a queueing network consisting of $I$ single server stations, and populated by a variety of different customer *types*, where each type has its own arbitrary, deterministic route through the network. As in Kelly (1979) and Harrison (1988), we define a different customer *class* for each stage along each customer type's route. Each customer class $k = 1, ..., K$ requires service at a particular station, and has its own general service time distribution with finite mean and variance. Thus, individual customers change class deterministically as they proceed through the network.

Whenever a customer completes the last stage of its route, it exits the network, and a new customer immediately enters, so as to keep the population size fixed at $N$ customers. The entering customer will be of class $k$ with probability $q_k$, independent of all previous history. Of course, $q_k > 0$ only for classes that correspond to the first stage along some customer type's route.

Notice that this is a *single chain* network, where the entering mix of the various customer types is fixed, as opposed to a *multichain* network where the population level of the various customer types is fixed. The single chain network is appropriate for a manufacturing setting, which is our primary interest. In a job shop, the product mix is typically specified by customer demand, and the most direct way to satisfy this mix in a closed network setting is to release new customers according to the appropriate entering class mix $q = (q_k)$. Our analysis allows the class of entering customer to be chosen in a deterministic (rather than Markovian) fashion according to the vector $q$, in which case the proposed policy remains unchanged. Also, customer routes are assumed to be deterministic for ease of presentation; probabilistic events that occur in a manufacturing setting, such as rework, scrapping, and server breakdown and repair, can be easily incorporated into the model (see Harrison 1988 for details).

The scheduling problem is to dynamically decide which class of customers to serve next at every station in the network. These decisions will be referred to as *sequencing* decisions. The objective of the scheduling problem is to maximize the long run expected average throughput rate of the network, which is the number of customer departures per unit of time. Since the customer population level is fixed, Little's formula (Little 1961) implies that this objective will also minimize the long run expected average sojourn time of customers, which is the amount of time a customer spends in the network. Since the entering class mix, customer routes, and mean service times are all fixed, maximizing the long run expected average throughput rate is equivalent to minimizing the long run expected average idleness rate for any arbitrary server, which is the fraction of time the server is idle.

## 3. The Workload Imbalance Formulation

Harrison (1988) has shown how to approximate the closed queueing network scheduling problem described in Section 2 by a Brownian control problem. In Section 2 of Harrison and Wein (1990), an equivalent formulation of this problem is derived for the two-station case. The new formulation is called a *workload formulation* because the state of the queueing system is described in terms of a two-dimensional workload process, rather than the $K-$dimensional queue length process. In Section 3 of Harrison and Wein (1990), the workload formulation is easily re-expressed in terms of a *workload imbalance* formulation (see equations (38)-(44) of that paper), where the state of the network is a one-dimensional workload imbalance process, which measures how imbalanced the workload of the first station is relative to the second station.

Readers are also referred to Wein (1991) for a similar multidimensional workload imbalance formulation. We will go directly to the workload imbalance formulation of the problem in order to avoid much unnecessary notation. As in Harrison and Wein (1990), the proposed sequencing policy depends only on the solution to the workload imbalance formulation.

Let $Q_k(t)$ be the number of class $k$ customers in the network at time $t$, and let $I_i(t)$ be the cumulative idleness incurred by server $i$ in the time interval $[0, t]$. The Brownian approximation is obtained by rescaling these two basic processes in terms of the total population size $N$. In particular, define the scaled *queue length process* $Z_k = \{Z_k(t), t \geq 0\}$ by

$$Z_k(t) = \frac{Q_k(N^2t)}{N}, \; t \geq 0 \;\text{ and }\; k = 1, ..., K, \tag{1}$$

and the scaled *cumulative idleness process* $U_i = \{U_i(t), t \geq 0\}$ by

$$U_i(t) = \frac{I_i(N^2t)}{N}, \; t \geq 0 \;\text{ and }\; i = 1, ..., I. \tag{2}$$

Notice that $Z_k(t)$ is interpreted as the fraction of customers in the network at time $t$ who are of class $k$. The vector processes $Z = (Z_k)$ and $U = (U_i)$ are the control processes in the workload imbalance formulation of the Brownian control problem, and these scaled processes will be referred to simply as the queue length and cumulative idleness processes, respectively. Since we will be dealing exclusively with the Brownian model in the next two sections, this should cause no confusion.

Let us define $M_{ik}$ to be the expected remaining processing time at station $i$ for a class $k$ customer until that customer exits the network. The $I \times K$ *workload profile matrix* $M = (M_{ik})$ depends on the mean processing time of each customer class and the detailed route of each customer type. Readers may refer to Table I and equation (24) in Section 7, where the entries of this matrix are displayed for a concrete example.

As mentioned in Section 2, newly injected customers are of class $k$ with probability $q_k$. For $i = 1, ..., I$, define $v_i = \sum_{k=1}^{K} M_{ik} q_k$, so that $v_i$ is the expected total time over the long run that server $i$ devotes to each newly arriving customer. Recall that in closed queueing networks, the vector of traffic intensities can only be determined up to a scale constant. As in Harrison and Wein (1990), the relative traffic intensitites $\rho = (\rho_i)$ will be scaled so that $\max_{\{1 \le i \le I\}} \rho_i = 1$. By Proposition 2 of Harrison and Wein (1990), it follows that $\rho_i = v_i / \max_{\{1 \le j \le I\}} v_j$, for $i = 1, ..., I$. The *balanced heavy loading conditions* for the closed network assume the existence of a sufficiently large integer $N$ such that the total population size is $N$ and $N|1 - \rho_i|$ is of moderate size for all $i = 1, ..., I$.

Define the $(I-1) \times K$ *workload imbalance profile matrix* $\hat{M} = (\hat{M}_{ik})$ by

$$\hat{M}_{ik} = \rho_I M_{ik} - \rho_i M_{Ik} \quad \text{for } i = 1, ..., I-1 \text{ and } k = 1, ..., K. \tag{3}$$

As in Harrison and Wein (1990), and Wein (1990b,1991), this matrix contains all the necessary information about each customer class to schedule the network under balanced heavy loading conditions.

Let $X$ be a $K-$dimensional Brownian motion process with drift vector $\delta$ and covariance matrix $\Sigma$, which are defined in equations (13)-(14) of Harrison and Wein (1990) in terms of the first and second moments of the service time distributions of the different customer classes, the routes of the various customer types, and the entering class mix. Also, let $B = (B_i)$ be defined by $B = TMX$, where the $(I-1) \times I$ matrix $T$ is given by

$$T = \begin{pmatrix} \rho_I & 0 & 0 & . & . & 0 & -\rho_1 \\ 0 & \rho_I & 0 & . & . & . & -\rho_2 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & \rho_I & 0 & -\rho_{I-2} \\ 0 & 0 & 0 & . & 0 & \rho_I & -\rho_{I-1} \end{pmatrix}, \tag{4}$$

so that $B$ is an $(I-1)$–dimensional Brownian motion process with drift $\mu = TM\delta$ and covariance $\Gamma = TM\Sigma M^T T^T$. Our proposed policy does not depend on the parameter values of the two Brownian motion processes (as mentioned earlier, the policy depends only on the matrix $\hat{M}$ in (3)), and hence applies equally well if the class of entering customer is chosen in a deterministic or Markovian manner, since the Brownian models arising from these two release mechanisms differ only in their covariance matrix. It is worth noting that the components of the drift vector $\mu$ are $\mu_i = N(\rho_i - \rho_I)$ for $i = 1,..,I-1$, by Proposition 3 of Harrison and Wein (1990), and thus the Brownian motion is driftless when the queueing network is perfectly balanced.

The approximating Brownian control problem is obtained by letting the customer population size $N \to \infty$. By Propositions 2 and 7 of Harrison and Wein (1990), the workload imbalance formulation of the Brownian control problem is to choose a $K$–dimensional process $Z$ and an $I$–dimensional process $U$, both of which are RCLL (right continuous with left limits), to

$$\text{minimize} \ \limsup_{t\to\infty} \frac{1}{t} E[U_1(t)] \tag{5}$$

$$\text{subject to} \ \ Z \text{ and } U \text{ are nonanticipating with respect to } X, \tag{6}$$

$$\sum_{k=1}^{K} \hat{M}_{ik} Z_k(t) = B_i(t) + \rho_I U_i(t) - \rho_i U_I(t) \ \text{ for } \ i = 1,\ldots,I-1 \text{ and } t \geq 0, \tag{7}$$

$$U \text{ is nondecreasing with } U(0) = 0, \tag{8}$$

$$\sum_{k=1}^{K} Z_k(t) = 1 \ \text{ for all } t \geq 0, \ \text{and} \tag{9}$$

$$Z(t) \geq \ 0 \ \text{ for all } t \geq 0. \tag{10}$$

We conclude this section with several comments on the workload imbalance formulation, which gets its name because the basic system state equation (7) is in terms of the $(I-1)$–dimensional workload imbalance process, which measures the total amount of work anywhere in the network for stations $1, ..., I-1$ at time $t$ relative to the amount of work in the network at station $I$ at time $t$. Notice that we have arbitrarily chosen to minimize the long run expected average idleness rate of server 1. Although $Z$ and $U$ are required to be nonanticipating with respect to the $K$−dimensional Brownian motion $X$, it turns out that the optimal processes will only depend on the $(I-1)$-dimensional Brownian motion $B$. Constraints (8)-(10) are straightforward , since the cumulative idleness process must be nondecreasing, the customer population size is fixed, and the queue length process must be nonnegative. Finally, there is not a unique way to transform the workload formulation into a workload imbalance formulation, and in Section 6 we discuss an alternative transformation.

## 4. The Workload Imbalance Polytope and Extremal Classes

For the two-station case, Harrison and Wein (1990) found an optimal solution $(Z^*, U^*)$ to the workload imbalance formulation (5)-(10), and interpreted this solution in terms of the original queueing system in order to find an effective sequencing policy. Unfortunately, we have been unable to find a closed form solution to (5)-(10) when $I > 2$. Instead, we will be satisfied with gaining a deep enough understanding of the problem so that an effective sequencing policy can be found.

We begin this section by verbally describing problem (5)-(10). Define the $(I-1)$-dimensional workload imbalance process $\hat{W} = (\hat{W}_i)$ by

$$\hat{W}_i(t) = \sum_{k=1}^{K} \hat{M}_{ik} Z_k(t) \quad \text{for } i = 1, ..., I-1, \text{ and } t \geq 0. \tag{11}$$

It is clear from equations (9)-(11) that the workload imbalance process must reside within the *workload imbalance polytope* defined by

$$\{(\hat{w}_1, ..., \hat{w}_{I-1}) : \hat{w}_i = \sum_{k=1}^{K} \hat{M}_{ik} z_k, i = 1, ..., I-1; \sum_{k=1}^{K} z_k = 1; z_k \geq 0, k = 1, ..., K\}. \tag{12}$$

This polytope is the convex hull of the $K$ columns of the workload imbalance profile matrix $\hat{M}$, where the $k^{th}$ column of $\hat{M}$ quantifies the workload imbalance of a class $k$ customer.

By equations (7) and (11), it follows that

$$\hat{W}_i(t) = B_i(t) + \rho_I U_i(t) - \rho_i U_I(t) \quad \text{for } i = 1, ..., I-1 \text{ and } t \geq 0. \tag{13}$$

Thus, the workload imbalance formulation can be analyzed in a two-step procedure. The first problem is to find an optimal control $U^*$ (that is nonanticipating with respect to $B$) to minimize (5) subject to constraints (8) and (13), and subject to the workload imbalance process $\hat{W}$ residing in the workload imbalance polytope defined in (12). The solution $U^*$ to the first problem will lead to an optimal workload imbalance process $\hat{W}^*$ via equation (13) with $U^*$ replacing $U$. The second problem is to choose an optimal process $Z^*$ that is nonanticipating with respect to $B$ and satisfies equations (9)-(11), with $\hat{W}^*$ replacing $\hat{W}$ in (11). We will now discuss the two problems in turn.

The first problem is a multidimensional ergodic singular Brownian control problem. The controller observes the $(I-1)$−dimensional Brownian motion $B$, exerts the nondecreasing controls $U_1, ..., U_I$, and the resulting process is the $(I-1)$−dimensional workload imbalance process given in (13). The objective is to exert as little of the controls as possible (recall that we arbitrarily chose to minimize $U_1$) subject to keeping the controlled process inside the workload imbalance polytope (12). The control problem is described as *singular* because the state of the controlled process can be instantaneously changed by the controller and, as a result, the optimal control process $U$ is continuous but singular (that is, the set of time points at which $U$ increases has measure zero).

When $I = 2$ (see Harrison and Wein 1990), the workload imbalance polytope is a closed interval on the real line, which will be denoted by $[a, b]$, the optimal control processes $U_1^*$ and $U_2^*$ are proportional to the local times at the respective boundaries, and thus the workload imbalance process is a one-dimensional *regulated* or *reflected* Brownian motion (abbreviated hereafter by RBM; see Harrison 1985 for a complete treatment) on the interval $[a, b]$. Since our objective function is to exert the control $U$ as little as possible subject to keeping $\hat{W}$ in $[a, b]$, it is not surprising that the control $U$ is exerted only when the process $\hat{W}$ reaches the two endpoints of the closed interval.

Unfortunately, closed form solutions to ergodic singular control problems have been restricted to one-dimensional problems (see, for example, Karatzas 1983, Taksar 1985, and Wein 1990a). When $I > 2$, the optimal control $U$ will again only be exerted when the $(I - 1)$–dimensional workload imbalance process $\hat{W}$ reaches the boundary of the polytope defined in (12). However, the problem is greatly complicated by the fact that the optimal angle of reflection (exerting different combinations of the components of $U$ yields $2^I$ possible angles of reflection; see Wein 1991 for details) off the faces of the polytope must be found. Kushner (1977,1990) has developed a numerical procedure (called the finite difference approximation method in Kushner 1977, and called the Markov chain approximation method in Kushner and Martins 1990) for solving a wide variety of control problems, including multidimensional ergodic singular control problems. By discretizing the state space and time, this technique allows one to approximate our ergodic singular control problem by a finite state Markov chain control problem with a long run average cost criterion, which in turn can be solved numerically using standard techniques. Kushner and Martins (1990) (and references therein) have developed weak convergence methods to prove that, as the discretization of time and space gets finer, the optimally controlled Markov chain (suitably interpolated) converges to the optimally controlled diffusion, and the optimal cost of the controlled Markov chain converges to the optimal cost of the singular control problem. This procedure was used in Wein (1991) to numerically solve a more difficult constrained ergodic singular control problem arising from a queueing network scheduling problem with controllable inputs. Although we have successfully employed this technique to find numerical solutions to the three-station examples in Section 5, the optimal angles of reflection are not reported here for reasons that will become clear below. However, it is interesting to note that the solution does not appear to be of a simple form, in that the angles of reflection are not constant on each face of the polytope.

We now turn to the second problem in the two step procedure to solve (5)-(10). Given an optimal workload imbalance process $\hat{W}^*$ (via equation (13)) from step one, choose a queue length process $Z^*$ that satisfies constraints (9)-(11), with $\hat{W}^*$ replacing $\hat{W}$ on the left side of equation (11). Let us again begin with the two-station problem considered in Harrison and Wein (1990). In this case, the one-dimensional workload imbalance process $\hat{W}$ is a RBM on the interval $[a, b]$, and $\hat{M}$ is a $K$−dimensional vector, where $\hat{M}_k$ is the workload imbalance for class $k$. Furthermore, $\min_{\{1\leq k\leq K\}} \hat{M}_k = a$ and $\max_{\{1\leq k\leq K\}} \hat{M}_k = b$, and suppose without loss of generality that $\hat{M}_1 = b$ and $\hat{M}_2 = a$, where class 1 is served at station 1 and class 2 is served at station 2. In order to allow the workload imbalance process to evolve in the entire workload imbalance polytope, only the customer classes that correspond to the extreme points of the polytope must have a positive queue length (i.e., $Z_k^*(t) > 0$); the other classes may have a zero queue length for all times $t$. The customer classes that correspond to the extreme points of the polytope will be called *extremal classes.* In the two-station case, the extreme points of the polytope are $a$ and $b$, and $\hat{M}_2 = a$ and $\hat{M}_1 = b$, and thus there are exactly two extremal classes, class 1 and class 2. If we force the other $K - 2$ customer classes to have zero queue length (i.e., $Z_k^*(t) = 0$ for $t \geq 0$ and $k = 3, ..., K$), then $Z_1^*(t) = \gamma(t)$ and $Z_2^*(t) = 1 - \gamma(t)$, where

$\gamma(t) = (\hat{W}^*(t) - a)/(b - a)$ is the unique solution to equations (9)-(11).

Before we turn to the case where $I > 2$, let us interpret the optimal solution $(Z^*, U^*)$ to the two-station case. The workload imbalance process is a RBM on $[a, b]$, and the server idleness is only incurred when the workload imbalance process equals $a$ or $b$. Furthermore, only two customer classes, denoted by classes 1 and 2, ever have a positive queue length. Under heavy traffic conditions, it is well-known (see Whitt 1971, Harrison 1973, Reiman 1983, Johnson 1983, Peterson 1991, and Chen and Mandelbaum 1989 for various queueing systems) that if a static priority discipline is used among the customer classes visiting a particular queue, only the lowest priority customer class will have a positive scaled queue length under heavy traffic conditions. The other customer classes will not see the system in heavy traffic, and thus their queue lengths will be negligible compared to the bottom priority class. Therefore, the solution is interpreted to mean that customers of class 1 (respectively, class 2) are served at station 1 (respectively, station 2) only when there are no other customers present there. Although some ambiguity remains in specifying the entire sequencing policy, the value of $\hat{M}_k$ offers a natural index with which to prioritize the remaining classes. In particular, the proposed workload balancing policy is to award higher priority at station 1 (respectively, station 2) to the classes with the smaller (respectively, larger) values of the index $\hat{M}_k$.

Returning to the case where $I > 2$, in general there may be more extremal classes than stations. Moreover, there will be at least one extremal class at each station. This second observation is most easily seen if we assume the network is perfectly balanced (that is, $\rho_i = 1$ for $i = 1, ..., I$). In this case, $\hat{M}_{ik} = M_{ik} - M_{Ik}$ and any class that achieves either the minimum or maximum value over classes $k = 1, ..., K$ of $\hat{M}_{ik}$ for some $i = 1, ..., I - 1$ will be an extremal class. For $i = 1, ..., I - 1$, one of the classes that achieves $\max_{1 \le k \le K} \hat{M}_{ik}$ must be served at station $i$ because the value of $\hat{M}_{ik}$ is reduced when a customer leaves station $i$, is increased when a customer leaves station $I$, and remains unchanged when a customer leaves any other station. By similar reasoning, one of the classes that achieves $\min_{1 \leq k \leq K} \hat{M}_{ik}$ must be served at station $I$, for $i = 1, ..., I-1$.

Thus, unlike the two-station case, there is not a unique combination of the extremal queue lengths $Z_k^*(t)$ that is consistent (in the sense of equation (11)) with the workload imbalance process when it is in the interior of the workload imbalance polytope. Therefore, although the extremal classes can be readily identified, there appear to be many possible solutions $Z^*$ that will allow the workload imbalance process to evolve in the entire workload imbalance polytope. Moreover, since there are more extremal classes than stations and since a static priority ranking of the customer classes would lead to only one class per station with a positive queue length, it appears that a dynamic sequencing policy is required, rather than a static policy, as in the two-station case.

To summarize this section, problem (5)-(10) has been decomposed into two problems. The first problem is a multidimensional ergodic singular control problem that does not appear to have a closed form solution. However, it is clear that the the controller exerts the cumulative idleness process $U^*$ only when the workload imbalance process $\hat{W}^*$ reaches the boundary of the workload imbalance polytope. Also, an approximate numerical solution that specifies the optimal angles of reflection off the polytope boundary can be obtained using the Markov chain approximation technique described in Kushner and Martins (1990). The second problem involves finding an optimal queue length process $Z^*$ that is consistent with the optimal workload imbalance process $\hat{W}^*$ derived from the first problem. Although there is not a unique solution to this problem, the extremal classes, which are the only classes that receive lowest priority at their respective stations, can be identified.

Because of the nonuniqueness of the solution to the second problem, much ambiguity remains in interpreting the solution to (5)-(10) in terms of the queueing system in order to obtain an effective dynamic sequencing policy. Moreover, it is not clear to us how to use the optimal angles of reflection to identify an effective sequencing policy. Thus, in the remainder of this paper, we will focus on static sequencing policies, and will only briefly discuss possible dynamic policies.

## 5. Static Sequencing Policies

A static sequencing policy uses a fixed priority ranking of the different customer classes at each server in the network. Perhaps the two most commonly studied static policies are the SEPT and SERPT rules. Under a static policy, only one class will have lowest priority at each server, and hence only $I$ customer classes will have a nonzero queue length in the approximating Brownian model. Thus, the workload imbalance process $\hat{W}$ will reside inside the $(I-1)$–dimensional simplex defined by the $I$ columns of the workload imbalance profile matrix $\hat{M}$ corresponding to the lowest priority classes. This simplex will be contained within the workload imbalance polytope defined in (12).

For any arbitrary static policy, suppose class $i$ is awarded lowest priority at station $i$, for $i = 1, ..., I$, and thus classes $I+1, ..., K$ are not bottom priority classes. Then for any value $\hat{W}(t)$ of the workload imbalance process in the $(I-1)$–dimensional simplex, there exists a unique nonnegative solution $Z^*(t)$ to the system of equations

$$\hat{W}(t) = \sum_{k=1}^{I} \hat{M}_{ik} Z_k(t), \tag{14}$$

$$\sum_{k=1}^{I} Z_k(t) = 1. \tag{15}$$

Moreover, since idleness would only be incurred at each station when no customers are present there, the control $U_i^*(t)$ in the idealized Brownian model is only exerted at times $t$ when $Z_i^*(t) = 0$. Thus, by equations (13)-(15), the workload imbalance process would behave as a RBM on the simplex generated by the $I$ lowest priority classes. Also, the angles of reflection off each face would be constant; see Chen (1987) for a definition of RBM on a simplex. Readers are referred to Figure 3 of Section 7, where two-dimensional simplices are shown for the SEPT and SERPT policies for a specific three-station example.

Recall that the primary performance measure for closed queueing networks is the mean throughput rate, which can be calculated from the mean idleness rates at the various stations. There are several numerical techniques (Harrison, Landau, and Shepp 1981 and Trefethen and Williams 1983 use conformal mapping for the two-dimensional case where the underlying Brownian motion process has zero drift, and the Markov chain approximation method of Kushner and Martins 1990 can be used for the general case) available for determining the steady state distribution of a RBM on a simplex and the mean rate of pushing off the boundaries, and the latter measure leads directly to an estimate of the mean idleness rate. Thus, we can approximately analyze the performance of any arbitrary static policy, such as SEPT and SERPT. However, these techniques require a substantial effort, perhaps more than many analysts would be willing to undertake in order to just compare different static policies.

As an alternative, we propose a very simple measure to crudely compare various static policies. To motivate our measure, consider the Brownian model of the perfectly balanced two-station closed network. In this case, the drift of the underlying one-dimensional Brownian motion $B$ is zero, and the steady state distribution of the RBM is uniformly distributed over the simplex, which in this case is the closed interval $[a, b]$. Moreover, the average idleness rate (or the average pushing off the two interval endpoints) is the same for each station, and is inversely proportional to $b - a$, the length of the interval (see Harrison and Wein 1990 for a closed form expression).

Now consider the general multistation case. If the RBM is uniformly distributed over the simplex, then a relative measure of the average idleness rate (or pushing off the boundaries) is the surface of the simplex divided by its volume. For example, in a three-station network, this measure is the perimeter of a triangle divided by its area. This ratio is easy to compute in general, since the volume and the surface of each face can be computed from a determinant. Although our relative measure is correct for the perfectly balanced two-station network, it is a very crude estimate for a multidimensional RBM, since the steady state distribution is not uniform, and the drift, covariance, and angles of reflection of the RBM are being ignored. However, the goal is to develop a very simple measure that hopefully captures the first-order effect that one would observe from a visual inspection of the simplices. Moreover, this approach to performance analysis also extends to a possibly optimal policy, since the numerical solution (via the Markov chain approximation method) to the singular control problem yields the average idleness rate, and a crude estimate of the average idleness rate is just the surface of the workload imbalance polytope divided by its volume. Although we have been unable to identify an optimal policy, one could use this technique to approximately compare the performance of an optimal policy to an arbitrary static policy.

Now that the performance of arbitrary static policies has been discussed, we are now ready to propose an effective static policy. The first step is to find the class from each of the $I$ stations so that the simplex generated by these classes (via the columns of the workload imbalance profile matrix $\hat{M}$) has the minimal ratio of surface-to-volume. For ease of presentation, let us denote these classes by $1, ..., I$, where class $i$ is served at station $i$. By the above discussion, it is clear that our crude measure of performance would predict that a sequencing policy awarding lowest priority to class $i$ at station $i$, for $i = 1, ..., I$, would achieve minimal mean idleness, and hence maximal mean throughput, among the class of static policies.

A simple extension to this idea will be used to prioritize classes $I+1, ..., K$ at their various stations, and hence to complete the specification of the sequencing policy. In order to prioritize the remaining customers at station $i$, let us suppose for the moment that class $i$, the lowest priority class at station $i$, did not exist. Then for each of the remaining classes at station $i$, which are indexed by $n = 1, ..., l_i$, we would compute the surface-to-volume ratio $R_n$ for the simplex generated by class $n$ and the remaining $I - 1$ bottom priority classes. Since the class with the smallest value of $R_n$ would receive lowest priority at station $i$ if class $i$ did not exist, our proposed sequencing policy awards higher priority at station $i$ to the classes with the larger values of $R_n$.

Although one could obtain a more reliable proposed policy by calculating the mean idleness rate using the sophisticated numerical techniques described earlier in place of our crude surface-to-volume measure, much more computation would be required. Furthermore, as will be seen in the next section, our crude relative idleness measure appears to be accurate enough to distinguish between the various static policies.

We have been unable to identify a simple static or dynamic policy that significantly outperforms the static policy described above. In the simulation experiment of Section 7, we also tested an alternative static policy that serves all extremal classes on a first-come first-served basis at their respective stations, and then prioritizes the non-extremal classes in the same order as they were served in the proposed static policy. The hope was that by allowing all extremal classes to have a positive queue length, the workload imbalance process would be allowed to move throughout the entire workload imbalance polytope, as opposed to only moving throughout the simplex of minimal surface-to-volume ratio. However, this policy did not perform significantly better than the proposed static policy, and thus the simulation results for this policy are not reported here.

We have had several ideas for dynamic policies. One policy employs dynamic reduced costs (as in Wein 1991) derived from the mathematical program of maximizing the minimum amount of work queued at any given station, subject to constraints (14)-(15) and (10), for any given value of $\hat{W}(t)$. A second policy would, given the value of $\hat{W}(t)$ at time $t$, derive the simplex of minimal surface-to-volume ratio (with one extreme point per station) containing $\hat{W}(t)$, and would award the lowest priority at time $t$ to the classes corresponding to the extreme points of the simplex. The remaining classes would be prioritized at time $t$ as in the proposed static policy. However, these two policies were not pursued because they would be extremely tedious to implement in a real time setting. Our goal instead is to find a simple and effective sequencing policy.

## 6. An Alternative Workload Imbalance Formulation

By equation (23) and Propositions 2 and 7 of Harrison and Wein (1990), the key to transforming the workload formulation of an $I-$station closed network into a corresponding workload imbalance formulation is to identify a *projection matrix* $P$ that satisfies $P\rho = 0$, where $\rho$ is the traffic intensity vector. This projection matrix yields the workload imbalance profile matrix $\hat{M} = (\hat{M}_{ik})$, which is defined by $\hat{M} = PM$. That is, the matrix $P$ projects class $k$'s workload profile vector $M_{\cdot k} = (M_{1k}, ..., M_{Ik})^T$ in the direction that is parallel to the vector of relative traffic intensities and onto an $(I-1)-$dimensional hyperplane.

Notice that there is no restriction on the choice of hyperplane onto which the workload profile vector is projected. We chose to employ the particular projection stated in (4) because of its relative simplicity and to maintain consistency with Harrison and Wein (1990) and Wein (1990a,b,1991). In these previous studies, a solution was found to the workload formulation and hence the particular transformation used in obtaining the workload imbalance formulation was irrelevant. However, in the present study, an explicit solution to the work'ad formulation is not obtained, and the proposed scheduling policy can depend on the particular transformation that is used. In particular, the transformation defined in (4) causes the workload imbalance process to measure the imbalance of the first $I-1$ stations relative to station $I$. Station $I$ was arbitrarily chosen as the reference station, and if we had chosen a different reference station, the set of extremal classes would not change, but the proposed scheduling policy could be different. Also, the transformation in (4) leads to an asymmetry in (13), in that the control $U_i$ affects only $\hat{W}_i$, for $i = 1, ..., I-1$, whereas $U_I$ affects the entire process $\hat{W}$.

A natural choice for thc hyperplane onto which the workload profile is to be projected is the hyperplane perpendicular to the projection direction. This is also the only choice that will guarantee that the policy obtained is independent of the order of the stations. This choice has an intuitive interpretation. To find the workload imbalance vector of a customer class, we decompose its workload vector into two orthogonal components. One component is proportional to the vector of relative traffic intensites, and its orthogonal complement is defined as the workload imbalance vector of that customer class. The first component represents a redistribution of the total workload in the network proportional to the traffic intensity vector. The second component is the workload imbalance vector that is the difference between the actual workload and an equivalent balanced load. Figure 1 illustrates how the workload imbalance vector is determined for the 2-dimensional case ($I = 2$).

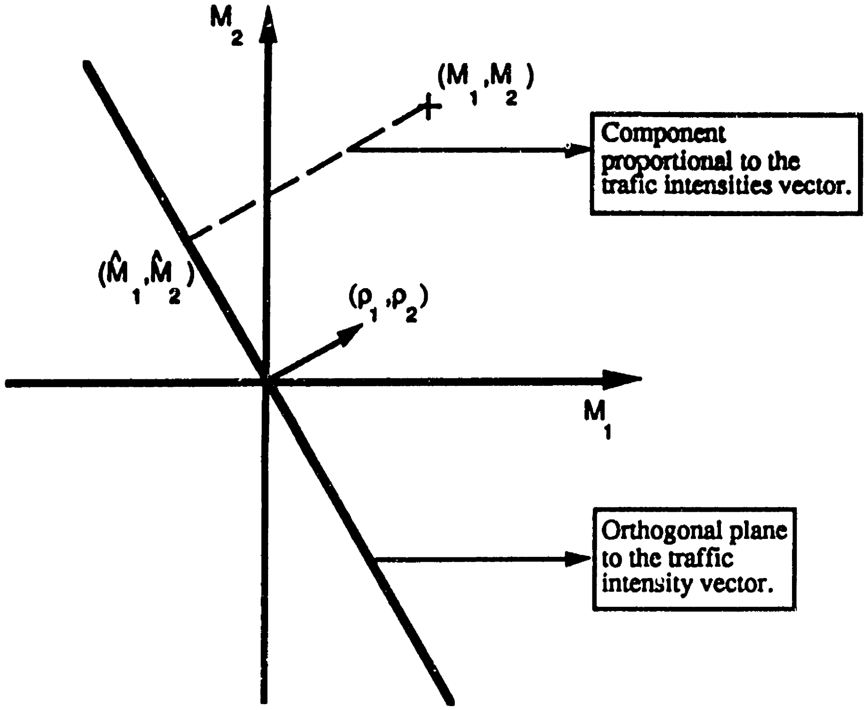

$M_2$
$(M_1, M_2)$
Component proportional to the trafic intensities vector.
$(\hat{M}_1, \hat{M}_2)$
$(\rho_1, \rho_2)$
$M_1$
Orthogonal plane to the traffic intensity vector.

Figure 1 : The workload imbalance vector for a two station network.

In matrix form, the workload imbalance profile matrix $\hat{M}$ can then computed as

$$\hat{M} = PM$$

Where $P$ is the orthogonal projection matrix given by

$$P = (\mathbf{1} - \frac{\rho\rho^T}{\rho^T\rho}), \tag{16}$$

where $\mathbf{1}$ denotes the $I \times I$ identity matrix, and $\rho = (\rho_1, \rho_2, \ldots, \rho_I)^T$. Then the workload imbalance profile matrix $\hat{M} = (\hat{M}_{ik})$, which is of dimension $I \times K$ rather than $(I-1) \times K$, is defined by $\hat{M} \; = \; PM$, yielding

$$\hat{M}_{ik} = M_{ik} - \left(\frac{\rho_i}{\sum_{j=1}^{I} \rho_j^2}\right)\left(\sum_{j=1}^{I} \rho_j M_{jk}\right) \quad \text{for } i = 1, ..., I \text{ and } k = 1, ..., K. \tag{17}$$

By (16), the matrix $P$ projects the workload profile vector $M_{\cdot k} = (M_{1k}, \ldots, M_{Ik})^T$ of each class onto the $(I-1)$-dimensional plane that is orthogonal to the traffic intensity vector and passes through the origin. Thus, the $I$--dimensional workload imbalance process $\hat{W}$, which equals $\{\hat{M}Z(t), t \geq 0\}$, actually *resides* in a polytope of dimension $I-1$. Therefore, by considering the workload imbalance polytope generated by the $K$ points $(\hat{M}_{1k}, ..., \hat{M}_{Ik})^T, k = 1, ..., K$, in the $(I-1)$–dimensional plane orthogonal to the traffic intensity vector, we can use the surface-to-volume algorithm described in the last section to obtain a proposed scheduling policy that is independent of a fixed reference station.

We tested both projections (that is, the transformations based on the matrix $T$ in (4) and the matrix $P$ in (16)) on the five numerical examples considered in the next section. The proposed scheduling policies under the two transformations were identical for the two three-station examples, and were nearly identical for the three four-station examples (for example, classes C3 and B4 were interchanged in Table IX). Also, neither policy dominated in terms of predicting the relative performance of static policies via the surface-to-volume ratio.

## 7. Examples

In this section, simulation results are reported for three networks, including two three-station networks and a four-station network. Although we believe this procedure remains effective for any number of stations, few factories have more than three or four bottleneck stations, and hence we did not examine larger networks. Five examples are treated in all; the three networks are studied under conditions of perfect balance (that is, the traffic intensity is the same for each station in the network), and then the four-station network is studied under two imbalanced scenarios.

There are two objectives in this simulation study: to assess the effectiveness of the proposed static sequencing policy described in Section 5, and to assess the accuracy of the surface-to-volume ratio in predicting the relative performance of various static policies. To achieve the first goal, five sequencing policies are tested for each example: the proposed static policy (denoted by BROWNIAN in the tables below), the first-come first-served (FCFS) policy, the SEPT rule, the SERPT rule, and the least work next queue (LWNQ) rule. This last rule, which gives dynamic priority at each station to the class whose next station has the least amount of work in it, appears to be a reasonable candidate for a closed network setting, where the sole issue is to avoid server idleness.

Recall that the objective of the scheduling problem is to maximize the mean throughput rate for a fixed population level $N$. In the simulation results below, the population size $N$ for each policy is set (via a one-dimensional search using simulation) so as to achieve a fixed mean throughput rate, and we will instead record the mean sojourn times. As mentioned earlier, minimization of mean sojourn time is equivalent to maximization of mean throughput rate in a closed network. We compare mean sojourn time at a specified throughput rate because this is how factories are generally run: they attempt to choose their customer population level to meet the specified exogenous demand rate, and smaller mean sojourn times imply better performance. For each policy, ten independent runs are made, each consisting of 10,000 customer completions and no initialization periods.

In order to assess the effectiveness of the surface-to-volume ratio in predicting the relative performance of static policies, the SEPT policy, the SERPT policy, and the proposed static policy are tested at constant population levels, and the mean idleness rate at each station is observed; notice that these measurements are more in line with the original problem statement of minimizing the idleness rate (or, equivalently, maximizing the throughput rate) for a fixed population level. For examples 1-3, the idleness rate for a particular scheduling policy is defined as the average of the mean idleness rates at each station in the network. For examples 4 and 5, the idleness rate is defined as the mean idleness rate at station 1, which as the heaviest loaded station. The *normalized idleness rate* of each policy, which is the idleness rate of the policy divided by the idleness rate of the proposed static policy, is then compared to the corresponding *normalized surface-to-volume ratio*, which is the surface-to-volume ratio of the policy divided by the surface-to-volume ratio of the proposed policy.

**Example 1.** The first network is populated by three types of customers, denoted by A, B, and C, and the specified mix employs equal quantities of all three types; that is, whenever a customer exits the network, the newly injected customer is of type A, B, or C with probability one-third. Table I describes the deterministic route of each customer type, and gives the mean service time for each stage of service. All service time distributions in this section are assumed to be exponential, although our results hold for any general service time distributions. Since each customer class corresponds to a combination of customer type and stage of completion, the twelve customer classes are designated (and ordered from $k = 1, ..., 12$) by (A1,A2,A3,B1,...,B5,C1,...,C4).

From Table I, we find that the $3 \times 12$ workload profile matrix $M$ is given by

$$M = \begin{pmatrix} 4 & 4 & 0 & 10 & 2 & 2 & 2 & 0 & 4 & 4 & 4 & 0 \\ 1 & 1 & 1 & 13 & 13 & 7 & 7 & 7 & 4 & 0 & 0 & 0 \\ 6 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 11 & 11 & 2 & 2 \end{pmatrix}, \tag{24}$$

where $M_{ik}$ is the expected remaining processing time at station $i$ for a class $k$ customer until that customer exits the network. Since $q = (\frac{1}{3}\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0\ 0\ \frac{1}{3}\ 0\ 0\ 0)^T$, we have $v_1 = v_2 = v_3 = 6$, implying $\rho_1 = \rho_2 = \rho_3 = 1$.

Using the projection $T$, the $2 \times 12$ workload imbalance profile matrix $\hat{M}$ is given by

$$\hat{M} = \begin{pmatrix} -2 & 4 & 0 & 9 & 1 & 1 & 2 & 0 & -7 & -7 & 2 & -2 \\ -5 & 1 & 1 & 12 & 12 & 6 & 7 & 7 & -7 & -11 & -2 & -2 \end{pmatrix}. \tag{25}$$

|  |  | MEAN |
|---|---|---|
| CUSTOMER |  | SERVICE |
| TYPE | ROUTE | TIMES |
| A | 3 → 1 → 2 | 6.0 4.0 1.0 |
| B | 1 → 2 → 3 → 1 → 2 | 8.0 6.0 1.0 2.0 7.0 |
| C | 2 → 3 → 1 → 3 | 4.0 9.0 4.0 2.0 |

**Table I.** Description of example 1.

The projection matrix $P$ is given by

$$P = \begin{pmatrix} 2/3 & -1/3 & -1/3 \\ -1/3 & 2/3 & -1/3 \\ -1/3 & -1/3 & 2/3 \end{pmatrix},$$

and workload imbalance profile matrix $\hat{M}^P$ is given by

$$\hat{M}^P = \begin{pmatrix} \frac{1}{3} & \frac{7}{3} & \frac{-1}{3} & 2 & \frac{-10}{3} & \frac{-4}{3} & -1 & \frac{-7}{3} & \frac{-7}{3} & -1 & 2 & \frac{-2}{3} \\ \frac{-8}{3} & \frac{-2}{3} & \frac{2}{3} & 5 & \frac{23}{3} & \frac{11}{3} & 4 & \frac{14}{3} & \frac{-7}{3} & -5 & -2 & \frac{-2}{3} \\ \frac{7}{3} & \frac{-5}{3} & \frac{-1}{3} & -7 & \frac{-13}{3} & \frac{-7}{3} & -3 & \frac{-7}{3} & \frac{14}{3} & 6 & 0 & \frac{4}{3} \end{pmatrix} \qquad (26)$$

The twelve points $(\hat{M}_{1k}, \hat{M}_{2k})$ are plotted in Figure 2, where the workload imbalance polytope, which is the convex hull of these points, is also displayed. Figure 3 displays the twelve points $(\hat{M}^P_{1k}, \hat{M}^P_{2k}, \hat{M}^P_{3k})$ in the plane orthogonal to the traffic intensity vector. In both cases six of the twelve classes are extremal classes, and the number beside each extremal point is the station that serves the corresponding extremal class. Recall that the static priority policy finds the simplex (generated by exactly one class from each station) of minimal surface-to-volume ratio, and awards lowest priority to these three classes at their respective station. Readers can easily see that in both Figures the two highest points and the lowest point, which correspond to class B1 at station 1, class B2 at station 2, and class C2 at station 3, generate the simplex of largest volume, and it turns out that this simplex also has minimal surface-to-volume ratio. Thus, these three classes receive lowest priority at their respective stations under the BROWNIAN policy. A complete specification of the three static policies is exhibited in Table II, where the parenthesis denotes a tie between two classes; in this case, the customers of the two classes are grouped together and served on a FCFS basis. In this example the policy derived by our heuristic was identical for projection methods. Figures 4 and 5 show the simplices for the SEPT, SERPT, and BROWNIAN policies under the projection $T$ and $P$ respectively. A visual inspection reveals that we would expect the SEPT policy to outperform the SERPT policy, since its simplex has a significantly larger volume.

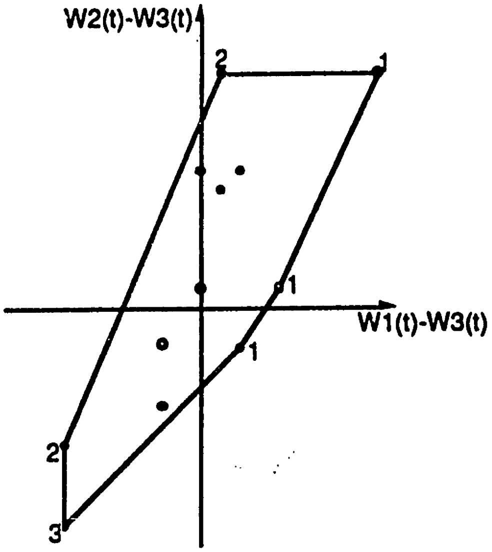


Figure 2 : The workload imbalance polytope under projection $T$ for example 1.

Simulation results for this example are reported in Tables III and IV. In Table III, the population size, mean sojourn time, and mean throughput rate, along with appropriate 95% confidence intervals, are reported for each of the five sequencing policies, where the throughput rate of 0.149 customers per unit time corresponds to a server utilization of 89.4%. It can be seen that the proposed static policy easily outperforms the other four policies, offering a 43.8% reduction in mean sojourn time versus

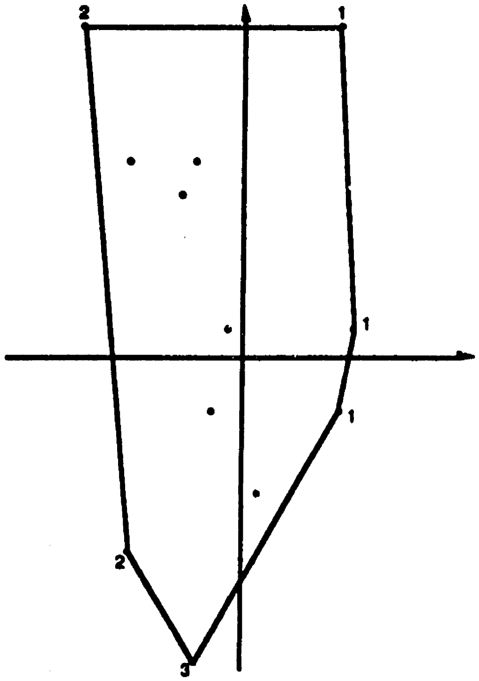


Figure 3 : The workload imbalance polytope under projection $P$ for example 1.

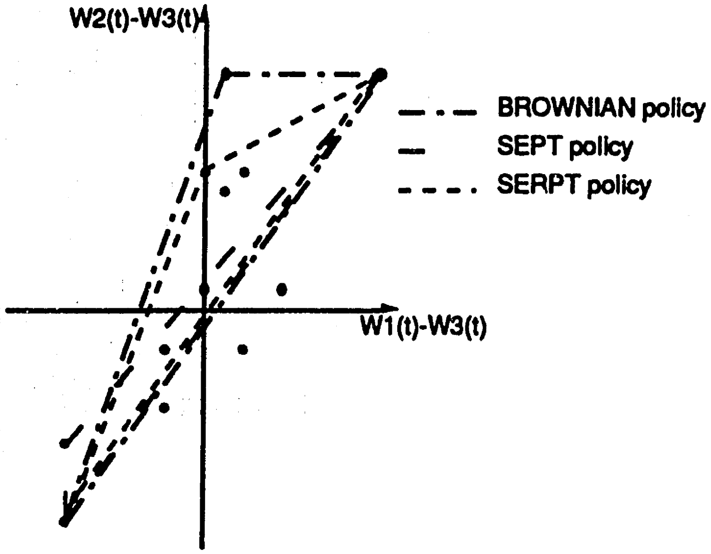


Figure 4 : The workload imbalance simplices under projection $T$ for the BROWNIAN, SEPT, and SERPT policies.

FCFS. Notice that the LWNQ policy does not offer much improvement over FCFS and, as expected, SEPT outperforms SERPT. The results for FCFS and SERPT do not match the corresponding simulation results in Section 10 of Wein (1991) because

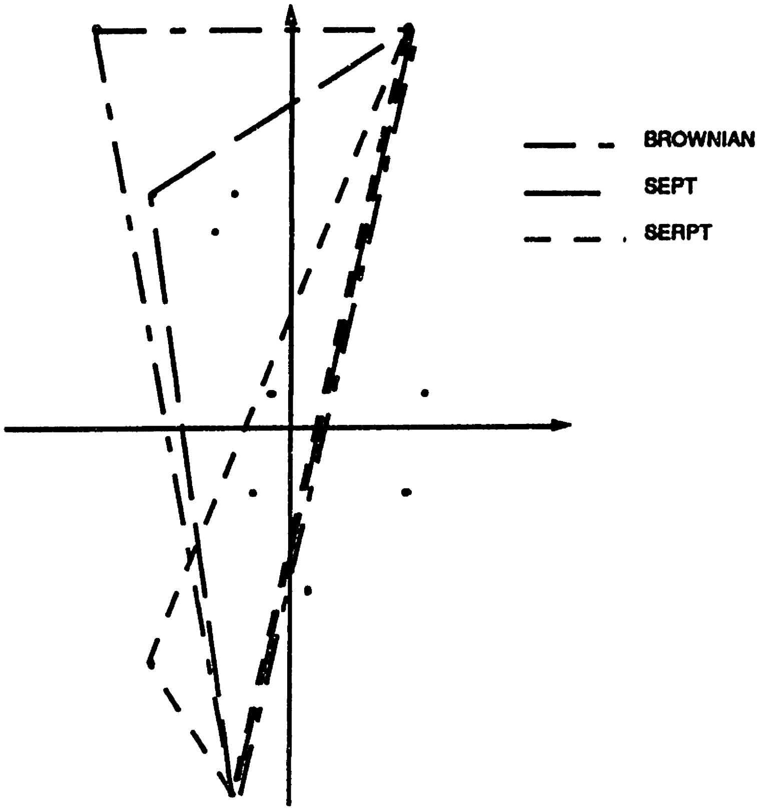


Figure 5 : The workload imbalance simplices under projection $P$ for the BROWNIAN, SEPT, and SERPT policies.

our study chooses the entering customer type in a Markovian manner, whereas the other study chooses the entering type in a deterministic manner.

| POLICY | STATION 1 | STATION 2 | STATION 3 |
|---|---|---|---|
| BROWNIAN | B4 C3 A2 B1 | A3 C1 B5 B2 | B3 C4 A1 C2 |
| SEPT | B4 (A2,C3) B1 | A3 C1 B2 B5 | B3 C4 A1 C2 |
| SERPT | A2 C3 B4 B1 | A3 B5 B2 C1 | C4 B3 A1 C2 |

**Table II.** Static sequencing policies for example 1.

In Table IV, the three static policies are compared at three different population levels, and the observed idleness rates, normalized so that the idleness rate of the

| SEQUENCING POLICY | POPULATION SIZE | MEAN SOJOURN TIME | MEAN THROUGHPUT |
|---|---|---|---|
| BROWNIAN | 14 | 93.8 (±0.57) | 0.149 (±.0008) |
| SEPT | 20 | 134 (±0.66) | 0.149 (±.0007) |
| LWNQ | 24 | 161 (±1.05) | 0.149 (±.0010) |
| FCFS | 25 | 167 (±1.05) | 0.149 (±.0010) |
| SERPT | 30 | 201 (±1.20) | 0.149 (±.0009) |

**Table III.** Comparison of mean sojourn times for example 1.

proposed policy is one, are compared to the normalized versions of the estimated idleness rates (via the surface-to-volume ratios). If the surface-to-volume ratio was accurate, we would expect to see the simulated normalized idleness rates approach the estimated idleness rates as the population size increases. For example, we predict that the SERPT rule will have 2.7 times as much idleness as the BROWNIAN policy when the population size is very large. When the population size is 45, the SERPT rule actually incurs 2.6 times as much idleness as the BROWNIAN policy, and thus the surface-to-volume ratio is quite accurate in this case. Although the ratio is not as accurate a predictor in the SEPT case, the measure correctly predicts the relative performance of the three policies. The surface to ratio figures in this and the following tables are those obtained using the $P$ projection, the figures obtained using the the $T$ projection are very close and are reported in Chevalier and Wein (1992).

Before turning to example 2, we want to mention that the workload imbalance polytope can be helpful in developing a fast heuristic solution to a related scheduling problem considered in Wein (1991). This study develops a customer release and priority sequencing policy to minimize mean sojourn time subject to a minimum mean

| STATIC<br>SEQUENCING<br>POLICY | POPULATION<br>SIZE | NORMALIZED<br>IDLENESS<br>RATE | NORMALIZED<br>SURFACE-TO-<br>VOLUME RATIO |
|---|---|---|---|
| BROWNIAN | 15 | 1.00 | 1.00 |
| SEPT | 15 | 1.39 | 1.32 |
| SERPT | 15 | 1.68 | 2.53 |
| BROWNIAN | 30 | 1.00 | 1.00 |
| SEPT | 30 | 1.60 | 1.32 |
| SERPT | 30 | 2.30 | 2.53 |
| BROWNIAN | 45 | 1.00 | 1.00 |
| SEPT | 45 | 1.77 | 1.32 |
| SERPT | 45 | 2.60 | 2.53 |

**Table IV.** Actual and predicted normalized idleness rates for example 1.

throughput rate constraint. The resulting constrained singular ergodic Brownian control problem is to find a region in $R^{I-1}$ in which to reflect the workload imbalance process. When there is perfect balance between the stations in the two-station case, it turns out that the region derived in the controllable inputs problem is homothetic (that is, of similar shape) to the workload imbalance polytope of the corresponding closed network problem. Moreover, this relationship appears to roughly hold in the multistation case; readers may compare the similarity in shapes of the workload imbalance polytope in Figure 1 with the optimal reflecting boundary in Figure 5 of Wein (1991), which also considered the network described in Table I. This is significant because the identification of the workload imbalance polytope is much less computationally burdensome than the derivation of the optimal reflecting boundary.

**Example 2.** This example is also a three-station network visited by three customer types. The customer routes and mean service times are given in Table V, and the mix of customer types is again $(1/3, 1/3, 1/3)$. Readers may verify that $\rho_1 = \rho_2 = \rho_3 = 1$, and the workload imbalance profile matrices for both projection are given by

$$\hat{M} = \begin{pmatrix} -1 & -2 & 4 & 4 & -1 & -4 & -4 & 2 & -3 & -3 \\ -1 & -1 & 5 & 0 & 2 & 2 & -4 & -1 & -1 & -3 \end{pmatrix}. \qquad (27)$$

$$\hat{M}^P = \begin{pmatrix} -1/3 & -1 & 1 & 8/3 & -4/3 & -10/3 & -4/3 & 5/3 & -5/3 & -1 \\ -1/3 & 0 & 2 & -4/3 & 5/3 & 8/3 & -4/3 & -4/3 & 1/3 & -1 \\ 2/3 & 1 & -3 & -4/3 & -1/3 & 2/3 & 8/3 & -1/3 & 4/3 & 1 \end{pmatrix}. \qquad (28)$$

Although we do not exhibit the simplices for the static policies here, a visual inspection of these simplices suggests that the BROWNIAN policy should outperform the SEPT policy, which in turn should outperform the SERPT policy. In this case again the policy found by our heuristic is identical for both projections.

| | | MEAN |
|---|---|---|
| CUSTOMER | | SERVICE |
| TYPE | ROUTE | TIMES |
| A | 1 → 3 → 2 → 1 | 1.0 6.0 5.0 4.0 |
| B | 1 → 2 → 3 | 3.0 6.0 4.0 |
| C | 1 → 2 → 3 | 5.0 2.0 3.0 |

**Table V.** Description of example 2.

Simulation results for Example 2 are found in Tables VI and VII. The mean throughput rate of 0.210 in Table VI corresponds to a mean server utilization of 91.0%. Once again, the BROWNIAN policy outperforms the other four policies, and offers a 32.2% reduction in mean sojourn time versus FCFS. The LWNQ policy did not perform as well as FCFS and, as predicted, SEPT outperformed SERPT. The normalized ratios clearly overestimate the normalized idleness rates in Table VII. However, the relative values of the three normalized idleness rates were predicted reasonably accurately when $N = 45$, since (2.95-1.00)/(7.64-1.00)=.294, and (1.67-1.00)/(2.93-1.00)=.347.

| SEQUENCING <br> POLICY | POPULATION <br> SIZE | MEAN <br> SOJOURN TIME | MEAN <br> THROUGHPUT |
|---|---|---|---|
| BROWNIAN | 17 | 80.7 (±0.37) | 0.210 (±.0009) |
| SEPT | 22 | 105 (±0.62) | 0.210 (±.0013) |
| FCFS | 25 | 119 (±0.55) | 0.210 (±.0010) |
| LWNQ | 29 | 138 (±0.75) | 0.210 (±.0011) |
| SERPT | 45 | 213 (±1.44) | 0.210 (±.0014) |

**Table VI.** Comparison of mean sojourn times for example 2.

**Example 3.** This four-station network, which is described in Table VIII, contains four customer types and a total of twenty customer classes. A newly injected customer is of each type with probability 0.25, and thus the network is again perfectly balanced ($\rho_i = 1$ for $i = 1, ..., 4$). The scheduling policy obtained from the $P$ projection is provided in Table IX. The policy obtained from the $T$ projection is identical unless that the priorities of the customer classes B4 and C3 are reversed at station 3.

The simulation results for example 3 are displayed in Tables X and XI (because the results are identical for the policies derived from both projections, only one set of results is displayed). As can be seen from Table XI, the normalized surface-to-volume ratio for the SEPT policy is only 1.28 in this case, and so we would predict that the difference in performance between the BROWNIAN and SEPT policies would be less

| STATIC<br>SEQUENCING<br>POLICY | POPULATION<br>SIZE | NORMALIZED<br>IDLENESS<br>RATE | NORMALIZED<br>SURFACE-TO-<br>VOLUME RATIO |
|---|---|---|---|
| BROWNIAN | 15 | 1.00 | 1.00 |
| SEPT | 15 | 1.21 | 2.95 |
| SERPT | 15 | 1.74 | 7.64 |
| BROWNIAN | 30 | 1.00 | 1.00 |
| SEPT | 30 | 1.27 | 2.95 |
| SERPT | 30 | 2.40 | 7.64 |
| BROWNIAN | 45 | 1.00 | 1.00 |
| SEPT | 45 | 1.67 | 2.95 |
| SERPT | 45 | 2.93 | 7.64 |

**Table VII.** Actual and predicted normalized idleness rates for example 2.

| CUSTOMER<br>TYPE | ROUTE | MEAN<br>SERVICE<br>TIMES |
|---|---|---|
| A | 1 → 2 → 3 → 4 | 2.0 4.0 3.0 7.0 |
| B | 4 → 2 → 1 → 3 → 2 → 1 | 3.0 5.0 2.0 4.0 1.0 6.0 |
| C | 2 → 1 → 3 → 4 → 3 → 2 | 2.0 8.0 2.0 9.0 5.0 6.0 |
| D | 2 → 4 → 1 → 3 | 2.0 1.0 2.0 6.0 |

**Table VIII.** Description of example 3.

in this example than in the previous two examples. This prediction is verified in Table IX, where the desired throughput rate is 0.165, which corresponds to a server

| POLICIES: | BROWNIAN | SEPT | SERPT |
|---|---|---|---|
| STATION 1 | A1 D3 C2 B3 B6 | (A1,B3,D3) B6 C2 | B6 D3 B3 A1 C2 |
| STATION 2 | B5 D1 C1 A2 B2 C6 | B5 (C1,D1) A2 B2 C6 | C6 B5 D1 A2 B2 C1 |
| STATION 3 | A3 B4 C3 C5 D4 | C3 A3 B4 C5 D4 | D4 A3 (B4,C5) C3 |
| STATION 4 | D2 B1 C4 A4 | D2 B1 A4 C4 | A4 D2 C4 B1 |

**Table IX.** Static sequencing policies for example 1.

utlization rate of only 82.5%. Since the SEPT policy was unable to achieve this rate exactly, we have included two rows in Table X for this policy, where each row uses a different population size.

Once again, the BROWNIAN policy offers a significant reduction (38.0%) in mean sojourn time versus FCFS. There is a very wide range of performance among the policies, with the SERPT policy possessing a mean sojourn time that is 7.6 times larger than that of the BROWNIAN policy. In this example, the normalized surface-to-volume ratios underestimated the normalized idleness rates at $N = 45$, although the relative values of the three normalized idleness rates were accurately predicted, since (1.27-1.00)/(4.79-1.00)=.071, and (1.44-1.00)/(7.00-1.00)=.073.

**Example 4.** The purpose of examples 4 and 5 is to investigate the robustness of our approximation procedure with respect to the balance of the workload across the network. These two examples use the same customer routes and entering type mix as in example 3, but the processing times are altered to achieve an imbalanced workload. For this example, the processing times of all customer classes processed at station 2 (station 3 and 4, respectively) are multiplied by 0.95 (0.90 and 0.85, respectively). The resulting vector of relative traffic intensities is $\rho = (1.0, 0.95, 0.90, 0.85)^T$, and since the throughput of 0.165 corresponds to an 82.5% server utilization in example

| SEQUENCING POLICY | POPULATION SIZE | MEAN SOJOURN TIME | MEAN THROUGHPUT |
|---|---|---|---|
| BROWNIAN | 13 | 78.8 (±0.37) | 0.165 (±.0010) |
| SEPT | 13 | 79.4 (±0.50) | 0.164 (±.0010) |
| SEPT | 14 | 84.4 (±0.43) | 0.166 (±.0008) |
| FCFS | 21 | 127 (±0.73) | 0.165 (±.0014) |
| LWNQ | 55 | 332 (±2.42) | 0.165 (±.0012) |
| SERPT | 100 | 601 (±6.34) | 0.165 (±.0017) |

**Table X.** Comparison of mean sojourn times for example 3.

3, the server utilizations for the four stations are (82.5%,78.4%,74.2%,70.9%).

Because the network structure has remained unchanged, the BROWNIAN policy is very similar for examples 3, 4, and 5; the policies are identical for all three examples under the $P$ projection, and under the $T$ projection for example 3 and 4, and the policy for example 5 for the projection $T$ differs onl/ in that classes C3 and B4 are interchanged at station 3. Due to the low population levels employed in examples 4 and 5, only one population level for each policy achieved a throughput rate within 0.003 of the desired throughput rate of 0.165. However, all policies did have one population level that achieved a throughput between 0.164 and 0.166, and only these results are reported in Tables XII and XIV.

Also, when the population size is 60, the mean idleness rate for station 1 under the BROWNIAN policy is zero under the three digit precision of the computer simulation package SIMAN. Thus, we could not compute valid normalized idleness rates at this population level in Table XIII. This phenomenon also occured under the population levels of 40 and 60 in Example 5, and thus Table XV reports results only for a

| STATIC<br>SEQUENCING<br>POLICY | POPULATION<br>SIZE | NORMALIZED<br>IDLENESS<br>RATE | NORMALIZED<br>SURFACE-TO-<br>VOLUME RATIO |
|---|---|---|---|
| BROWNIAN | 20 | 1.00 | 1.00 |
| SEPT | 20 | 1.09 | 1.27 |
| SERPT | 20 | 2.37 | 4.79 |
| BROWNIAN | 40 | 1.00 | 1.00 |
| SEPT | 40 | 1.29 | 1.27 |
| SERPT | 40 | 4.58 | 4.79 |
| BROWNIAN | 60 | 1.00 | 1.00 |
| SEPT | 60 | 1.44 | 1.27 |
| SERPT | 60 | 7.00 | 4.79 |

**Table XI.** Actual and predicted normalized idleness rates for example 3.

population size of 20.

The results presented in Tables XII and XIII suggest that the BROWNIAN scheduling rule performs roughly as well in this case as in the previous example. Although the percentage reductions in mean sojourn time achieved by the BROWNIAN policy are smaller in Table XII than in Table X, the normalized idleness rates are higher in Table XIII than Table XI.

**Example 5.** Here, we alter example 3 by multiplying the processing times at stations 2, 3, and 4 by 0.9, 0.8, and 0.7, respectively. At a throughput rate of 0.165, the server utilization levels for the four stations are (82.5%,74.2%,66.0%,57.8%). The BROWNIAN policy requires only eight customers to achieve the desired throughput

| SEQUENCING POLICY | POPULATION SIZE | MEAN SOJOURN TIME | MEAN THROUGHPUT |
|---|---|---|---|
| BROWNIAN | 10 | 60.2 (±0.22) | 0.166 (±.0006) |
| SEPT | 10 | 60.7 (±0.32) | 0.165 (±.0009) |
| FCFS | 14 | 84.9 (±0.53) | 0.165 (±.0010) |
| LWNQ | 18 | 109 (±0.59) | 0.165 (±.0009) |
| SERPT | 19 | 115 (±0.68) | 0.165 (±.0010) |

**Table XII.** Comparison of mean sojourn times for example 4.

| STATIC SEQUENCING POLICY | POPULATION SIZE | NORMALIZED IDLENESS RATE | NORMALIZED SURFACE-TO-VOLUME RATIO |
|---|---|---|---|
| BROWNIAN | 20 | 1.00 | 1.00 |
| SEPT | 20 | 1.25 | 1.27 |
| SERPT | 20 | 3.33 | 4.90 |
| BROWNIAN | 40 | 1.00 | 1.00 |
| SEPT | 40 | 2.00 | 1.27 |
| SERPT | 40 | 16.5 | 4.90 |

**Table XIII.** Actual and predicted normalized idleness rates for example 4.

rate, and hence the load on this network is neither heavy nor balanced.

Nonetheless, the results in Tables XIV and XV are very encouraging. Although SEPT performs slightly better than the BROWNIAN policy in Table XIV, the BROWNIAN policy still offers a significant improvement over FCFS. Moreover, under the fixed population level of twenty customers, the normalized idleness rate of SEPT and SERPT are higher in Table XV than in Table XIII; thus, at a fixed population size, the effectiveness of the BROWNIAN policy relative to these two static policies appears to be increasing as the network load becomes more imbalanced.

| SEQUENCING POLICY | POPULATION SIZE | MEAN SOJOURN TIME | MEAN THROUGHPUT |
|---|---|---|---|
| SEPT | 8 | 48.2 (±0.22) | 0.166 (±.0008) |
| BROWNIAN | 8 | 48.6 (±0.25) | 0.165 (±.0009) |
| FCFS | 10 | 60.9 (±0.38) | 0.164 (±.0010) |
| LWNQ | 11 | 66.2 (±0.29) | 0.166 (±.0007) |
| SERPT | 14 | 84.6 (±0.36) | 0.165 (±.0007) |

**Table XIV.** Comparison of mean sojourn times for example 5.

| STATIC SEQUENCING POLICY | POPULATION SIZE | NORMALIZED IDLENESS RATE | NORMALIZED SURFACE-TO-VOLUME RATIO |
|---|---|---|---|
| BROWNIAN | 20 | 1.00 | 1.00 |
| SEPT | 20 | 1.35 | 1.27 |
| SERPT | 20 | 6.80 | 5.01 |

**Table XV.** Actual and predicted normalized idleness rates for example 5.

## 8. Conclusions

We should note that although the SEPT policy outperformed the SERPT policy in all five examples, counterexamples to this phenomenon can be easily constructed. Readers are referred to the two-station closed network example in Harrison and Wein (1990), where the SEPT policy is easily outperformed by SERPT. However, the Brownian analysis does explain why the SERPT policy will often perform poorly in a closed network under balanced heavy loading conditions. The lowest priority class at each station under SERPT is the class with the maximum value of $\sum_{i=1}^{I} M_{ik}$, and these classes usually correspond to early stages on the customers' routes. Since $Mq$ is proportional to the vector $\rho$ of traffic intensities (whose components are close to each other in value by the balanced heavy loading conditions), these classes will not often be extremal classes of the workload imbalance polytope, unless there are significant differences in workload imbalance across entering customer types.

It is interesting to note that in Tables IV, VII, XI, and XIII, the normalized idleness rates of SEPT and SERPT increase as the population size increases, and thus the BROWNIAN policy's relative performance is better at higher population levels. This may be due in part because the policy is derived under balanced heavy loading conditions, and in part because, as in open networks, the improvements from scheduling may increase as network congestion increases. Thus, the similarity in performance between the BROWNIAN and SEPT policies in Tables X, XII, and XIV may be partly attributed to the relatively low population sizes considered.

In none of our examples did the choice of a projection matrix make a significant difference. Nevertheless, as the size of the network increases, the dimension of the projection space will increase. As a result the possible distortion due to the choice of a particular projection augments too. The orthogonal projection $P$, although requiring a little more computation, has the advantage of having a certain flavor of 'objectivity' that would probably make it a more robust choice.

In closing, we comment on the range of applicability of the proposed scheduling policy with respect to the balanced heavy loading conditions, which require a large population size $N$ and a well balanced set of relative traffic intensities $\rho = (\rho_i)$. Our numerical experience has suggested that a good indicator of the magnitude of the load on a closed queueing network is the population size divided by the number of stations (that is, $N/I$). In Tables XII and XIV, $N/I$ equals 2.5 and 2.0, respectively, under the BROWNIAN policy, and this policy is more effective than FCFS and comparable to SEPT. Similarly, in Section 4 of Wein and Ou (1991), this policy's performance is better than FCFS and comparable to SEPT in a two-station example where $N/I$ varies between 1.5 and 2.5. Thus, we suspect that the BROWNIAN policy can be safely applied whenever $N/I$ is greater than three, although we have not tested the policy on networks with more than four stations. The policy's insensitivity to the magnitude of the load is not very surprising, since one would expect that a policy effective at reducing server idleness in a balanced closed network with many customers would remain so in the same network with less customers.

The last two examples show that the policy appears to be surprisingly robust with respect to the imbalance of the network's load, at least when the ratio of the smallest to the largest traffic intensity in the network is greater than or equal to 0.7. In fact, comparing Tables XI, XIII, and XV, the relative effectiveness (as measured by the normalized idleness rates) of the BROWNIAN policy at a fixed population level appears to *increase* as the imbalance of the the workload increases. Finally, as mentioned in the Introduction, further study is required to assess the effectiveness of the BROWNIAN procedure for scheduling a large network with many bottleneck and nonbottleneck stations. However, the simulation results in this section suggest that the simple procedure of using the proposed algorithm on the *entire network* (that is, if the actual network has $I$ stations, then employ the $(I-1)$–dimensional workload imbalance polytope, regardless of whether or not the balanced heavy loading conditions hold) is worthy of further investigation.

In summary, we have analyzed a Brownian approximation to the scheduling problem of maximizing the mean throughput rate of a general multistation, multiclass closed queueing network. The insights gained from this analysis have led to an identification of an effective static policy, and to a crude but robust procedure for predicting the performance of an arbitrary static sequencing policy. We believe the most interesting aspect of this study is the dramatic impact that different static policies can have on system performance.

## References

Baskett, F., K. M. Chandy, R. R. Muntz, and F. G. Palacios. 1975. Open, Closed and Mixed Networks of Queues with Different Classes of Customers. *J. Assoc. Comput. Mach.* **22**, 248-260.

Chen, H. 1987. Stochastic Flow Networks: Bottleneck Analysis, Fluid Approximations, and Diffusion Limits. Unpublished Ph.D. thesis, Dept. of Engineering-Economic Systems, Stanford U., Stanford, CA.

Chen, H. and A. Mandelbaum. 1989. Discrete Flow Networks: Diffusion Approximations and Bottlenecks. To appear in *Annals of Probability*.

Harrison, J. M. 1973. A Limit Theorem for Priority Queues in Heavy Traffic. *J. Appl. Prob.* **10**, 907-912.

Harrison, J. M. 1985. *Brownian Motion and Stochastic Flow Systems.* John Wiley and Sons, New York.

Harrison, J. M. 1988. Brownian Models of Queueing Networks with Heterogeneous Customer Populations, in W. Fleming and P. L. Lions (eds.), *Stochastic Differential Systems, Stochastic Control Theory and Applications,* IMA Volume **10**, Springer-Verlag, New York, 147-186.

Harrison, J. M., H. J. Landau, and L. A. Shepp. 1985. The Stationary Distribution of Reflected Brownian Motion in a Planar Region. *Annals of Probability* **13**, 744-757.

Harrison, J. M. and L. M. Wein. 1989. Scheduling Networks of Queues: Heavy Traffic Analysis of a Simple Open Network. *Queueing Systems* **5**, 265-280.

Harrison, J. M. and L. M. Wein. 1990. Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Closed Network. *Operations Research* **38**, 1052-1064.

Johnson, D. P. 1983. Diffusion Approximations for Optimal Filtering of Jump Processes and for Queueing Networks. Unpublished Ph.D. thesis, Dept. of Mathematics, Univ. of Wisconsin, Madison.

Karatzas, I. 1983. A Class of Singular Stochastic Control Problems. *Adv. Appl. Prob.* **15**, 225-254.

Kelly, F. P. 1979. *Reversibility and Stochastic Networks*, John Wiley and Sons, New York.

Klimov, G. P. 1974. Time Sharing Service Systems I. *Th. Prob. Appl.* 19,532-551.

Kushner, H. J. 1977. *Probability Methods for Approximations in Stochastic Control and for Elliptic Equations.* Academic Press, New York.

Kushner, H. J. 1990. Numerical Methods for Stochastic Control Problems in Continuous Time. *SIAM J. Control and Optimization* **28**, 999-1048.

Kushner, H. J. and F. L. Martins. 1990. Numerical Methods for Stochastic Singularly Controlled Problems. Technical Report, Div. Applied Math., Brown U., Providence, R. I.

Little, J. D. C. 1961. A Proof of the Queueing Formula $L = \lambda W$. *Operations Research* **9**, 383-387.

Peterson, W. P. 1991. A Heavy Traffic Limit Theorem for Networks of Queues with Multiple Customer Types. *Mathematics of Operations Research* **16**, 90-118.

Reiman, M. I. 1983. Some Diffusion Approximations with State Space Collapse. *Proc. Intl. Seminar on Modeling and Performance Evaluation Methodology*, Springer-Verlag, Berlin.

Taksar, M. I. 1985. Average Optimal Singular Control and a Related Stopping Problem. *Mathematics of Operations Research* **10**, 63-81.

Trefethen, L. N. and R. J. Williams. 1986. Conformal Mapping Solution of Laplace's Equation on a Polygon with Oblique Derivative Boundary Conditions. *J. Comp. Appl. Math.* **14**, 227-249.

Wein, L. M. 1990a. Optimal Control of a Two-Station Brownian Network. *Mathematics of Operations Research* **15**, 215-242.

Wein, L. M. 1990b. Scheduling Networks of Queues: Heavy Traffic Analysis of a Two-Station Network With Controllable Inputs. *Operations Research* **38**, 1065-1078.

Wein, L. M. 1991. Scheduling Networks of Queues: Heavy Traffic Analysis of a Multistation Network With Controllable Inputs. To appear in *Operations Research.*

Wein, L. M. and J. Ou. 1991. The Impact of Processing Time Knowledge on Dynamic Job-Shop Scheduling. To appear in *Management Science.*

Whitt, W. 1971. Weak Convergence Theorems for Priority Queues: Preemptive-Resume Discipline. *J. Appl. Prob.* **8**, 74-94.

# Part II : Inspection for Circuit Board Assembly

## 1. Introduction

This part considers the problem of inspection in a circuit board assembly plant. With the increasing sophistication of manufacturing technology, boards of increasing complexity are being produced, which make testing an increasingly complex process. Currently, inspection costs can account for over half of the total manufacturing cost, and hence optimizing the utilization of inspection resources is a crucial task.

The assembly of circuit boards is performed in a single manufacturing stage, which is followed by several successive inspection stages. Assembled circuit boards have to be inspected for manufacturing defects as well as for defective components. The inspection process at each stage includes three different activities: testing, diagnosis and repair. Testing involves deciding whether to accept or reject a board; diagnosis is finding the defect on a rejected board; and repair consists of correcting that defect. Diagnosis is often the most difficult and time consuming task and the degree of difficulty depends on the board type and the type of test that is performed. The main focus of this paper is on testing, and henceforth repair refers to diagnosis as well as repair.

The attractiveness of a test depends on the test's cost, diagnostic power, coverage and measurement errors. The cost for performing a test on a particular board depends on the type of that board. The *diagnostic power* of a test is the amount of information that a test provides for diagnosis, and it strongly influences the time required and the cost for a repair. The *coverage* of a test is the extent to which defect can be detected. The error in the measurements on which the decision to accept or reject a board is based determines the prevalence of type I (false rejects) and type II (false accepts) errors. Defects are considered to be of different types, and the diagnostic power, coverage, and measurement errors of a test can be different for each type of defects. For example, if we look at Figure 2 we see that defect types linked to defective assembly (such as opens and shorts) are very well covered at the first inspection stage, whereas defective components are much harder to detect and some defects of this type can only be detected at the last inspection stage.

We consider two interrelated decisions. The first decision is to decide at which stage(s) to inspect a board; this problem is known in the literature as the *inspection allocation problem*. At the stages where inspection occurs, the *testing policy* decides whether to accept or reject each board based on the noisy measurements obtained from the test. The objective is to minimize the total expected cost of quality, which includes cost for testing, repair and defective items shipped to a customer. Since we assume that every defective board is repaired, no scrapping cost is included.

We first show that the determination of an optimal testing policy can be reduced to the problem of finding a point on the optimal tradeoff curve between type I and type II errors. The problem of finding an effective policy jointly for the allocation of inspection and for testing is then solved numerically. Two by-products of our analysis are of significant practical value. First, we can determine the cost reduction that would be achieved by reducing the measurement noise of the test equipment. This quantity can help circuit board manufacturers evaluate new test equipment, and can assist test equipment manufacturers focus their R&D efforts and market their equipment. Although inspection is necessary in the context of circuit board assembly, quality improvement efforts are also vital to the success of the company.

We can also derive the marginal benefit from reducing various types of defects. These quantities can be used to focus quality improvement efforts in an economic fashion.

This problem came to our attention while working with a Hewlett-Packard circuit board assembly facility. We include a case study of this facility, where our model is being applied. The case study includes a description of the facility, the methodology for gathering the data, which has been disguised, and the proposed policy. Our numerical results suggest that a 10-20% cost reduction can be achieved relative to this facility's current inspection policy by optimizing the allocation of inspection alone. The full model was not applied because we could not build estimates that were accurate enough to ensure a robust solution, but the insights gained from the model enable us to propose a testing policy that is robust and good. We are beginning to implement our policies and are not in a position to report on the cost reduction realized by the facility.

Many papers have been published on the optimal allocation of inspection in multistage serial systems. Early work on this problem (see for example Lindsay and Bishop [4], White [7]) assumed perfect inspection (i.e., no type I or type II errors). Later work allowed for imperfect inspection (Eppen and Hurst [2], Yum and McDowell [8] and Garcia-Diaz et al. [3]), where one determines the number of times that a test should be repeated. A survey of work published on inspection allocation can be found in Raz [5]. Recently, Villalobos et al. [6] studied a dynamic version of the same problem, where inspection of an item at a particular stage can depend on the result of the inspection of that item at previous stages. The existing literature on this topic has not addressed the presence of distinct defect types, the joint optimization of inspection allocation and testing or the application of such type of model to an industrial facility.

Section 2 presents a detailed formulation of the problem, which is then analyzed in Section 3. A case study is presented in Section 4, and conclusions are drawn in Section 5.

## 2. Problem Formulation

A typical flow chart of the assembly of circuit boards is presented in Figure 1. The main manufacturing stage is the circuit board assembly, where all the components are soldered onto the printed circuit boards. At the system assembly, the different boards are plugged into the final product. The *in circuit test* takes a measurement for each of the individual components soldered on the board. The *functional test* assesses the response of a board to simulated working conditions. At the *system test*, each board is tested as part of a complete system. Many variants of the configuration displayed in Figure 1 are possible. For example, there could be multiple levels of each test, or the system assembly step could be performed in several stages, and a test could be performed on each subassembly. On the other hand, some tests might not be present.

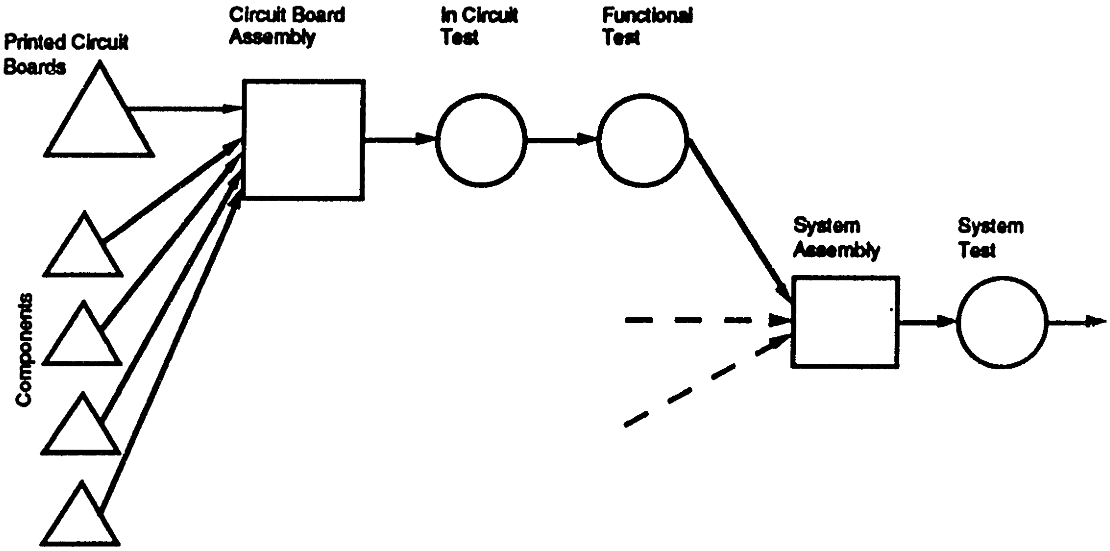


Figure 1 : Flow Chart of Circuit Board Assembly.

As mentioned earlier, an important characteristic of each test is its coverage. We will assume that for each type of defect, the successive tests that are performed on the circuit boards have an increasing coverage. Figure 2 illustrates this property, which we call *hierarchical test coverage*, for the example introduced above. Since they cover more defect types, successive tests tend to be more complex and more expensive to perform. However, as the complexity of the test increases, the precision of the information provided for diagnosis decreases. Consequently, diagnosis takes longer and has to be performed by more qualified personnel. The hierarchical assumption in test coverage holds for the circuit board assembly systems we have encountered. This assumption will also hold in manufacturing settings where additional work is performed between successive tests, and each test measures the cumulative functionality of the manufactured item.

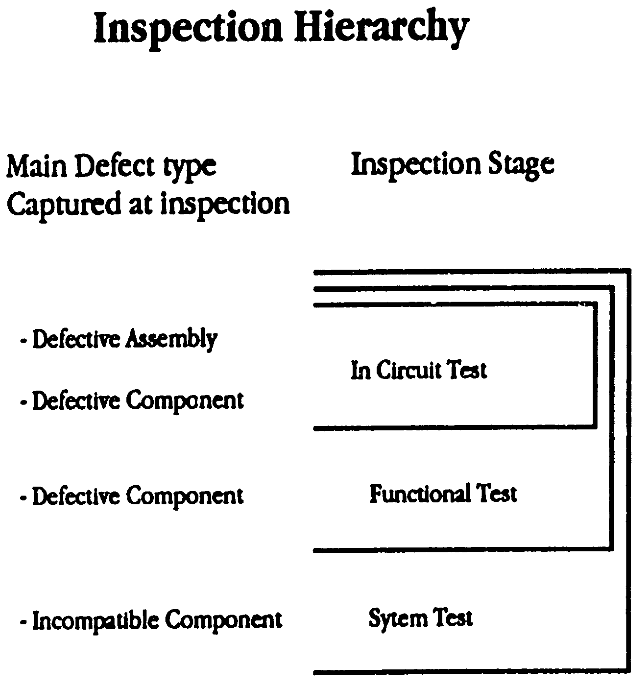


Figure 2 : Hierarchical test coverage.

In this section we will fromulate the problem for a single board in isolation from the other boards in the system. In the Subsection 3.3 we will discuss how this formulation can be used to solve the problem taking into account the dependencies between different board types.

A test consists of a series of upto a few thousand measurements. These measurements can relate to different components on a board, or to different aspects of the overall functional performance of a board. Most of these measurements are subject to some noise, and the measurement error can be separated into two parts: predictable error and random error. The predictable error is caused by the particular board type configuration. This error will not vary across boards of the same type, and can be predicted using historical data about past measurements. The random error is due to general environmental conditions and the precision limitation of the measurement device. The distribution of this error will depend on the type of measurement being taken, but is otherwise completely random. The utilization of statistical analysis to estimate the predictable error can be very useful; we encountered many cases (previously unbeknownst to the facility managers) where the predictable error was an order of magnitude higher than the random error. However, statistical analysis has to be performed on a continual basis; even small changes in the design or in the manufacturing process can drastically change the systematic error of a measurement.

The testing policy $T_n$ at stage $n$ specifies an interval for each measurement such that a board will be accepted if every measurement lies inside its interval and rejected otherwise. Because of the random error, it is impossible to completely eliminate type I and type II errors. Larger intervals will lead to the acceptance of more boards, which will reduce the number of good boards falsely rejected (type I errors) but increase the number of defective boards falsely accepted (type II errors). Similarly, smaller intervals will have the opposite effect.

For tests that have binary results, only one testing policy is possible. Examples of such situations are systemwide functional tests and tests for opens and shorts. For the purpose of generality, we will allow the model to have a choice of testing policies at each stage. However, the model easily accomodates the case where the set of possible testing policies at one or more stages is reduced to a single policy.

We assume there are $I$ different types of defects, but each measurement can detect only one type of defect. The number of measurements taken at stage $n$ that detect type $i$ defects is $K_{in}$. Let $v^m_{in_k}$ be the value obtained from the $k^{\text{th}}$ measurement for type $i$ defects at stage $n$. We assume that

$$v^m_{in_k} = v^t_{in_k} + \epsilon_{in_k},$$

where $v^t_{in_k}$ is the true value of the component being measured;

$\epsilon_{in_k}$ is the measurement noise, we assume that this noise is independent from $v^t_{in_k}$, and that the measurement errors are independent across different measurements. These assumption seems to be quite reasonable in practice.

Let $p_{in_k}(x)$ be the probability that the true value $v^t_{in_k}$ is inside $G_{in_k}$, the interval in which the true value should be for the proper functioning of the board. In mathematical terms $p_{in_k}(x)$ is defined as

$$p_{in_k}(x) = \Pr[v^t_{in_k} \in G_{in_k} \mid v^m_{in_k} = x].$$

We can express the function $p_{in_k}(x)$ in terms of the inputs of the problem, namely the tolerance interval $G_{in_k}$, the density function $\xi_{in_k}(x)$ of the distribution of the true values of the quantity measured, and the density function $e_{in_k}(x)$ of the distribution of the measurement error. It follows that

$$p_{in_k}(x) = \frac{\int_{G_{in_k}} e_{in_k}(x-y)\xi_{in_k}(y)\,dy}{\int_{-\infty}^{+\infty} e_{in_k}(x-y)\xi_{in_k}(y)\,dy}. \tag{1}$$

Let us assume that we will accept (that is decide that no defect was detected) the $k^{\text{th}}$ measurement for type $i$ defects at stage $n$ if it lies inside $[L, U]$. We define $\alpha_{in_k}(L, U)$ to be the probability that the measurement is outside the interval $[L, U]$ (i.e. the measurement is rejected) although the true value of the component is good (i.e. inside $G_{in_k}$). Similarly we define $\beta_{in_k}(L,U)$ as the probability that the measurement was accepted while the true value of the component is bad (i.e. outside $G_{in_k}$). Formally, we write

$$\alpha_{in_k} = \Pr\left[v^t_{in_k} \in G_{in_k} \text{ and } v^m_{in_k} \notin [L,U]\right]$$
$$\beta_{in_k} = \Pr\left[v^t_{in_k} \notin G_{in_k} \text{ and } v^m_{in_k} \in [L,U]\right].$$

Let $\alpha_{in}(T_n)$ be the expected number of false defects of type $i$ per board at stage $n$, and let $\beta_{in}(T_n)$ be the expected number of defects of type $i$ present on a board at stage $n$ that are not detected at that stage. We can write

$$\begin{aligned} \alpha_{in}(T_n) &= \sum_{k=1}^{K_{in}} \alpha_{in_k}(L_{in_k}, U_{in_k}), \\ \beta_{in}(T_n) &= \sum_{k=1}^{K_{in}} \beta_{in_k}(L_{in_k}, U_{in_k}), \end{aligned} \tag{2}$$

with $T_n = \{(L_{in_k}, U_{in_k}) \mid i = 1, \ldots, I; \text{ and } k = 1, \ldots, K_{in}\}$.

If we let $f_{in_k}(x)$ be the density function of the measured values, the probability $\alpha_{in_k}(L,U)$ can be expressed as

$$\begin{aligned} \alpha_{in_k}(L,U) &= \int_{-\infty}^{L} p_{in_k}(x) f_{in_k}(x)\,dx + \int_{U}^{\infty} p_{in_k}(x) f_{in_k}(x)\,dx \\ &= \int_{-\infty}^{\infty} p_{in_k}(x) f_{in_k}(x)\,dx - \int_{L}^{U} p_{in_k}(x) f_{in_k}(x)\,dx \\ &= \int_{G} \xi_{in_k}(x)\,dx - \int_{L}^{U} p_{in_k}(x) f_{in_k}(x)\,dx, \end{aligned} \tag{3}$$

where $\Pi_G$ is the proportion of good boards being tested.

Similarly, $\beta$ is

$$\begin{aligned} \beta_{in_k}(L,U) &= \int_{L}^{U} (1 - p_{in_k}(x)) f_{in_k}(x)\,dx \\ &= \int_{L}^{U} f_{in_k}(x)\,dx - \int_{L}^{U} p_{in_k}(x) f_{in_k}(x)\,dx \\ &= \int_{L}^{U} f_{in_k}(x)\,dx + \alpha_{in_k}(L,U) - \int_{G} \xi_{in_k}(x)\,dx. \end{aligned} \tag{4}$$

Puting together equations (2), (3) and (4) we get

$$\alpha_{in}(T_n) = \sum_{k=1}^{K_{in}} \left[\int_{-\infty}^{L_{in_k}} p_{in_k}(x) f_{in_k}(x)\,dx + \int_{U_{in_k}}^{\infty} p_{in_k}(x) f_k(x)\,dx\right] \tag{5}$$

and

$$\beta_{in}(T_n) = \sum_{k=1}^{K_{in}} \int_{L_{in_k}}^{U_{in_k}} (1 - p_{in_k}(x)) f_{in_k}(x)\, dx, \tag{6}$$

For technical reasons that will become apparent later, we also assume that $p_{in_k}(x)$ and $f_{in_k}(x)$ are continuous unimodal functions for all $i$, $n$ and $k$.

The objective is to minimize the total expected cost of quality, which includes the costs of testing, repair and defects leaving the plant. We consider a per unit testing cost $t_n$ at stage $n$, which typically includes operator time, test engineering, equipment cost and various overhead costs. No fixed cost is included in the model. The repair cost $r_{in}$ is the total cost incurred to diagnose and repair a defect of type $i$ on board at stage $n$. The same repair cost is incurred, whether the defect is a real defect or a false defect. We assume that all defective boards are repaired. The cost $f$ of a defect on a board that leaves the plant includes the cost of a field repair, the cost of the analysis and repair of the defective boards that come back to the plant, and a cost measuring the customer's loss of goodwill. The cost is per defect and not per defective system because if there are two or more defects on a system it is likely that those defects would appear to the customer at different moments and as a result each defect generates the same cost. This assumption also simplifies the model, and the results obtained would be very similar if a cost was incurred per defective system because of the very low number of defects leaving the plant make the appearance of multiple defects on the same system very unlikely.

As a consequence of the hierarchical test coverage assumption, more defects become detectable at each inspection stage. We will model this as if, on average, $d_{in}$ new defects of type $i$ appeared on the board at stage $n$. Let the average number of defects of type $i$ per board that leave stage $n$ be denoted $\delta_{in}$. Note that $\delta_{in}$ depends on the inspection policy at all earlier stages. Note also that $d_{in}$ and $\delta_{in}$ are expected values. The distribution of the random variables from which those expected values come does not matter because all the costs in the model are linear and we do not consider dynamic policies (i.e. policies where the decision to inspect depends upon what was observed earlier from the board). If inspection is performed at stage $n$ then the expected cost of testing and repair at that stage is

$$t_n + \sum_{i=1}^{I}\Big((\delta_{i,n-1} + d_{in} - \beta_{in}(T_n))r_{in} + \alpha_{in}(T_n)r_{in}\Big), \tag{7}$$

if $T_n$ is the testing policy used, and the expected number of defects of type $i$ per board leaving stage $n$

$$\delta_{in} = \beta_{in}.$$

On the other hand, if no inspection is performed at stage $n$, then no cost is incurred at that stage. The expected number of type $i$ defects leaving stage $n$ in this case is

$$\delta_{in} = (\delta_{i,n-1} + d_{in}).$$

As a result, the problem is to decide whether to inspect at each stage and if so what testing policy $T_n$ to use – if such a choice exists –, such as to minimize the total inspection cost (including the cost of defects leaving the plant).

## 3. Analysis

In this section, we solve the problem formulated in Section 2. In the Subsection 3.1, the testing problem is reduced to finding a point on the optimal tradeoff curve between type I and type II errors. In the Subsection 3.2, the problem itself is numerically solved. Different extensions of the basic problem are described in the last three subsections.

### 3.1. The Testing Problem

The purpose of this subsection is to find a set of testing policies that are optimal with respect to the tradeoff between type I and type II errors in the sense of Pareto optimality (i.e., it is impossible to reduce one type of error without increasing the other type of error). This will enable us to restrict our attention to policies in that set when searching for an optimal testing policy.

The minimization of the inspection cost in expression (7) can be written as a dynamic programming equation

$$\begin{aligned} J_n(\rho_1,\rho_2,\dots,\rho_I) = \mathrm{Min}\Bigg[ & J_{n+1}(\rho_{1,n-1}+d_{1n},\rho_{2,n-1}+d_{2n},\dots,\rho_{I,n-1}+d_{In}), \\ & t_n + \min_{T_n}\Big[\sum_{i=1}^{I}\Big((\rho_{i,n-1}+d_{in}-\beta_{in}(T_n))r_{in}+\alpha_{in}(T_n)r_{in}\Big) \\ & + J_{n+1}(\beta_{1n}(T_n),\beta_{2n}(T_n),\dots,\beta_{In}(T_n))\Big]\Bigg], \end{aligned} \tag{8}$$

where $J_n(\delta_{1n},\delta_{2n},\dots,\delta_{In})$ is the total inspection cost from stage $n$ until exiting the plant.

We can rewrite the second minimization in equation (8) in a more concise manner as follows

$$\min_{T_n}\Big\{h_n\Big(\alpha_{1n}(T_n),\alpha_{2n}(T_n),\dots,\alpha_{In}(T_n)\Big)+g_n\Big(\beta_{1n}(T_n),\beta_{2n}(T_n),\dots,\beta_{In}(T_n)\Big)\Big\} \tag{9}$$

where $h_n(x_1,x_2,\dots,x_I)=\sum_{i=1}^{I}x_i r_{in}$

and $g_n(x_1,x_2,\dots,x_I)=J_{n+1}((x_1,x_2,\dots,x_I)+\sum_{i=1}^{I}(\delta_{i,n-1}+d_{in}-x_i)r_{in}$.

If $\alpha_{in}(T_n)$ and $\beta_{in}(T_n)$ in expression (9) are replaced with the expressions obtained in equations (5)–(6), then the derivative of this function with respect to the upper acceptance limit $U_{in_k}$ is

$$\begin{aligned} -\frac{\partial h_n}{\partial x_i}&(\alpha_{1n},\alpha_{2n},\dots,\alpha_{In})p_{in_k}(U_{in_k})f_{in_k}(U_{in_k}) \\ &+\frac{\partial g_n}{\partial x_i}(\beta_{1n},\beta_{2n},\dots,\beta_{In})\,(1-p_{in_k}(U_{in_k}))\,f_{in_k}(U_{in_k}), \end{aligned}$$

where in order to simplify notation $\alpha_{in}$ stands for $\alpha_{in}(T_n)$ and $\beta_{in}$ stands for $\beta_{in}(T_n)$. This expression will be equal to zero if

$$p_{in_k}(U_{in_k})=\frac{\frac{\partial g_n}{\partial x_i}(\beta_{1n},\beta_{2n},\dots,\beta_{In})}{\frac{\partial h_n}{\partial x_i}(\alpha_{1n},\alpha_{2n},\dots,\alpha_{In})+\frac{\partial g_n}{\partial x_i}(\beta_{1n},\beta_{2n},\dots,\beta_{In})}. \tag{10}$$

The second derivative of the objective function (9) will be positive if $f'_{in_k}(U_k) < 0$ and $p'_{in_k}(U_k) < 0$. This is what one would expect, since the upper limit is at a point where the frequency of measurement decreases as well as the probability that a measurement corresponds to a valid board.

The same argument can be used to show that the optimal lower acceptance limit is such that

$$p_{in_k}(L_{in_k}) = \frac{\frac{\partial g_n}{\partial x_i}(\beta_{1n}, \beta_{2n}, \dots ,\beta_{In})}{\frac{\partial h_n}{\partial x_i}(\alpha_{1n}, \alpha_{2n}, \dots ,\alpha_{In}) + \frac{\partial g_n}{\partial x_i}(\beta_{1n}, \beta_{2n}, \dots ,\beta_{In})}. \tag{11}$$

The second derivative of the objective function (9) will be positive if $f'_{in_k}(L_k) > 0$ and $p'_{in_k}(L_k) > 0$. Again, this is consistent with our intuition, since the lower limit is at a point where the frequency of measurements increases, as well as the probability that a measurement corresponds to a valid component.

From the definition of $h_n$ and $g_n$, we find that $\frac{\partial h_n}{\partial x_i}(\alpha_{1n}, \alpha_{2n}, \dots ,\alpha_{In}) = r_{in}$ and $\frac{\partial g_n}{\partial x_i}(\beta_{1n}, \beta_{2n}, \dots ,\beta_{In}) = \frac{\partial J_{n+1}}{\partial \rho_i}(\beta_{1n}, \beta_{2n}, \dots ,\beta_{In}) - r_{in}$. Equations (8) and (9) now become

$$p_k(L_k) = p_k(U_k) = \frac{r_{in}}{\frac{\partial J_{n+1}}{\partial \rho_i}(\beta_{1n}, \beta_{2n}, \dots ,\beta_{In})}. \tag{12}$$

This expression has an intuitive meaning : the probability that a component is good at the acceptance and rejection cutoff points should be equal to the marginal cost of a false defect divided by the marginal cost of a false accept. Notice that the right hand side is a constant for $i = 1, \dots , I$ and $k = 1, \dots ,K_{in}$, which we denote by $C$.

The existence of lower and upper limits $L_k$ and $U_k$ satisfying the condition (12) is guaranteed by the assumption stated in Section 2 about the continuity and unimodality of $f_k(x)$ and $p_k(x)$. We will not pursue weaker necessary and sufficient conditions for the existence of these optimal upper and lower bounds, since in our case study these functions will be approximated by Gaussian density functions that satisfy the necessary conditions.

If tests at previous stages attempt to find the same type of defects as the $k^{\text{th}}$ measurement, then it is likely that the distribution of the quantity measured, $\xi_k$, will depend on the inspection policy followed at previous stages. Consequently, $p_k(x)$ and hence the set of Pareto optimal testing policies will depend on the inspection policy at the previous stages.

If $p_{in_k}(x)$ is independent of the inspection policy at previous stages, equations (5), (6) and (12) can then be used to compute in advance all possible optimal testing policies and build the functions $\alpha_{in}(C)$ and $\beta_{in}(C)$. As a result, the minimization in the dynamic programming recursion does not have to explicitly consider all possible testing policies, but only these two functions. Figure 3 shows the tradeoff curve of type I errors ($\alpha_{in}(C)$) versus type II errors ($\beta_{in}(C)$) obtained by letting $C$ vary, for a particular defect type encountered in our case study.

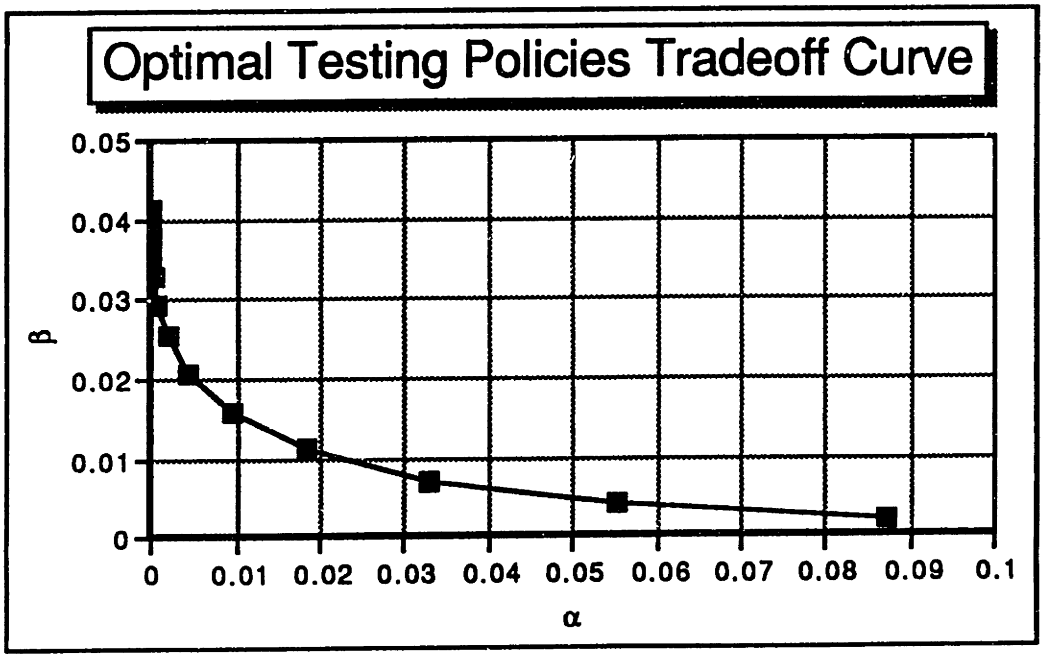


Figure 3 : $\alpha_{in}(C)$ versus $\beta_{in}(C)$.

If the measurement noise and the value of the component being measured are both normally distributed, then $p_k(x)$ has a relatively simple form (all subscripts $_{in_k}$ are being dropped for better readability)

$$p(x) =$$

$$\frac{1}{2}\Big(\mathrm{erf}\big(\frac{(G_u-\mu_\xi)\sigma_e^2+(G_u+\mu_e-x)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2+\sigma_\xi^2)}}\big)-\mathrm{erf}\big(\frac{(G_l-\mu_\xi)\sigma_e^2+(G_l+\mu_e-x)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2+\sigma_\xi^2)}}\big)\Big) \tag{13}$$

where:

$\mu_e$ and $\sigma_e$ are the expected value and the standard deviation, respectively, of the measurement noise;

$\mu_\xi$ and $\sigma_\xi$ are the expected value and the standard deviation, respectively, of the true values of the component being measured; we will also refer to $\mu_\xi$ as the nominal value of the component;

$G_l$ and $G_u$ are the lower and upper limits respectively of the interval $G$ such that the measured component is good if its true value lies inside it;

erf is the error function, $\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}\,dt$, a plot of this function appears in Figure 4.

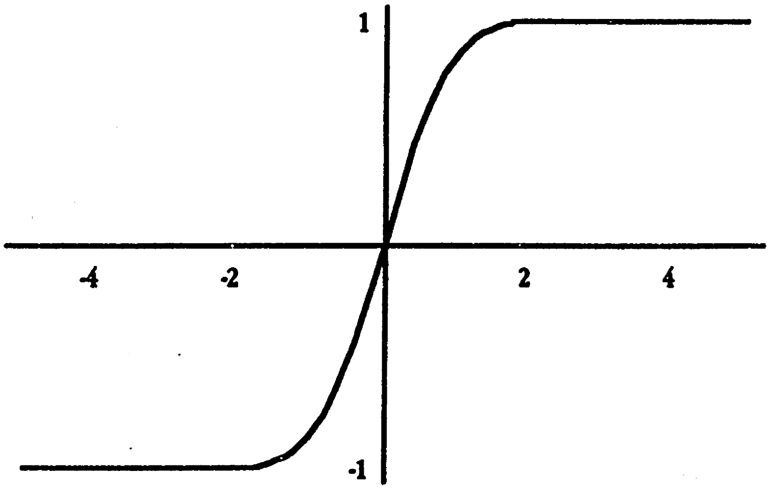


Figure 4 : the error function.

Moreover, if the interval $[G_l, G_u]$ can be written as $[\mu_\xi - s, \mu_\xi + s]$, in other words the interval is centered around the nominal value, then the expression for $p(x)$ becomes

$$p(x) = \frac{1}{2}\Big(\mathrm{erf}\big(\frac{s\sigma_e^2 + (\mu_\xi + \mu_e + s - x)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big) - \mathrm{erf}\big(\frac{-s\sigma_e^2 + (\mu_\xi + \mu_e - s - x)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big)\Big)$$

In this case, if we let $x = \mu_\xi + \mu_e + z$ we have

$$p(\mu_\xi + \mu_e + z) = \frac{1}{2}\Big(\mathrm{erf}\big(\frac{s\sigma_e^2 + (s - z)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big) - \mathrm{erf}\big(\frac{-s\sigma_e^2 + (-s - z)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big)\Big),$$

because $\mathrm{erf}(-x) = -\mathrm{erf}(x)$ we have that

$$\begin{aligned} p(\mu_\xi + \mu_e + z) &= \frac{1}{2}\Big(-\mathrm{erf}\big(\frac{-s\sigma_e^2 + (-s + z)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big) + \mathrm{erf}\big(\frac{s\sigma_e^2 + (s + z)\sigma_\xi^2}{\sigma_e\sigma_\xi\sqrt{2(\sigma_e^2 + \sigma_\xi^2)}}\big)\Big) \\ &= p(\mu_\xi + \mu_e - z). \end{aligned}$$

This implies that if the measurement noise and component values are normally distributed, and the tolerance interval for good components is centered around the nominal value, then the lower and upper limits to accept the component should be centered at the nominal value plus the expected value of the measurement noise. The expected value of the measurement noise is what was referred to earlier as the predictable component of the measurement noise. Hence, if we know the expected value of the measurement error, then we can immediately derive the entire set of Pareto optimal testing policies (as long as we assume normality and a centered tolerance interval, which hold in most practical cases). As will be seen in the case study, the possibility to identify a priori the set of optimal testing policies can be very useful, especially when some data cannot be estimated with accuracy.

## 3.2. Finding a Global Inspection Policy

The problem is difficult because the optimal testing policy at any stage will depend on the inspection policy at all the previous stages as well as on the inspection policy at all the subsequent stages. The optimal testing policy depends on the inspection policy at subsequent stages because the marginal cost of a defect leaving the current stage depends on it. The optimal testing policy depends on the inspection policy at previous stages because the distribution of true values of the components being measured depends on it. Two types of solution techniques can be applied – either a backward method such as the classic dynamic programming algorithm, or a forward method such as an exhaustive search. Finding the optimal policy using either of these two techniques would be prohibitively tedious, consequently we are going to develop a procedure to find a good solution without any guarantee of optimality.

In order to apply dynamic programming we could assume that the dependency of the functions $p_k(x)$ on the inspection policies at the previous stages can be captured entirely by the vector $\delta_{n-1}$ of defects leaving the previous stage. This assumption would enable us to build a model that would be a close representation of the real situation. In this case, one possible way to find an optimal solution is to numerically solve the dynamic programming equation (5). This, however, would require a discretization of the $I$-dimensional state space of incoming defects, which involves a lot of computations.

Hence, the solution technique proposed here is based on exhaustive search, and the utilization of equation (12) to fine-tune the testing policy. The typical size of type of problem considered here is not likely to be very large and consequently an exhaustive search can be performed very quickly. Also, this technique can easily take advantage of the fact that not all defect types are influenced by the testing policies.

## The exhaustive search

Let $\pi_n$ represent the inspection policy at stage $n$, and let $\Pi_m = (\pi_1, \pi_2, \ldots, \pi_m)$ denote the inspection policy for the line up to stage $m$. The policy $\pi_n$ either states that no inspection is performed or else it specifies the testing policy at that stage.

Of course, $\Pi_N$ is a complete inspection policy for the entire line. The cost $c_{\Pi_m}$ of inspection policy $\Pi_m = (\pi_1, \ldots, \pi_{m-1}, \pi_m)$ can be calculated from the cost $c_{\Pi_{m-1}}$ of inspection policy $\Pi_{m-1} = (\pi_1, \ldots, \pi_{m-1})$ as follows

$$c_{\Pi_m} = \begin{cases} c_{\Pi_{m-1}} + t_n + \sum_{i=1}^{I} \Big[\alpha_{i,\pi_m} + (d_{i,\Pi_{m-1}} + \delta_{im} - \beta_{i,\pi_m})\Big] r_{im} & \text{if } \pi_m = \text{inspection;} \\ c_{\Pi_{m-1}} & \text{if } \pi_m = \text{no inspection.} \end{cases}$$

The number of defects of type $i$, $d_{i,\Pi_m}$, that leave stage $m$ under policy $\Pi_m$ is then

$$d_{i,\Pi_m} = \begin{cases} \beta_{i,\pi_m} & \text{if } \pi_m = \text{inspection;} \\ d_{i,\Pi_{m-1}} + \delta_{im} & \text{if } \pi_m = \text{no inspection.} \end{cases} \qquad \text{for all } i.$$

The procedure will end by choosing the policy $\Pi_N$ that minimizes the total inspection cost over the entire line that is given by $c_{\Pi_N} + f \sum_{i=1}^{I} d_{i,\Pi_N}$.

## Fine-tuning the testing policy

Once the inspection allocation policy is fixed, it is possible to use equation (12) to find the optimal testing policy at each stage. In particular, if the inspection policy for all stages following $n$ is fixed, then the derivative of $J_{n+1}$ with respect to its $i^{\text{th}}$ argument is a constant for each $i$. This constant is the expected cost of having one more defect of type $i$ leaving stage $n$ and will be calculated below. Let $q_{im}$ be the probability that a defect of type $i$ arriving at stage $m$ is detected at stage $m$. If we assume that all defects of type $i$ reaching stage $m$ are equally likely to be detected, then

$$q_{im} = \frac{\delta_{i,m-1} + d_{im} - \delta_{im}}{\delta_{i,m-1} + d_{im}}$$

The probability that a defect of type $i$ leaving stage $n$ is repaired at stage $m > n$ is

$$\Big(\prod_{l=n+1}^{m-1} q_{il}\Big)(1 - q_{im}),$$

As a result, the expected cost that this defect will create for the system is

$$\frac{\partial J_{n+1}}{\partial \rho_i}(\cdot) = \sum_{m=n+1}^{N} \Big[\Big(\prod_{l=n+1}^{m-1} q_{il}\Big)(1 - q_{im}) r_{im}\Big] + \Big(\prod_{l=n+1}^{N} q_{il}\Big) f. \tag{14}$$

Here again the fine tuning can be performed in a forward manner to try to use the most accurate estimates of $\alpha_{in}$ and $\beta_{in}$ at each stage, or in a backward manner to try to use the most accurate estimates of $J_{n,i}$ at each stage.

### Discussion

Since this procedure works in a 'forward' mode, that is, from stage 1 to $N$, it can be carried out without making any assumptions about the dependence of the functions $p_k$ on the inspection strategy at previous stages. Nevertheless, if there are many tests for which this dependency occurs, it will be time consuming to generate a different set of testing policies for each scenario. Note that the dynamic programming approach would probably require even more testing policies to be generated during the optimization process over the entire discretized space. The author could not find any way to overcome this potential problem.

In the exhaustive search, we will only consider a small subset (about 5 policies per test) of the continuous range of Pareto optimal testing policies. Many tests in circuit board assembly are non-parametric, and hence only a small fraction of the total number of tests will have their testing policies restricted to a subset. Furthermore, the Pareto optimal curve for each parametric test (see Figure 3) is smooth and can be quite accurately represented by a small set of points. As a result, the cost reduction achieved by fine-tuning the testing policy is likely to be at least an order of magnitude lower than the cost reduction obtained from the optimal allocation policy with the reduced set of testing policies. Therefore, the two step procedure will yield the optimal policy or at least a policy very close to optimality. The fine-tuning also has another use. As the fabrication process evolves and the defect rates change, the testing policy could be easily changed, whereas a change in the inspection allocation would be quite disruptive. As a result, it makes sense to try to adjust the testing policies relatively frequently to get the best out of the allocation policy that will remain fixed over longer periods of time.

It must be noted that this fine-tuning procedure does not guarantee an optimal overall policy. The only way to find a solution that is guaranteed to be very close to the optimal solution would be to perform an exhaustive search with a large set of testing policies for each stage that has a continuous set of testing policies. Unfortunately quickly becomes computationally intractable as the number of testing policies considered increases. Nevertheless, it is likely that the two step approach will give a near optimal solution, when the conditions mentioned earlier are met. One must also keep in mind that it is fruitless to seek a solution whose precision is greater than that of the data that can be gathered.

## 3.3 Dependencies Between Different Board Types

Until now, we have been considering only one type of circuit board. If different boards were completely independent then all of our results would carry over. However, some tests actually measure several boards together, in which case either all the boards are tested or none are tested. Thus, the inspection allocation decisions taken independently might yield an infeasible solution.For shared tests that are non-parametric, this can be dealt with by employing a Lagrangian relaxation approach, where the total cost of the test is split among the different boards such that the solutions found independently coincide. To illustrate this, let us suppose the total cost of a test that covers two boards is $t$. We run the model for the boards independently with the cost $t$ split between the two boards arbitrarily. For example the cost can be split proportionally to the comlexity of the two boards. Then, if the decisions for the two boards agree we have a feasible and optimal solution. If the solution do not agree, we modify the allocation of $t$ between the two boards such as to increase the part that is allocated to the board for which the solution includes the shared test and consequently decrease the part that is allocated to for which the solution does not include the shared test. For example the solution for board 1 uses the test but the solution for board two does not, to board 1 and decrease the part that is allocated to board 2. The problem is then solved again for both boards. And if the solutions still do not agree we iterate with the cost reallocation procedure. Many choices are possible to search for an allocation of costs. Since by lack of data this procedure could not be applied in our case study we are not in a position to report on the efficiency of any particular algorithm. Note that this approach becomes almost impossible to use with a parametrical test. In the plant studied in the case study, the tests that were measuring more than one board were the more comprehensive system tests which are non-parametric tests.

Another cause of dependencies across boards is congestion. If too many boards are to be tested at the same stage, then the testing load may be undesirably high at that stage. Moreover, if many of these boards are not tested at the previous stages, then many repairs will be required at that stage, which leads to additional work. A Lagrangian relaxation method, where the price of inspection at the congested stages would be increased untill the load of the congested state reaches a more desirable level, could be used to solve the problem. Moreover, the price attributed to a congested stage by the Lagrangian routine could be used for inspection capacity decisions.

### 3.4. Improved Testing Equipment

Our model explicitly takes into account the effect of measurement noise, which enables us to determine the value of testing equipment with lower measurement noise. This question is very important, since testing equipment is extremely expensive and it is not clear, a priori, how a reduction in the measurement errors will affect the overall cost of inspection.

To evaluate the value of improved testing equipment in the context of our model, the type I versus type II tradeoff curve for the new equipment is computed, and subsequently the two step algorithm outlined in Subsection 3.2 is carried out using this new curve. The optimal total quality cost for the new equipment can then be compared with the corresponding value for the old equipment.

An experiment was performed in order to illustrate the magnitude of the savings achievable with more precise equipment. We assumed that a hypothetical new tester would have a measurement noise whose standard deviation was half of that of the current tester. Figure 5 shows the tradeoff curve for the new and original test equipment. This figure shows that both types of errors could simultaneously be divided by at least two using the new equipment. If we assume similar error reductions for all components on the board, then the reduction in total quality cost is on the order of 5 percent.

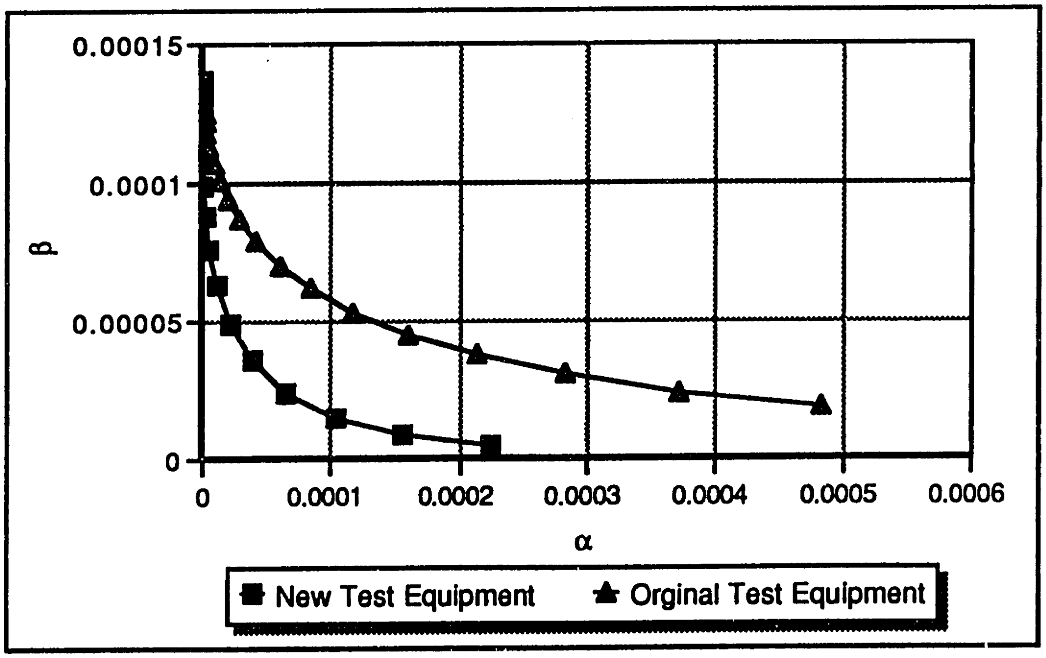


Figure 5 : Improvement of testing errors.

## 3.5. Allocation of Quality Improvement Efforts

Engineering resources to work on quality improvements are limited, and consequently it is important to dedicate resources to projects where an improvement would have the maximum impact. In a slightly different context, Albin and Friedman [1] show that the choice of the most valuable project is not always straightforward. In particular, their paper shows that the traditional Pareto chart is not an adequate tool when defects are clustered. In the problem considered here, it is also possible that a Pareto Chart would be misleading because different types of defects are detected in different proportions at different stages and thus the cost of different types of defects can be very different.

Equation (14) gives the expected cost that a defect of type $i$ will generate for the system if it becomes detectable at stage $n$. Recall that this quantity is a constant under any fixed inspection policy. Hence, the total system cost due to type $i$ defects on a given board type is

$$q_i = \sum_{n=1}^{N} \frac{\partial J_{n+1}}{\partial \rho_i}(\cdot) d_{in} \qquad (15).$$

By summing the cost of a particular defect type for each board type we can compute the overall effect of each defect type. Equation (15) can also be used to assess incremental defect reduction,

$$\frac{q_i}{\sum_{n=1}^{N} d_{in}}$$

is the marginal cost reduction obtained from reducing type $i$ defects. The marginal cost reduction obtained from reducing type $i$ defects at stage $n$ is

$$\frac{\partial q_i}{\partial d_{in}} = \frac{\partial J_{n+1}}{\partial \rho_i}(\cdot)$$

## 4. Case Study

We now describe how the model was applied at the Hewlett Packard facility. This plant has three inspection stages as shown in Figure 1. Approximately 50 different board types are manufactured at this facility, and these boards go into various models of a single line of final products. The number of boards of each type produced is relatively low, ranging from 1,000 to 10,000 per year. Hence, tests must be flexible in order to accommodate many different board types. The nature of the final product requires very strict tolerances on the circuit boards and on their components, which partially explains why this facility currently tests all boards at all stages. Figure 2 illustrates roughly the structure of the tests performed. There is currently no systematic procedure for determining testing policies at this facility, the testing policy is highly dependent on the particular engineer in charge of it. It is also estimated that at this facility the total cost of inspection represents about half of the total manufacturing cost.

The first part of this section explains how the relevant data was collected and the second part describes our results.

### 4.1. Data collection

Three different types of data have to be gathered in order to use this model. The first type of data concerns the cost parameters of the model. The second concerns the occurrence of defects, and the coverage and reliability of each inspection stage. The last and perhaps most difficult type of data to gather pertains to the testing process. We gouped the many different categories of defect that are used internally into 7 broader types. This grouping was done such as to put together the categories that are similar for testing purposes, they might differ for diagnosis purposes.

Testing and repair are two activities that are closely linked, since they are performed by the same people in the same work area. Their cost includes technician time, floor space, equipment and overhead for management time. To allocate cost between these two activities, we used estimates of the time that different engineers and technicians spend on each of these activities. Both of these costs vary for each board type; in particular, they will depend on the complexity and the design of a board. These costs are proportional to the overall volume of inspection that takes place at a stage, and therefore these costs are proportional to the number of boards processed.

In the case study the procedure to estimate these cost parameters varied depending on how detailed the available data was. For some stages we could get estimates of the each of the different overhead costs (engineering, equipment, floor space, etc.) individually. In these cases we split each of these costs according to how much could be allocated either to testing (and testing maintenance) or repair (and diagnosis). The total cost for each of these activities was then divided by the total time the boards underwent testing and repair respectively. This gives a rate of cost for each activity. Multiplying the average time it took to test or repair a particular board type gives the estimate for the corresponding cost. In the cases where we could not obtain detailed overhead costs we multiplied the average time to test and repair a particular board type by a flat overhead rate that includes all the above costs to obtain estimates of the test and repair costs repespectively. In the case study the repair cost was considered to be identical for each type of defect type for a particular board type. This is an approximation of reality, but we did not have the means to gather enough data to estimate these costs for each individual defect type. The test and repair cost did however vary widely across board types.

The cost of a defect on a board leaving the plant includes the cost of an on site repair, analysis and repair of the defective board at the plant and the quality department's estimate of the cost of lost goodwill. This goodwill cost is a direct function of the competitive environment in which the company is operating. In a very competitive market a customer is much more likely to switch to another vendor if he is even slightly disappointed with the product. On the other hand if the product enjoys some kind of monopoly (or partial monopoly), then the customer will probably be more willing to tolerate small deficiencies of the product. In the case study we tried to estimate the expected number of lost sales that a defective unit might induce. The selling price of a unit times this number of lost sales was then used as the goodwill cost.

To derive estimates for the occurrence of defects and the coverage and reliability of each inspection stage, we used historical data about the total number of defects per board of each type detected at the different stages of inspection, $\phi_{in}$. Then the engineers in charge of each test were asked to estimate the proportion $a_{in}$ of defects of each type that they felt their test should detect, and the proportion $b_{in}$ of defects of each type detected by the test that were actually good boards (false defects). Some of these estimates were based on experiments, whereas other were based on the experience of the engineers with the process.

The proportion of false defects together with the observed number of defects enabled us to compute the number of real defects of each type detected at each stage $\gamma_{in} = \phi_{in} b_{in}$. The probability of type I errors (false rejects) at any stage is then $\alpha_{in} = \phi_{in} - \gamma_{in}$. From the total number of real defects of each type present on a board and the proportion of those defects that should be detected at each stage, we found the number of defects per board that become detectable at each stage $n$,

$$d_{in} = \sum_{m=1}^{N} \psi_{im} + \eta_i - \sum_{m=1}^{n-1} d_{im} \text{ for } i = 1, \ldots ,I; \text{ and } n = 1, \ldots ,N,$$

where $\eta_i$ is the number of failures per board that occured during the waranty period of the equipment (a failure during the waranty period was considered to be caused by an undetected defect). Finally, the probability of type II errors at each stage was derived by comparing this last quantity with the actual number of real defects detected at each stage $\beta_{in} = \gamma_{in} - d_{in}$.

In order to derive an optimal testing policy, we also need the input data $G$, $e(x)$ and $\xi(x)$ for each quantity measured at each stage. Recall that $G$ is the interval in which the true value of the quantity measured should be. It is often very hard to know what the exact limit values are on a component that will ensure the proper functioning of the board. This problem was avoided here by using the specifications to which the component was bought. The justification for this is that even if a component is bought with tighter specifications than actually needed, a component that does not meet these specifications signals some kind of abnormality, such as physical damage or a poor soldering, that may cause a real defect after the equipment has been utilized for some time.

To estimate the distribution of the measurement noise $e(x)$ and the distribution of the true value of the quantity measured $\xi(x)$, we only have the distribution of the measured values at our disposal. This measured value, which is recorded by the testing device, is the sum of the true value of the component and the measurement noise. Unfortunately, neither of these two quantities can be estimated independently. The true values are almost impossible to measure, since most components used at this facility are surface mount components, which are extremely small and fragile. Estimating the measurement noise is also a delicate task since the distribution of this noise will depend on many things, such as the type of component that is being measured, how the measurement is *guarded*† and the topology of the board. As

† Guarding is the technique used to try to isolate a component from the rest of the circuit board.

a result, the measurement noise can only be determined via experiments for each different board type.

A preliminary experiment was run to study the different sources of measurement noise at the in circuit test level. At the facility studied several in circuit testers, also called testheads were used in parallel. The boards are tested on the first available testhead. Consequently, we also wanted to find out what part of the measurement noise was attributable to variations between different testheads. The experiment that was performed repeated each measurement (there were 79 different measurement) $K = 10$ times consecutively on $H = 3$ different testheads for $B = 10$ different boards. This constitutes a little bit less than 24,000 data points. The measurement noise was modeled by expressing the measurements $y_{bhk}$ as

$$y_{bhk} = \mu + \tau_b + \theta_h + \psi_{bh} + \epsilon_{bhk} \qquad b = 1, \ldots, B \quad h = 1, \ldots, H \quad k = 1, \ldots, K; \tag{16}$$

where:

$\mu$ is the reference value for the component being measured;

$\tau_b$ is the average deviation of the measurements taken on the $b^{\text{th}}$ board, aand has zero mean and standard deviation $\sigma_\tau$;

$\theta_h$ is the average deviation of measurements taken on the $h^{\text{th}}$ head, and has zero mean and standard deviation $\sigma_\theta$;

$\psi_{bh}$ is the average deviation of measurements taken on the $h^{\text{th}}$ head and the $b^{\text{th}}$ board, and has zero mean and standard deviation $\sigma_\psi$;

$\epsilon_{bhk}$ is the residual variation of a measurement that cannot be explained by either the testhead or the board or the interaction between the testhead and the board, $\epsilon_{bhk}$ has zero mean and standard deviation $\sigma$.

Note that by using this model we make the implicit assumption that the residual noise $\epsilon$ has the same variance on all testheads. The data indicated that this assumption did not hold for all components in practice, which suggests that the three testheads were not equally calibrated, when the data was gathered. Without this assumption, the estimation of the model parameters would have been greatly complicated. Moreover, by only gathering data from three testheads, a good estimation of the distribution of the variance of the residual noise across different testheads was not possible.

We estimate $\mu$ by $\bar{y}$, which is the average of all measurements taken.

The variance of $\epsilon$ is estimated by

$$\hat{\sigma}^2 = \frac{\sum_{b=1}^{B} \sum_{h=1}^{H} \sum_{k=1}^{K} (y_{bhk} - \bar{y}_{bh})^2}{BH(K-1)}.$$

This is the variance of successive measurements of the very same component on the same testhead, and represents the precision limitation of the tester for the component. The variance of $\psi$ is estimated by

$$\hat{\sigma}_{\psi}^2 = \frac{\sum_{b=1}^{B} \sum_{h=1}^{H} (\bar{y}_{bh} - \bar{y}_b - \bar{y}_h - \bar{y})}{(B-1)(H-1)} - \frac{\sigma^2}{K},$$

and the variance of $\theta$ is estimated by

$$\hat{\sigma}_{\theta}^2 = \frac{\sum_{h=1}^{H} (\bar{y}_h - \bar{y})^2}{H-1} - \frac{\sigma_{\psi}^2}{B} - \frac{\sigma^2}{BK}.$$

These two variances represent the amount of variation that is linked to the variations between testheads. One might think that a fixed effects model would be more appropriate for the testheads, since the facility uses only a fixed number of different testheads. But the variation between testheads is evolving over time as a result of usage, maintenance, calibration, etc. Consequently, the variation between testheads is a continuous random variable and the random effects model is appropriate.

The average of all measurements taken on the $b^{\text{th}}$ board equals $\mu + \tau_b$, and $\tau_b$ can be considered to be the variation in the measured values associated with the individual components on the $b^{\text{th}}$ board. The variance of $\tau$ can be estimated by

$$\hat{\sigma}_\tau^2 = \frac{\sum_{b=1}^{B}(\bar{y}_b - \bar{y})^2}{B-1} - \frac{\sigma_\psi^2}{H} - \frac{\sigma^2}{HK}.$$

Using the estimated parameters from the statistical model in equation (16), the parameters of the distributions of the measurement noise and the component values can be estimated. We assume that $\mu_\xi$, the nominal value of the component is known. Then we have

$$\begin{aligned} \hat{\mu}_e &= \bar{y} - \mu_\xi \\ \hat{\sigma}_e^2 &= \hat{\sigma}^2 + \hat{\sigma}_\psi^2 + \hat{\sigma}_\theta^2 \qquad (17) \\ \hat{\sigma}_\xi^2 &= \hat{\sigma}_\tau^2 \end{aligned}$$

Appendix A contains tables and figures for each type of component (resistors, capacitors, transistors, diodes and inductors) with the estimates that were derived from this experiment. We see that for all the inductors in this sample the estimated variance of the measurement noise was higher than the estimated variance of the true values of these components. This implies that the measurements taken cannot distinguish good and defective components. For all the other components we distinguish a clear pattern where most of the components have an estimated noise variance much smaller than the estimated variance of their true values. Those components can tested with good accuracy. A small minority of components fall into the opposite category, their estimated measurement noise is almost as large or larger than the estimated variance of their true values, in which case the test is not reliable. This has important practical implications, it means that some components should not be tested (or alternatively a new test should be devised for them).

An important question for the design of future experiments is whether the variance of the noise associated with the different testheads can be predicted. It is much easier to run experiments on a single testhead than on several testheads, because in the latter case, experiments would be much more difficult to schedule in a production process. To address this question, a regression was run to try to predict $\sigma_e$ from $\sigma$ and the absolute value of $\mu_e$. The coefficients of correlation were consistently high, for example, $r^2 = 0.92$ for the resistors and $r^2 = 0.98$ for the capacitors .

To check whether this is also valid for other boards, data was collected from another board during production. For that board, all measured values of all components on approximately 80 boards were recorded. This data enabled us to estimate the total variation of the measurements $\sigma_e + \sigma_\xi$. Appendix B shows charts of the distribution of the total variance for the different components of each type on the board. We wanted to know if this distribution was consistent with what was observed on the experimental board. It can be seen, by comparing the graphs of Appendix B with those of Appendix A, that the results are indeed consistent. For example, most resistors have a total standard deviation around 0.3% in both Appendix A and B; most capacitors have a total standard deviation between 2 and 4% in both cases. These results are very encouraging, but a more thorough investigation of this issue is still needed. A complete experiment similar to the one run on the first board should be repeated on some other boards before reaching a definitive conclusion on this point.

If we assume that the distributions of the measurement noise and of the values of the true component values are Gaussian, then the estimates from (17) enable us to compute the tradeoff curve between type I and type II errors for each component. On the other hand, a typical board has several hundred components, and consequently it would be very time consuming to compute this curve for each individual component. A possible approach is to group components in subsets that are assumed to have very similar properties (i.e. the standard deviation of the measurement noise and the standard deviation of the component values represent nearly the same percentage of the nominal value of the component). Another important reason for grouping the observations of different components together is to increase the precision of the estimate of $\sigma_\xi$ and $\sigma_e$. The confidence intervals on the estimates for each individual component are quite large, and it is not economically feasible to run an experiment of this size on a large population of boards,.

## 4.2. Results

We first used the model presented here to find the optimal inspection allocation policy while keeping the testing policies fixed. For this purpose, three circuit boards were chosen that were representative of the variety of boards manufactured. These boards were produced in volumes that represent the average for the facility, and their yield ranged from relatively low to relatively high. One board contains mostly digital components, another has mostly analog components and the third one is mixed. . Indeed the digital or analog nature of a board is a major factor in how each test is applied. Also, these three boards did not share any test and thus can be considered independently. It turned out that when we applied the optimization of the testing policies the optimal policy was extremely sensitive to the estimates of $\sigma_\xi$ and $\sigma_e$, hence we used a different approach to derive a robust testing policy.

### Optimal allocation policy

There were important variations across boards in the number of defects present and in the effectiveness of the different stages of inspection. Figure 6 displays the frequency of the different defect types detected at each stage for the three boards under consideration. Each defect type is represented by the same pattern on all three charts. The values on each chart are arbitrary in order to disguise the data, but the relative values across the three charts are approximately correct. These graphs show that the number of defects and the predominant types of defects, vary significantly across the different boards. For example, type 3 boards have roughly three times as many defects as type 1 board types, and the medium gray defect type is predominant on type 1 boards but hardly present on type 2 boards.

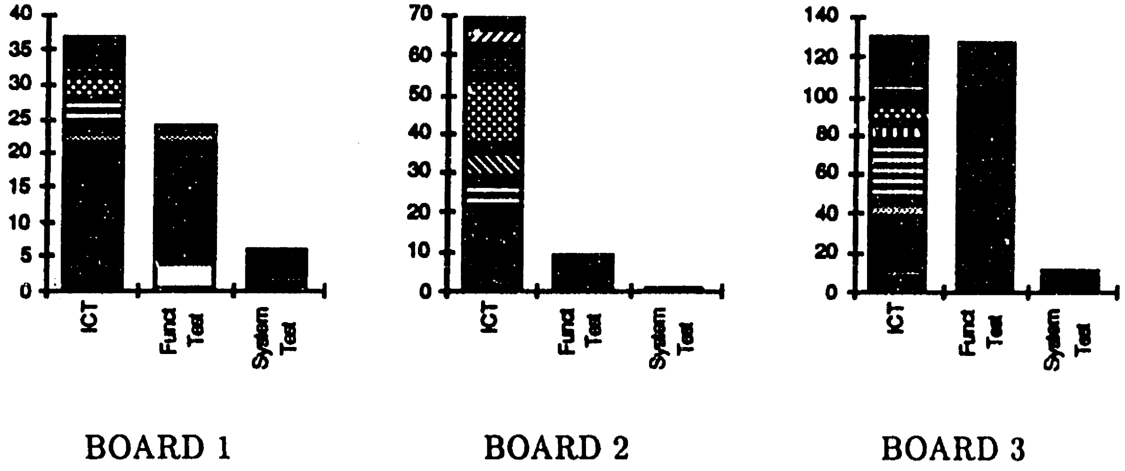


Figure 6 : Frequency of defects detected at each stage for each board.

Consequently, it is not surprising to see in Table I that the optimal policy varies for the different boards. Since the total inspection cost represents about half of the total manufacturing cost, the savings realized by the optimal policy in the different cases are significant. This study was performed with fixed testing policies because of our inability to build sufficiently accurate estimates of the measurement errors. Hence, the probability of type I and type II errors of the current testing policy were used.

TABLE I : Optimal policies and savings.

| | Current Inspection Policy | Optimal Inspection Policy | Saving |
|---|---|---|---|
| Board 1 | 1-2-3 | 1-3 | 20% |
| Board 2 | 1-2-3 | 2 | 23% |
| Board 3 | 1-2-3 | 2-3 | 6.5% |

## Application of the full model

Our intent was to apply the optimization of the testing policy at the in circuit testing stage. Based on the experiment described in the previous subsection, we obtained estimates of the standard deviation of the measurement noise and the true component values. Because individual components are very reliable (the failure rate is typically less than 1 in 10,000), the optimal testing policy is extremely sensitive to these parameters. Even estimates that could be obtained from a large experiment (recall that this plant produces boards in relatively low volumes) would not be accurate enough to ensure that a robust testing policy would result from the optimization.

Nevertheless, the insight gained from the model can be used to improve the current testing policy. Figure 7 shows the form of the function $p(x)$ in a typical case where

$$|G_u - G_l| > 6\sigma_\xi, \tag{18}$$

$$\sigma_\xi > 3\sigma_e. \tag{19}$$

The inequality (18) says that the parts are very reliable, in particular it says that the machine capability index is greater than 1. The inequality (19) says that the measurement noise is sufficiently low to have an accurate test.

In the case of normally distributed measurement noise and component values, equation (13) we see that in this case $p(x)$ is the difference of two error functions. If conditions (18)–(19) are also satisfied, then as the value of one of the error functions goes from -1 to 1, the other error function will hold constant at 1. Recall that from equation (12) we know that the optimal policy will have cutoff points such $p(L) = p(U)$ equal the marginal cost of a false accept over the marginal cost of a false defect. This ratio will be strictly between 0 and 1. This means that the optimal cutoff points will most probably be located in the steep parts of the curve in Figure 7. The middle points of the ascent and descend are the points at which the argument of

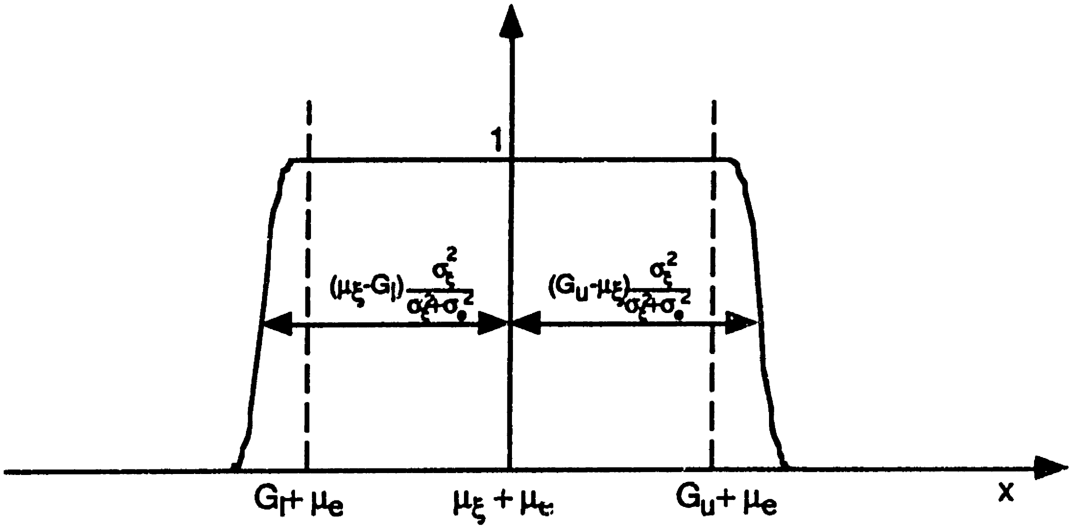

1
$(\mu_\xi - G_l)\frac{\sigma_\xi^2}{\sigma_\xi^2+\sigma_e^2}$
$(G_u - \mu_\xi)\frac{\sigma_\xi^2}{\sigma_\xi^2+\sigma_e^2}$
$G_l + \mu_e$
$\mu_\xi + \mu_e$
$G_u + \mu_e$
x

Figure 7 : The function $p(x)$.

either error function equals 0. These points can be found from equation (13) to be

$$x_l = \mu_\xi + \mu_e + (G_l - \mu_\xi)(\frac{\sigma_\xi^2 + \sigma_e^2}{\sigma_\xi^2}),$$

and

$$x_u = \mu_\xi + \mu_e + (G_u - \mu_\xi)(\frac{\sigma_\xi^2 + \sigma_e^2}{\sigma_\xi^2}).$$

This expression has an intuitive meaning : the acceptance interval should be shifted from the tolerance interval by the expected value of the measurement noise; the tolerance on both sides should be multiplied by a factor that is the ratio between the sum of variance of the true values of the component and the measurement noise over the variance of the true values of the component. This ratio shows very clearly how the acceptance interval is affected by the measurement noise.

Moreover, conditions (18)–(19) imply that one term of the right hand side of equation (13) will always be equal to 1 or -1. This in turn implies that $p(L) = p(U)$ for any $L$ and $U$ such that

$$L = \mu_\xi + \mu_e + (G_l - \mu_\xi)z \quad \text{and} \quad U = \mu_\xi + \mu_e + (G_u - \mu_\xi)z,$$

for any $z > 0$. Note that $\mu_\xi$ and $\mu_e$ are expected values that are much easier to estimate with accuracy than variances.

As a result, we can use $x_l$ and $x_u$ as target values for the lower and upper limits. Even if the estimates $\hat{\sigma}_\xi$ and $\hat{\sigma}_e$ are inaccurate, we will still have a pareto optimal policy.

It has been observed that currently about 55% of the rejected components at this test are good components. To illustrate how this would compare with the our proposed policy, Figure 8 shows the same tradeoff curve as Figure 3, but with a straight line that represents the policies such that 55% of the rejected components are good components. The point labeled a represents the point for which $L = x_l$ and $U = x_u$, and hence if the actual policy used is on the tradeoff curve in the neighborhood of a, then a significant improvement over the current policy even if the estimates $\hat{\sigma}_\xi$ and $\hat{\sigma}_e$ are inaccurate.

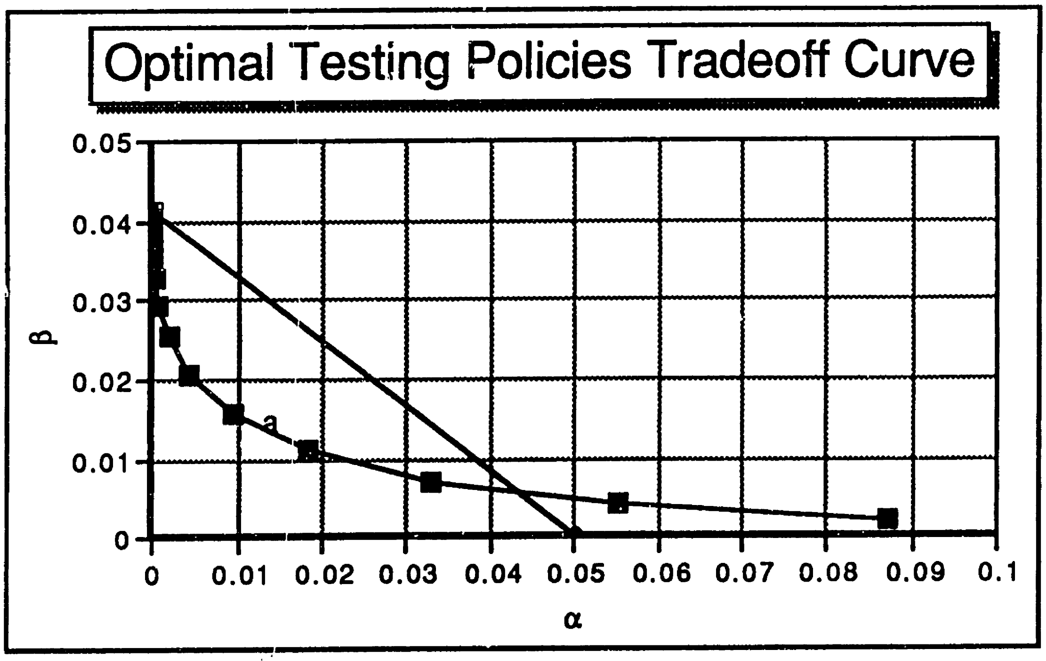


Figure 8 : The current testing policy and the optimal tradeoff curve.

Finally, it is interesting to note that for other types of tests that have lower yields (for example, tests that are more comprehensive), it is possible to estimate $\sigma_\xi$ and $\sigma_\epsilon$ with greater accuracy. In general it is possible to optimize the testing policy in high volume and/or low yield situations.

## 5. Conclusions

The model presented here is built on two key features: hierarchical test coverage and measurement errors in the testing process. Although this model was developed in the framework of circuit board assembly, these features are present in many other situations. Indeed, hierarchical test coverage is a very natural property when inspection is done at different stages in a manufacturing process; the increasing test coverage is then a direct consequence of the fact that potential defects are added between inspection stages by the manufacturing operations taking place at that time. It is also very common for test measurements to be subject to some noise, particularly in the manufacturing of high precision products.

The case study reveals that important savings (on the order of 10 to 20%) can be achieved by choosing the optimal inspection allocation policy. Since the cost of inspection represents about half of the total direct manufacturing cost, these cost savings are significant. The discussion regarding the gathering of the necessary data shows that the inspection allocation portion of the model is not very difficult to implement in an industrial setting. The optimization of the testing policy is harder to implement because of the difficulty in estimating the necessary parameters. It is nevertheless possible to use the insight gained from the model to find a good testing policy that is robust with respect to the errors in parameter estimation. In the case study, we see that this policy significantly outperforms the policy that is currently in use.

In this part, we started from a real-life problem and built a model to solve this problem. The model is still relatively crude and we were unable to find a procedure that could solve the problem efficiently. Nevertheless, it is hoped that this work will help future researchers find better models for the large number of problems of this nature that are beginning to emerge. Indeed, with the increasing performance that are sought from manufactured products it seems very likely that inspection will become a more and more important aspect of production management.

Several issues are not addressed in this thesis that would require further research. In particular the effects of the inspection policy on the learning curve of a product have not been modeled, queueing effects were not included, no precise model was build to determine the influence of the choice of a testing strategy upon the subsequent inspection stages.

## References

[1] Albin, S. L. and Friedman, D. J., "Off-Line Quality Control in Electronics Assembly: Selecting the Critical Problem," Working Paper, Rutgers University (1991).

[2] Eppen, G. D. and Hurst, E. G., "Optimal Allocation of Inspection Stations in a Multistage Production Process," *Management Science* **20** (1974), pp. 1194-1200.

[3] Garcia-Diaz, A., Foster, J. W. and Bonyuet, M., "Dynamic Programming Analysis of Special Multistage Inspection Systems," *IIE Transactions* **16** (1984), pp. 115-125.

[4] Lindsay, G. F. and Bishop, A. B., "Allocation of Screening Inspection Effort–A Dynamic-Programming Approach," *Management Science* **10** (1967), pp. 342-352.

[5] Raz, T., "A Survey of Models for Allocating Inspection Effort in Multistage Production Systems," *Journal of Quality Technology* **18** (1986), pp. 239-247.

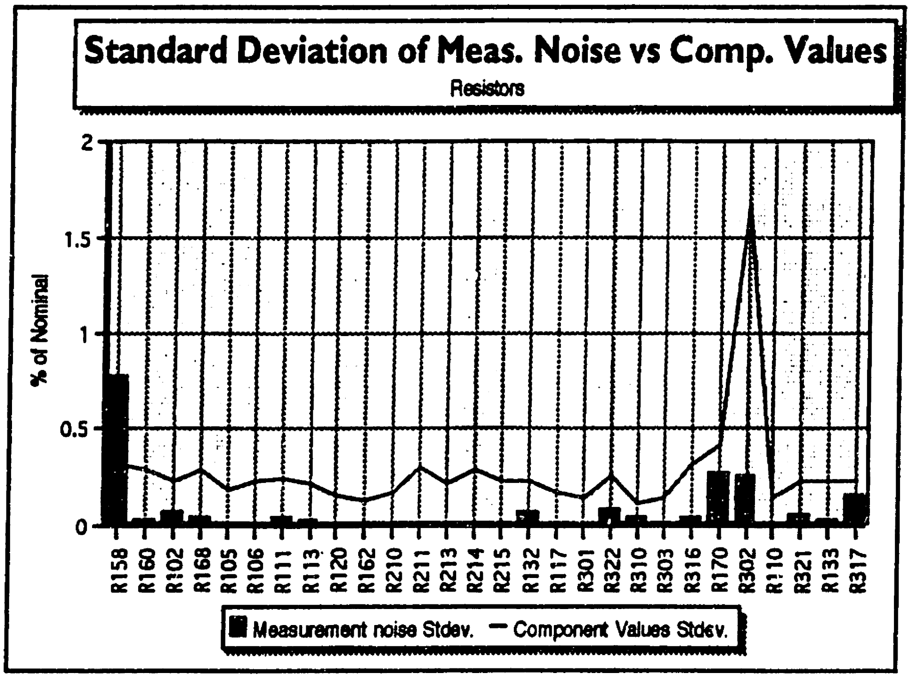
[6] Villalobos, J. R., Foster, J. W. and Disney, R. L., "Flexible Inspection Models of Partially Observable Manufacturing Processes," Working Paper, Department of Industrial Engineering, Texas A&M University (1991).

[7] White, L. S., "The Analysis of a Simple Class of Multistage Inspection Plans," *Management Science* **9** (1966), pp. 685-693.

[8] Yum, B. J. and McDowell, E. D., "The Optimal Allocation of Inspection Effort in a Class of Nonserial Production Systems," *IIE Transactions* **13** (1981), pp. 285-293.

# Resistors

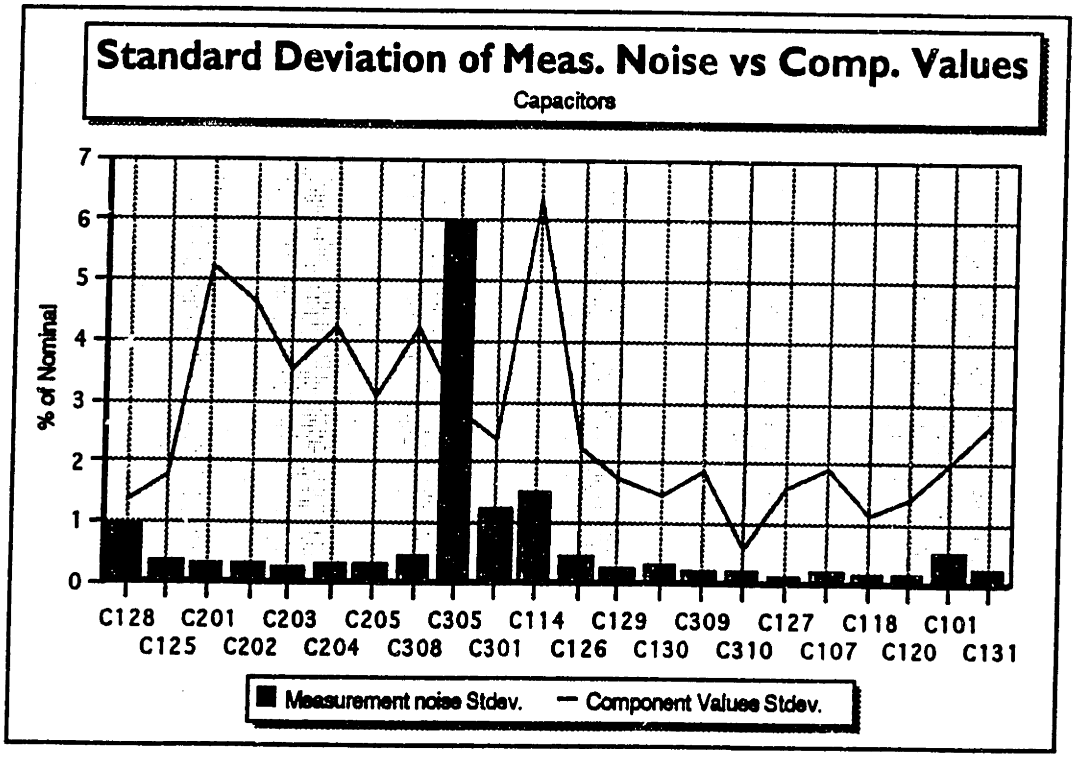
| Component | Nominal value (V) | Bias (%) | Meas. noise Stdev. (%) | Comp. value Stdev. (%) |
|---|---|---|---|---|
| R158 | 10 | 1.6580 | 0.7778 | 0.3208 |
| R160 | 21.5 | 0.7194 | 0.0224 | 0.2938 |
| R102 | 31.6 | 0.3621 | 0.0655 | 0.2384 |
| R168 | 100 | -0.0983 | 0.0370 | 0.2835 |
| R105 | 215 | -0.1170 | 0.0157 | 0.1758 |
| R106 | 1000 | 0.0142 | 0.0118 | 0.2279 |
| R111 | 1000 | 0.0530 | 0.0421 | 0.2449 |
| R113 | 1000 | 0.0965 | 0.0177 | 0.2088 |
| R120 | 1000 | 0.1492 | 0.0118 | 0.1502 |
| R162 | 1000 | 0.1059 | 0.0120 | 0.1192 |
| R210 | 1000 | 0.0582 | 0.0116 | 0.1718 |
| R211 | 1000 | 0.0280 | 0.0112 | 0.3055 |
| R213 | 1000 | 0.0896 | 0.0125 | 0.2082 |
| R214 | 1000 | 0.0408 | 0.0126 | 0.2850 |
| R215 | 1000 | -0.0084 | 0.0130 | 0.2286 |
| R132 | 1780 | 1.9721 | 0.0700 | 0.2187 |
| R117 | 2150 | 0.0126 | 0.0123 | 0.1623 |
| R301 | 4640 | -0.0987 | 0.0101 | 0.1435 |
| R322 | 5110 | -0.4706 | 0.0792 | 0.2541 |
| R310 | 8250 | -0.1981 | 0.0400 | 0.1090 |
| R303 | 10000 | -0.1581 | 0.0111 | 0.1422 |
| R316 | 14700 | -0.1642 | 0.0313 | 0.3167 |
| R170 | 17800 | -2.2224 | 0.2614 | 0.4085 |
| R302 | 21500 | -5.1729 | 0.2507 | 1.6664 |
| R110 | 31600 | 0.0407 | 0.0162 | 0.1472 |
| R321 | 68100 | -0.2850 | 0.0608 | 0.2262 |
| R133 | 121000 | -0.1053 | 0.0269 | 0.2252 |
| R317 | 464000 | 0.0122 | 0.1526 | 0.2318 |


Standard Deviation of Meas. Noise vs Comp. Values

Resistors

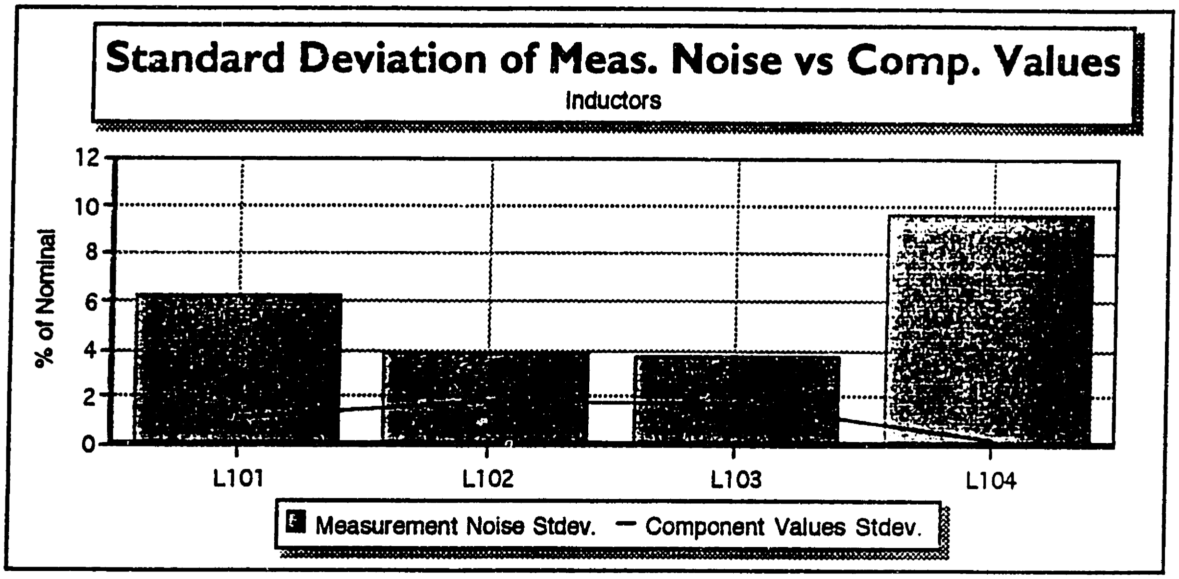
## Capacitors

| Component | Nominal value (µF) | Bias (%) | Meas. noise Stdev. (%) | Comp. value Stdev. (%) |
|---|---|---|---|---|
| C128 | 0.01 | -2.2309 | 0.9593 | 1.3864 |
| C125 | 0.1 | 2.9324 | 0.3602 | 1.8012 |
| C201 | 0.1 | 1.3018 | 0.3156 | 5.2466 |
| C202 | 0.1 | 1.0444 | 0.3189 | 4.6113 |
| C203 | 0.1 | 2.5904 | 0.2824 | 3.5374 |
| C204 | 0.1 | -0.3741 | 0.3247 | 4.2337 |
| C205 | 0.1 | 2.5666 | 0.3238 | 3.0688 |
| C308 | 0.1 | 2.8296 | 0.4630 | 4.2328 |
| C305 | 0.3 | 1.9978 | 5.9763 | 2.8549 |
| C301 | 0.4 | -2.8866 | 1.2112 | 2.3799 |
| C114 | 0.47 | -12.5308 | 1.4739 | 6.3439 |
| C126 | 0.47 | 0.8126 | 0.4203 | 2.2399 |
| C129 | 0.47 | -0.2691 | 0.2845 | 1.7317 |
| C130 | 0.47 | 0.7670 | 0.3225 | 1.4685 |
| C309 | 0.47 | -0.3155 | 0.1996 | 1.8899 |
| C310 | 1 | -7.6399 | 0.1894 | 0.5920 |
| C127 | 6.8 | -1.5503 | 0.1315 | 1.6210 |
| C107 | 6.9 | 2.3444 | 0.2334 | 1.9289 |
| C118 | 6.9 | -2.1074 | 0.1436 | 1.1643 |
| C120 | 7.2 | -0.1760 | 0.1595 | 1.4322 |
| C101 | 33 | 0.1986 | 0.5414 | 1.9831 |
| C131 | 33 | -0.0072 | 0.2814 | 2.6723 |



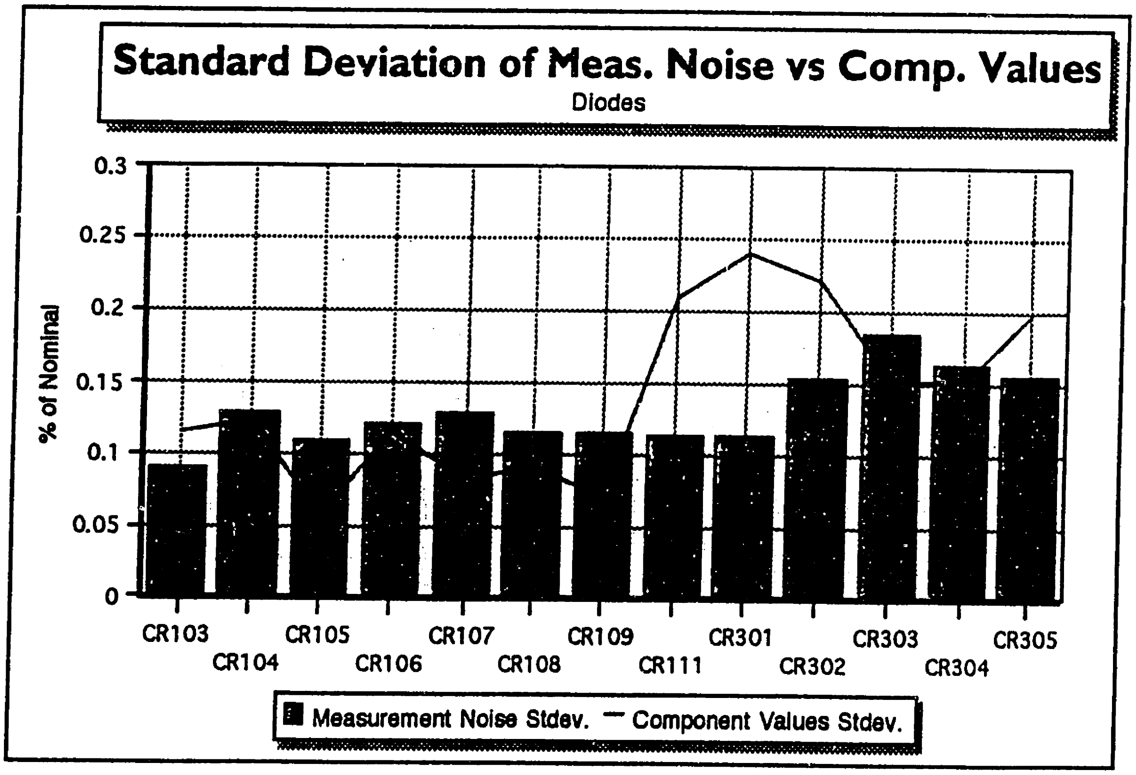
# Inductors

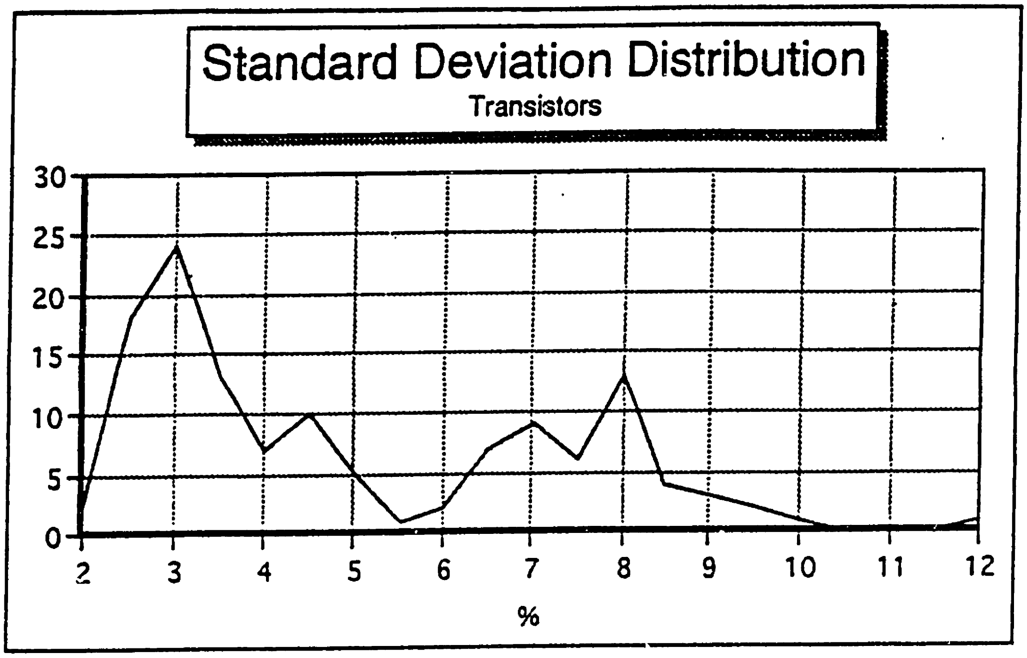
| Component | Nominal value (µF) | Bias (%) | Meas. noise Stdev. (%) | Comp. value Stdev. (%) |
|---|---|---|---|---|
| L101 | 1 | 37.5506 | 6.3128 | 1.1392 |
| L102 | 1 | 46.9465 | 3.7830 | 1.8730 |
| L103 | 1 | 45.8132 | 3.6897 | 1.7435 |
| L104 | 1 | 37.0869 | 9.6679 | 0.2264 |

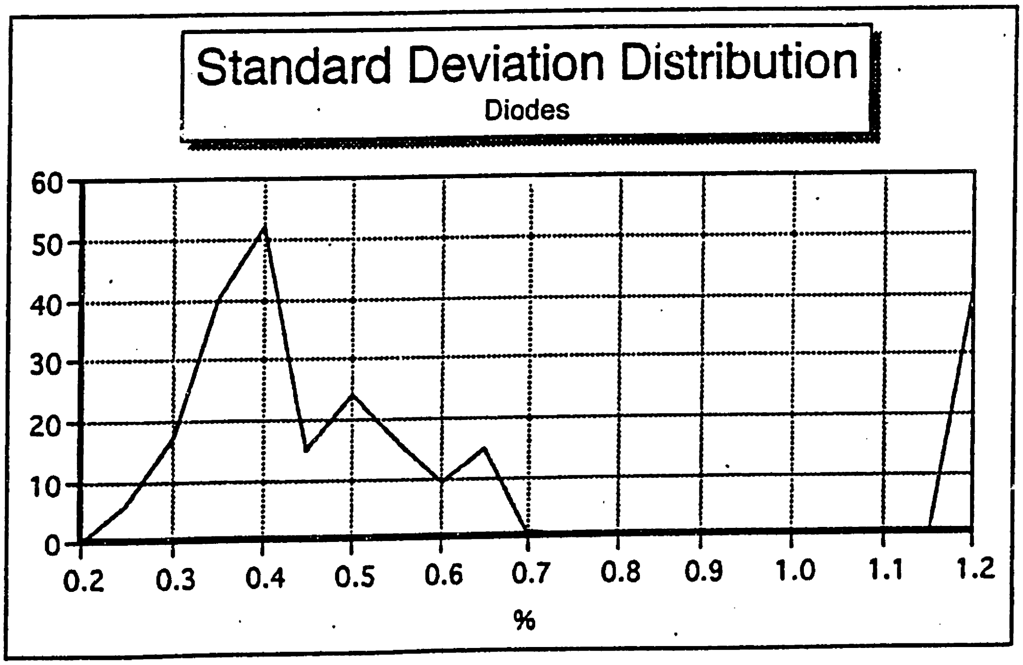


# Diodes

| Component | Nominal value (V) | Bias (%) | Meas. noise Stdev. (%) | Comp. value Stdev. (%) |
|---|---|---|---|---|
| CR103 | 1.622672 | N/A | 0.0890 | 0.1148 |
| CR104 | 0.70446 | N/A | 0.1272 | 0.1220 |
| CR105 | 0.724097 | N/A | 0.1102 | 0.0556 |
| CR106 | 0.726004 | N/A | 0.1210 | 0.1145 |
| CR107 | 0.72484 | N/A | 0.1269 | 0.0821 |
| CR108 | 0.723124 | N/A | 0.1144 | 0.0904 |
| CR109 | 0.723348 | N/A | 0.1136 | 0.0695 |
| CR111 | 2.168599 | N/A | 0.1133 | 0.2104 |
| CR301 | 0.717516 | N/A | 0.1131 | 0.2404 |
| CR302 | 0.590526 | N/A | 0.1536 | 0.2206 |
| CR303 | 0.59102 | N/A | 0.1833 | 0.1520 |
| CR304 | 0.595119 | N/A | 0.1643 | 0.1513 |
| CR305 | 0.597192 | N/A | 0.1566 | 0.1970 |

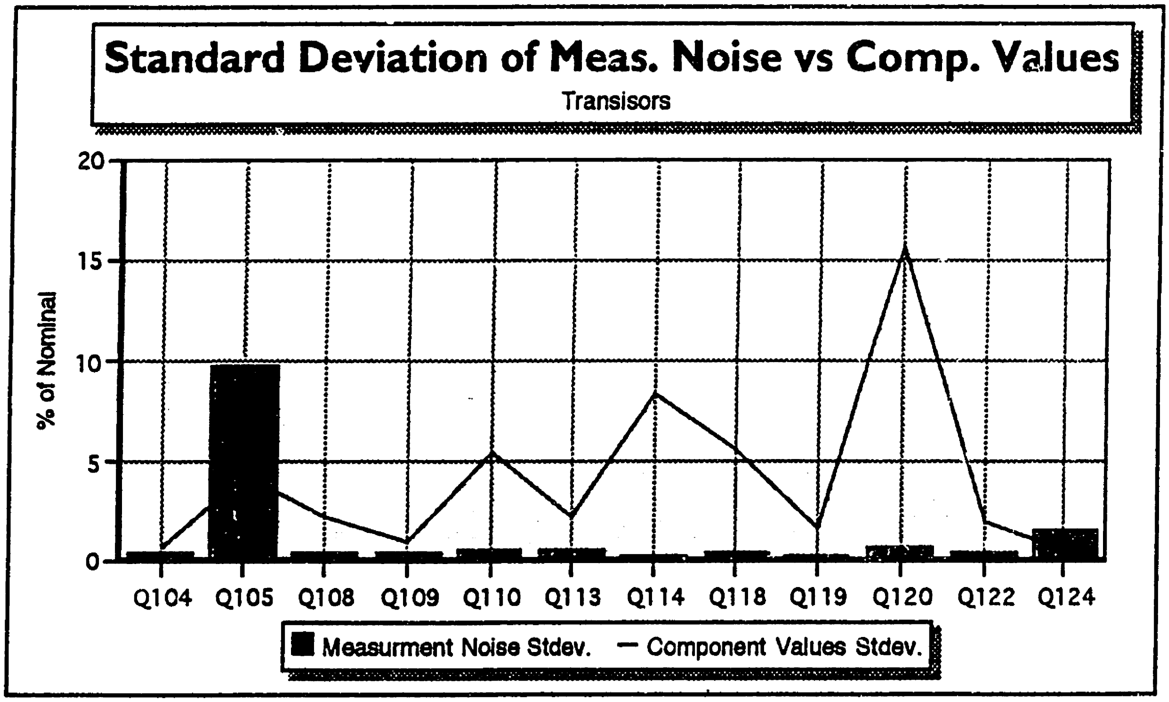



Standard Deviation Distribution
Transistors
30
25
20
15
10
5
0
2
3
4
5
6
7
8
9
10
11
12
%


Standard Deviation Distribution
Diodes
60
50
40
30
20
10
0
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
%

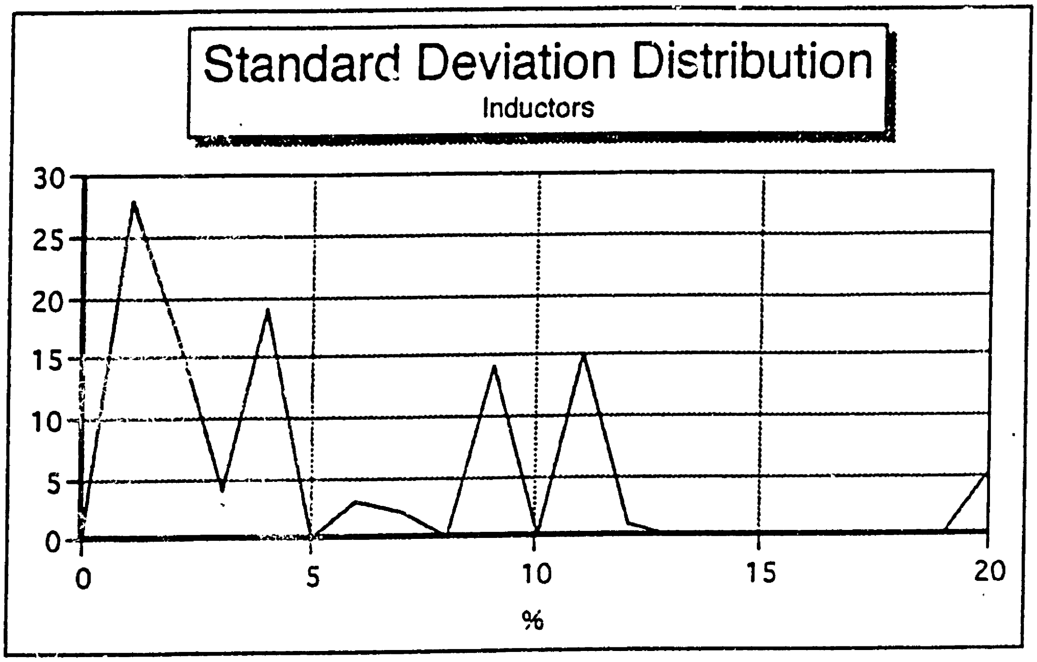
# Transistors

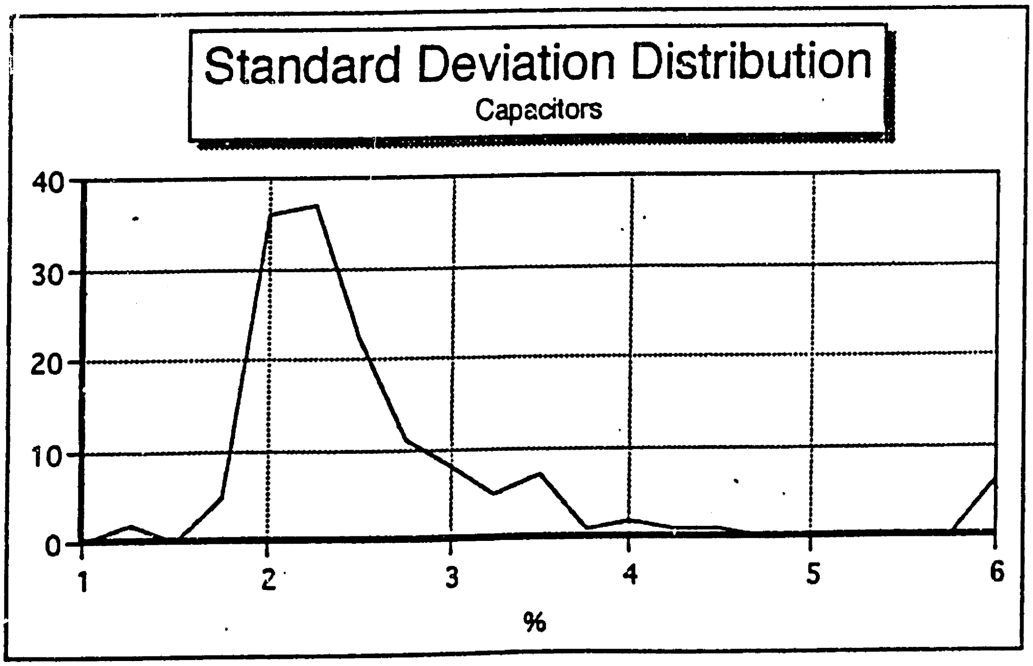
| Component | Nominal value (V) | Bias (%) | Meas. noise Stdev. (%) | Comp. value Stdev. (%) |
|---|---|---|---|---|
| Q104 | 50.1388 | N/A | 0.3870 | 0.7738 |
| Q105 | 59.753133 | N/A | 9.6449 | 4.3180 |
| Q108 | 92.116 | N/A | 0.4122 | 2.2315 |
| Q109 | 37.7534 | N/A | 0.3575 | 0.9633 |
| Q110 | 28.299333 | N/A | 0.5867 | 5.4653 |
| Q113 | 28.703933 | N/A | 0.5577 | 2.2126 |
| Q114 | 36.0998 | N/A | 0.2703 | 8.3509 |
| Q118 | 36.404233 | N/A | 0.3368 | 5.5723 |
| Q119 | 37.419267 | N/A | 0.2479 | 1.6809 |
| Q120 | 34.036667 | N/A | 0.6842 | 15.5969 |
| Q122 | 48.807333 | N/A | 0.3554 | 1.9639 |
| Q124 | 75.055467 | N/A | 1.5632 | 0.4967 |

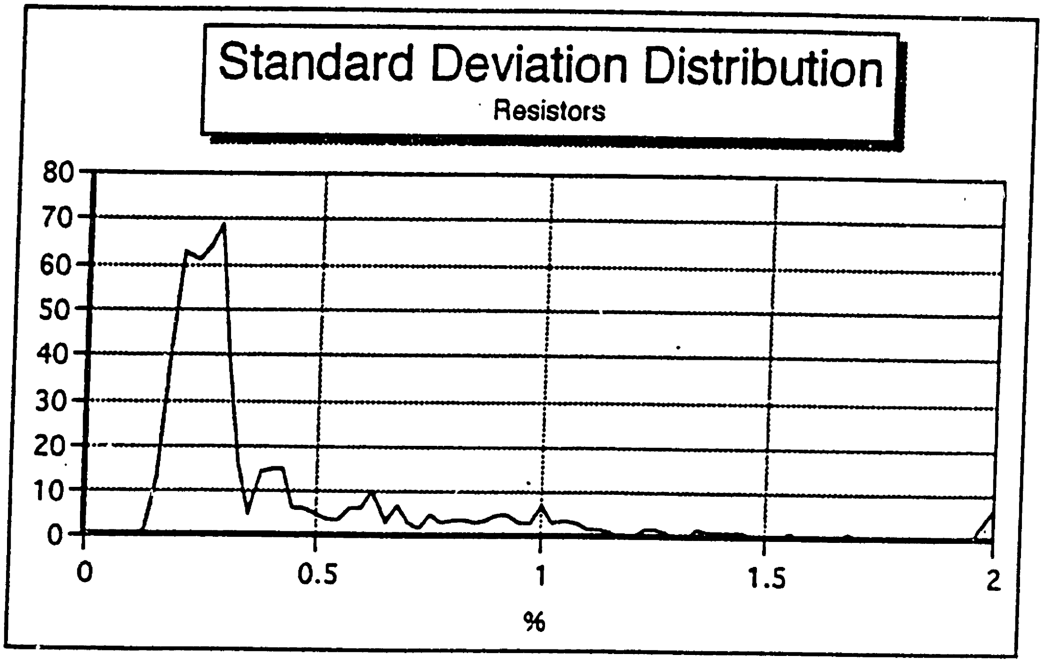


Standard Deviation of Meas. Noise vs Comp. Values

Transisors


Standard Deviation Distribution
Inductors
30
25
20
15
10
5
0
0
5
10
15
20
%


Standard Deviation Distribution
Capacitors
40
30
20
10
0
1
2
3
4
5
6
%


Standard Deviation Distribution
Resistors
80
70
60
50
40
30
20
10
0
0
0.5
1
1.5
2
%