UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

THE EFFECT OF INTERMITTENT FORGETTING UPON LEARNING AND PRODUCTIVITY WITHIN PRODUCTION SYSTEMS

A Dissertation

SUBMITTED TO THE FACULTY

in partial fulfillment of the requirements for the

degree of

DOCTOR OF PHILOSOPHY

By

TIMOTHY R. BRIDGES

Norman, Oklahoma

2000

UMI Number: 9975792

UMI®

UMI Microform 9975792

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

THE EFFECT OF INTERMITTENT FORGETTING
UPON LEARNING AND PRODUCTIVITY
WITHIN PRODUCTION SYSTEMS

A Dissertation APPROVED FOR THE
SCHOOL OF INDUSTRIAL ENGINEERING

BY

Dr. Adedeji Badiru

Dr. Robert Schlegel

Dr. Mary Court

Dr. Hank Grant

Dr. Richard Tersine

ACKNOWLEDGEMENTS

I am indebted primarily to Dr. Adedeji Badiru for his patience and support during this long endeavor. His advice, encouragement and steering have kept me focused on the material and the potential uses of work performed in this much understudied area. I am also indebted to the other members of the committee for their valuable time and inputs toward reaching this goal.

I am also indebted to my wife Nancy and my entire family who have suffered through the loss of my attention and companionship while I pursued this effort and continued to maintain an active business life. Without their understanding and patience I could not have concluded this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

ABSTRACT

THE EFFECT OF INTERMITTENT FORGETTING UPON LEARNING AND PRODUCTIVITY WITHIN PRODUCTION SYSTEMS

BY: TIMOTHY R. BRIDGES

MAJOR ADVISOR: ADEDEJI BADIRU

This dissertation addresses the effects of intermittent forgetting upon the learning and industrial production process. This work can apply to any process in which learning must occur and may re-occur at a later point in time following a break. Learning occurs whenever a new endeavor is accepted and continues through the life cycle of that endeavor. During the learning process and during breaks away from the learned information, forgetting occurs unless action is taken to preclude it. Through the use of simulation, this document demonstrates the effects of intermittent forgetting upon learning, productivity and overall performance using a resolution function when the time of occurrence and the duration of forgetting are predicted using a simulated queue. The resulting, simulated queues provide the times at which intermittent forgetting is initiated and the duration of forgetting. Accounting for the reduced overall rate of learning, forgetting is understood even less than learning, yet it

cannot be ignored when planning the production schedule or constructing bids within a manufacturing environment. This work begins with a review of the latest findings affecting learning and forgetting, and includes the models to describe these processes. A C++ program was then run to generate the queues and compute the result upon overall learning and relearning following production breaks. The model used was multivariate using units produced and times as means to measure productivity. The model reviewed four situations:

1. No Forgetting
2. Minimal Intermittent Forgetting
3. High Intermittent Forgetting
4. Continuous Forgetting.

It was demonstrated that with a reasonable model to describe the learning and forgetting processes, an impact due to intermittent forgetting can be predicted and subsequently minimized. People cannot operate at peak levels continuously and it is at those times when humans are vulnerable to forgetting. Also, when not engaged within the learned activity for a period of time, the more complex aspects of the activity will be less likely to be recalled without help. This occurs in industry when production breaks occur and in all activities in which learning new information is required but will not be utilized continuously. Generally, the longer the break between the use of information, the more difficult it is to recall. This excludes those instances when during brief breaks a portion of the cognitive process may continue and actual learning, not forgetting, continues.

The results of the study showed that a significant portion of resultant learning (15-20 percent) can be affected by intermittent forgetting during a cyclic production process. It is clear that the various types of learning that occur within a production environment must be analyzed independently in order to determine a means to minimize forgetting impacts. This is due to the occurrence of three types of activities occurring simultaneously: worker learning, organizational learning, and process improvement. Forgetting applies primarily to the human element, assuming that organizational learning and process improvements are documented and not lost nor difficult to retrieve. The results of this study apply in the academic world as well as the industrial world wherever forgetting is taking place which can hamper the overall learning process.

1.0 Introduction

This research concentrated in the area of learning and forgetting where learning is assumed to be continuous with forgetting occurring intermittently. To investigate this scenario, a single server queue is used in a simulation to generate arrivals, service completions, and the existence of queues when arrivals occur faster than servicing. The existence of a queue denotes a forgetting period. Thus, forgetting begins when a queue is established and forgetting ends when there no longer is a queue. Arrivals occur at a variable rate thus increasing the variability in the time between arrivals, which is described by an exponential process. The service rate remains constant with service times also described by an exponential process. A number of replications of a next event Monte Carlo simulation is run with the queue beginning and ending times recorded and averaged over the selected number of replications.

These times are then applied using varying amounts of forgetting during the learning process in a production scenario. It is expected that the importance of forgetting upon the learning process will be highlighted when considering employee/system performance. Applications will be sought in the training arena, manufacturing economic analysis, manpower requirements and scheduling, production planning, bid preparation, budgeting, and overall resource allocation.

Production progress functions are a method of measuring and estimating the rate at which an active organization learns to produce a product. This type of learning has been found to follow the same negative power functions that are used for human learning. However, the learning rate is considerably faster, the most common rate being the 80% curve with the dependent variable expressed as cost per unit of time per unit produced.

Production progress thus occurs in many industries at a rapid rate. It continues for large numbers of units and in a real sense, production progress is the antithesis of standard costs because production progress causes costs to drop 20% or more every time the units produced doubles. Thus, costs are constantly changing in the startup phases of all production efforts, whether producing an item or performing a service. Knowledge of production progress functions is very important to those who are involved in bidding, cost analysis, and the pricing of products and services.

2.0 Background

In order to study forgetting effects upon the learning process, an understanding of how humans learn and how they apply that learning to their behavior is essential. This is the motivation for studying learning because it can be employed in so many areas of the manufacturing and business planning process. Possible uses include:

a. The management control system that is the primary mechanism used to assess managerial effectiveness.

b. The learning curve in budgeting to improve managerial planning and control of operating costs.

c. Enhancing performance measurement by inclusion of a "learning effect variance".

d. Employing learning in Break Even Analysis to better define the variable cost per unit.

e. Evaluating the effects of pre-production planning in flattening the learning curve.

f. The potential contribution of the learning curve in the evaluation of learning on its own products (bidding) as well as on products of its suppliers (purchasing).

g. The evaluation of a potential supplier to ensure the supplier is incorporating learning into the cost and pricing of their product.

h. Using the learning curve in computing the economic order quantity or the economic production quantity.

i. The application of the learning model to accident experience to validate the point of learning required to affect accident statistics.

j. The learning curve to predict the level of warranty service required on newly manufactured items produced early in the production phase of applied learning.

k. Utilizing the learning or experience curve to make major decisions in the marketing area which ultimately affects manufacturing operations. The major assumption made is that the manufacturer who has produced the most units probably has the lowest unit cost due to learning. As manufacturing experience is gained, the lowest unit cost is achieved via the learning curve phenomenon. Then, market dominance becomes a reality by using price as a competitive strategy.

l. Incorporating the learning phenomenon in the aggregate planning model for more realistic information for managerial planning purposes.

2.1 Definitions

In order to proceed, a few terms must be defined; namely, learning, memory, and forgetting. Webster defines learning as gaining knowledge or understanding of or a skill in something by study, instruction, or expression. Another definition further attempts to describe learning as that which assures cognition in the sense of mechanisms for the processing of information during its initial acquisition and storage. Memory is defined as an individual's internal representation of all that has been acquired and stored. Many use learning for the storing of new information and memory for maintaining that storage for potential retrieval resulting in behavioral expression.

Forgetting is associated with the inability to recall previously understood information stored in memory. For the purposes of this study, learning will consist of the comprehension and storage of information and retention will be referred to as the retention in storage and recall of information.

2.2 Knowledge Acquisition

The brain does not store a picture of an event. It does not directly record anything that it is shown. What the brain does is store a record of neural activity that takes place in the learner's sensory and motor systems as they interact with the environment. Thus, when learners place an image in their mind, they store components in many different places and construct pathways among the places so that the entire system storage and pathways can fire up as an image when the learners recall the experience.

The only way the brain absorbs data is through the sensory perceptions that enter through the body's five senses. Anything that a person does, perceives, thinks, or feels while acting in the world gets processed through complex systems of pathways and storage. The brain categorizes non-language sensory perceptions of the world in different places. Shapes and colors are stored in different places. Movement, sequence, and emotional states are also stored separately. Textures and aromas are stored elsewhere. Aspects of language are also stored in various parts of the brain. Nouns are separated from verbs, and phonemes are separated from words. As the brain constructs connections among the brain cells, it connects the organizations of words, objects, events, and relationships in successively interwoven layers of categories. The result is that human knowledge is stored in clusters and organized within the brain into systems that people use to interpret familiar situations and to reason about new ones (Lowery, 1998).

Information is simply stored in different areas of our brain. Sprenger calls the separate pathways to the information stored in various areas of the brain as memory 'lanes' and she has identified five separate lanes; semantic, episodic, procedural, automatic, and emotional. The structure in the brain called the hippocampus controls the declarative memory. The hippocampus does not store the memories themselves, but it catalogs them (Sylvester, 1995). Declarative memory consists of semantic information (facts, places, names) and episodic information (episodes of one's life). Both types of declarative memory can be "declared", or stated, and are believed to be stored in the outer layer of the brain, the cortex (Wolfe, 1998). Semantic memory, because it deals with words, is the lane most relevant to education and initial learning. It is also the most difficult memory lane to use. The hippocampus is like a filing cabinet with two drawers, one semantic and the other episodic. Because it identifies sensory memories worth saving, it controls access to the episodic memory lane (Sylvester, 1995). Episodic memory is location-driven; remembering where you were when you learned something can help trigger the memory.

The procedural memory lane is found in a brain structure called the cerebellum. This formation deals with posture, balance, and some event memory. Procedural memories are sometimes called "muscle memories" or "how-to" memories because they refer to physical movements such as riding a bicycle, driving a car, and tying a shoelace. Procedural memory consists of information or procedures that have been learned at the automatic level that often are accessed without conscious attention. Neuroscientists believe that the physiological process underlying procedural memory is one in which brain cells (neurons) that "fire together, wire together." In other words, circuits or networks of neurons that are used over and over get accustomed to firing together and eventually become hard wired and fire automatically. It is interesting to note that Madeline Hunter, a noted educational author, used the phrase "Practice doesn't make perfect, it makes permanent" (Wolfe, 1998).

The automatic memory lane is also located in the cerebellum where it is sometimes called reflexive or stimulus-response memory. Automatic memory is triggered by such things as flash cards, music, and other repetitive devices that are not necessarily physical. Some examples of information stored in automatic memory are the alphabet, the multiplication tables, and song lyrics.

The final lane is the emotional memory. The amygdala, a structure in the forebrain, is in charge of all emotional memories. If the perception has some emotional content, then the amygdala processes that memory. Like the hippocampus, the amygdala does not store all memories; it simply catalogs them (Sprenger, 1998). When information is first received through the senses it is sent to the small, almond-shaped amygdala which checks the information for its emotional content. If the brain determines that the information is threatening or evokes deep emotion, it immediately sends chemical messages throughout the body to prepare the organs to adjust their activity level to match the demands of the situation. When this occurs the cortex becomes much less efficient, in a sense it downshifts when the brain interprets a threatening situation. This can slow the creative, rational processing of information. Thus, emotion becomes a double-edged sword. If the event or information being evaluated by the amygdala has little or no value, the brain has a tendency to drop it. If the emotional content is too high, downshifting in the cortex can occur, and conscious, rational processing becomes less efficient (Wolfe, 1998).

Many other researchers prefer to call our memory as sensory storage, short-term or working memory, and long-term memory. Sensory storage involves a temporary storage mechanism that can extend sensory inputs for a very short period after the cause has been eliminated. The most prominent of these is iconic storage for visual inputs and echoic storage for auditory inputs. Working or short-term memory encodes information received from the senses and holds it in working memory where it is coded with three different codes: visual, phonetic, and semantic. The first two are visual and phonetic or auditory codes that represent stimuli. Semantic codes are abstracts of the meaning of a stimulus. Any single stimulus could evoke all three codes (Sanders and McCormick, 1993). Conscious thought occurs in working memory where calculations occur, information is interpreted, and information is applied for problem resolution. Mayhew (1992) adds another form called intermediate-term memory where intermediate solution results are stored. Information in working memory is transferred to long-term memory by semantic coding, in other words by providing meaning to the information and relating it to information previously stored in long-term memory (Sanders & McCormick, 1993).

The process of learning is complex. Humans use prior knowledge to interpret new material in terms of established knowledge. This initial processing step is unconscious and appears to be accomplished by the brain as it searches through previously stored information and looks for relevant hooks for the new information. The brain constantly searches through existing neural networks to find a way to make sense of incoming data. An anticipatory set increases the possibility that the brain will search through the right networks and attend to the information that is relevant for a particular topic or issue (Wolfe, 1998). New knowledge (learning) is actually gained from this rearrangement of prior knowledge into new connections (Lowery, 1998). Synaptic connections between the neurons in the brain are frequently only temporary. What is desired to show acquisition of information is the growth of an axon of one nerve cell with the dendrites of another. These occur when the experiences understood are both novel and coherent (Jensen, 1998). If the experiences were familiar, then the connections are simply strengthened. Thus, each new challenge does two things: it provides a rehearsal of prior knowledge constructions, thus making them more permanent, and provides something new that the brain can assimilate into its prior constructions, thus enriching and extending those constructions. When prior information is strengthened leading to improved performance it is called practice. Practice is useful in a limited context, but it has little transferability. Rehearsal, however, takes place when people do something again in a similar but not identical way to reinforce what they have learned while adding something new. New additions increase the likelihood that the knowledge learned is not task-specific. Non-task-specific experiences increase the likelihood that the knowledge will be transferable and useful in a variety of ways. Rehearsals strengthen the connections among the storage areas within the brain systems (Lowery, 1998). If connections are not strengthened, they will disengage and fade away - Use it or lose it (Diamond & Hopson, 1998).

It has been shown that the brain exhibits growth cycles that are known to repeat themselves several times, most notably between birth and 30 years of age. Humans have an opportunity for relearning skills and reshaping networks that they missed learning in earlier cycles. Wolfe and Brandt (1998) refer to times of increased learning as "windows of opportunity" or "critical periods" when learning should be maximized. This cyclical property seems to explain the remarkable human capacity for plasticity, including recovery from damaging environments and neural injuries, especially when later development occurs in a benevolent, nurturing environment (Diamond and Hopson, 1998). This rule includes restructuring neural networks throughout a person's life since documented evidence shows how stroke victims suffering damage in one part of the brain can reconstruct networks to access information. People do not lose the knowledge that allows them to function, they merely lose the network to that information and another path must be established.

2.3 Learning, Retention and Rates

Learning and retention are independent concepts. The degree of retention can, however, be determined by the degree of learning. Some distinctions between learning and retention are:

1. Retention is measured when the interval since the last training trial exceeds the training intertrial interval.

2. Retention refers to the expression of previously acquired information at some point after one is removed from the physical presence of the information.

3. Functional may be the strongest distinction. Existing learning literature provides that:

a) Learning is faster when a target list of verbal items has a higher degree of meaningfulness. But, once learned, meaningfulness has no influence on retention.

b) The rate of learning between individuals can be significant. However, the rate of forgetting may be very much the same.

c) In paired association tasks, the spacing of trials has little effect on learning, but the greater the recall practice, the more markedly enhanced is retention. Thus, if one wishes to assess differences in forgetting, the degree of original learning must be held constant.

In most studies, researchers study the variables that make a difference in rates of learning and the immediate performance of that learning with little concern for processes responsible for retention. William James (1890) emphasized that people had a basic trait of retentiveness. He agreed that retention of required knowledge might be made more effective by certain mnemonic techniques; but merely as a way to begin study, not an advantage for remembering. Because in early Roman times information was presented verbally, the discipline of mnemonics techniques for accurate recall were taught. Recent research confirms that the mnemonics technique affects learning rather than retention (Spears and Riccoli, 1985).

The technique of pairing names with discrete objects or spatial locations is known as the "method of loci" and has been applied to pairing basic ideas with distinctive spatial locations. Through the ages down to Bacon and Descartes, memory was maintained as an art to be practiced rather than an object for scientific study.

It was not until the late 19th century that information about the nature of memory processing became sufficiently systematic to qualify as a science. Sechenov and Pavlov were concerned with processes that affect retention and forgetting, and they suggested - memory mechanisms - by which acquired information might be maintained following learning and how to modify subsequent behavior. Also during the 1800's, the study of memory and retention focused on the parts of the brain that hosted information acquired, some even feeling that head bumps reflected behavior because of the area of the brain in which they were located.

2.4 Learning Applied to Tasks

Dar-El, Ayas, and Gilad (1995) believe that information is symbolically encoded and held within the memories of the cognitive system. The long-term memory holds the user's mass of available knowledge. The short-term memory allows the user to use this information. When a simple manual task is performed, use of the cognitive system is minimal because there is no need to retain or withdraw any information from cognitive memories. Complex tasks, on the other hand, require retrieval of information from these cognitive memories. The more complex a task, the greater is the need to access long-term memory and the longer the time for motor response. In complex tasks, both cognitive and motor learning occurs.

Hancock and Bayha (1977) believed similarly that learning can be separated into two major areas: that which occurs when a person is performing a task repetitively and that which occurs while an organization produces many units of a particular product. The former area is called "human learning", and the latter the "production progress function", which is the term generally used in the literature.

According to published research, the main factors affecting the rate of learning can be classified into two categories. The first category includes factors related to the complexity of the task such as cycle length, amount of uncertainty in the motions involved, amount of prior training, and the extent of thinking and decision making required to execute the task. The second category includes factors related to the capabilities of individuals such as age, mental and physical capabilities, the existence of an incentive to improve, and the level of past performance.

"Threshold Learning" is defined as that learning which takes place prior to workers barely knowing how to do a job without external assistance. Carlson and Rowe (1976) refer to this period as the Incipient Phase because very little productive learning occurs during this period. Threshold learning is not included in the learning curve equations because the methods used are so variable that they cannot be predicted with any accuracy. Thus, the threshold learning time has to be added to the time predicted by any equations. The learning curve equations contain the "conditional learning time", which is the time that it takes the worker to learn after he or she barely knows how to do the work. Observation, not actual performance of a task, results in the longest threshold learning times. Unfortunately, this is the method that is most commonly used in industry. Considerable cost savings are possible by using formal instruction by people who are thoroughly familiar with the operations to be performed and who have the ability to teach the method sequences to the worker properly. Although the learning curve equations suggest that people learn at a smooth rate, this is probably not the usual case.

At the early stages of learning, the operator uses the cognitive system to perform the task correctly, i.e., follow instructions, remember the sequence of operations, develop the correct method, etc. As experience is gained in task execution, the time spent on cognitive learning is sharply reduced and the performance time is dominated by motor learning. Thus, when simple tasks are initiated, only motor responses occur. A hypothetical task requiring only cognitive skills will evolve only the cognitive phase of learning. However, most industrial tasks require the simultaneous use of both motor and cognitive skills and consequently their learning curves are assumed to be composed of the two hypothetical parallel phases.

According to Fleischman and Rich (1963), different characteristics appear to limit the performance stages of practice. At the early stages, cognitive abilities are dominant, whereas later, the motor activities become the limiting factor. Hancock and Foulke (1963), define the two phases of learning as: threshold learning and conditioned learning, which, by their definitions, are respectively equivalent to cognitive and motor learning. Hancock and Foulke investigated only motor learning, i.e., the conditioned part of the learning process. Except for the work of Hancock and Foulke, no other studies appear in the literature that distinguishes between cognitive and motor learning.

When humans begin to perform a motor task, they need to use their eyes very frequently to get information. As they continue to repeat the task, they "chunk" information where chunking in this application refers to a process wherein people get more information per eye fixation and thus, with increasing numbers of cycles, the need to use their eyes to obtain the necessary information to complete a task becomes less and less.

People not only attempt to reduce cycle times by chunking information, but they also attempt to get the information they need by using their lower-order senses, especially their kinesthetic (sense of position) and tactile sense (sense of touch). The motivation of this effort is that the use of the eyes consumes a large amount of time as compared to the use of the lower order senses. They use their eyes a minimum amount. Therefore, the industrial engineer can tell the experience level of a worker by watching his or her eyes. The worker who can look all around and still work without interruption is a skilled worker who does not have to rely heavily on visual information to complete a task. This situation bothers many managers because they feel that the workers are not paying attention to what they are doing and sometimes discipline them erroneously for it.

All of this may explain why Glover found it necessary to add a work commencement factor to obtain a good fit to his data, and it may partly explain the post-learning drift observed by Towill and Kaloo (1978). Learning slope values published in the literature consequently apply to both cognitive and motor components of learning. What we observe in reality is the combined effect of these two processes. Thus, during the early repetitions, when cognitive processes dominate the learning curve, the learning constant b can be expected to have a high value, whereas as experience is increased, the magnitude of b begins to drop towards a value nearer that associated with the motor process.

Now, consider learning within a manufacturing environment where man and machine operate together. Hirsch studied the relationship of learning when in a machining environment and he noted that the progress ratio decreases as the proportion of machine paced labor to total labor increases. Hirsch also found in his study that approximately 87% of the changes in direct labor requirements were associated with changes in technical knowledge that he interpreted as organizational learning.

Yelle elaborated upon the machine extensive manufacturing area noting Baloff's conclusion that the possibility of plateauing is much higher in this environment because the opportunities to learn are less, even when employing machines with some learning capability. Conway and Shultz (1959) highlighted this when they noticed a further progression down the learning curve when a process was moved to another firm employing a different mix of machine-human labor.

2.5 Expressing Learning Mathematically

Another early mathematical presentation of learning was that of Snobby and Wright in 1936. They developed an analytical model in which task performance time is represented as a *power function* of the cumulative number of repetitions, with the following form:

$$T(s) = as^{-m}$$

where:

s - the cumulative number of repetitions

$T(s)$ - the performance time of the sth repetition

a - a starting point parameter ($a = T(1)$)

m - a curvature parameter

and $m = \log \Phi / \log 2$

with Φ = the learning rate

$1 - \Phi$ = the progress ratio

Exponential equations applied to learning curves are of the following general form:

$$y = k\left(1 - e^{-t/R}\right)$$

where e is the base of the natural logarithm, y is some measure of learning, and t is the amount of time or the number of trials of training, study, or practice. Parameter k defines the predicted asymptote for performance. Parameter R is the learning rate parameter, which specifies how quickly the asymptote k is approached. The larger the value of R, the slower the rate of improvement in performance.

2.6 Alternate Learning Models

In their textbook of mathematical psychology, Restle and Greeno (1970) called their model of learning a "replacement model" because it suggests that learning is a process through which incorrect response tendencies are replaced by correct response tendencies. Presently, psychologists have little interest in the shape of the learning curve. Nevertheless, reliance on the exponential equation has not declined, and it appears in the foundations of many recent theoretical treatments of learning in both humans and non-humans.

Restle and Greeno also contrasted the replacement model with what they called the "accumulation model" of learning. L.L. Thurstone (1919) first introduced the accumulation model in a monograph entitled "The Learning Curve Equation". According to the accumulation model, learning is a process by which correct response tendencies increase steadily with practice and compete with incorrect response tendencies, which are constant. This model predicts that the learning curve can be expressed by a hyperbolic equation. Restle and Greeno noted that counterexamples to the smooth, monotonic, concave-downward functions predicted by both models are abundant. S-shaped curves are often observed that have an initial positively accelerating segment followed by a negatively accelerating one. Simple, positively accelerating curves are rather less common but can also be obtained, for instance, when subjects are required to learn difficult, pair-associated items. In other experiments learning seems to occur in bursts of insight. Subjects make no visible progress until the solution to a problem is suddenly achieved. Learning curves for Morse code sometimes contain plateaus where little progress is made in the middle of training, only to be followed by new increases in the learning rate with further experience.

The constant rate condition seems to rule out application of the models in the cases where the subject's motivational levels can be expected to fluctuate significantly. Ideally, the models would be compared in learning situations where total performance is a function of responding on a large number of distinct, equally difficult subtasks where the learner's motivational state is constant across training and test levels.

In all of the cases studied by Restle and Greeno, the accumulation model accounts for more of the variance than the replacement model. It was also determined that as the length of presentation is increased, the replacement model predicts no increased recall.

The major empirical finding of the review by Restle and Greeno was that the accumulation model describes the shape of most group learning curves better than the replacement model. No systematic deviations from the predictions of the accumulation model were apparent. In addition, it was repeatedly noted that the accumulation model provides better estimates of *k*, the learning asymptote, and that estimates of *R* also varied in an orderly manner. Thus, it was concluded that the accumulation model provides a better fit to the data than the replacement model. The theoretical significance of the findings was less clear. It is interesting to note that both the matching law and the accumulation model view behavior as a choice between appropriate and inappropriate behaviors.

3.0 Learning Curve Expressions

In addition to the power and exponential forms of the learning curve, many others have been derived. One of the best summaries of learning curve development is from Badiru (1992). In his review he highlights univariate as well as multivariate expressions of learning curve treatments. The univariate curves discussed were:

a. The log-linear model (Power model)(Wright, 1936)

b. The S-curve (Carr, 1946)

c. The Stanford-B Model (Asher, 1956)

d. DeJong's Learning Formula (DeJong, 1957)

e. Levy's Adaptation Function (Levy, 1965)

f. Glover's Learning Function model (Glover, 1966)

g. Pegel's Exponential Function (Pegel, 1969, 1976)

h. Knecht's Upturn Model (Knecht, 1974)

i. Yelle's Product Model (Yelle, 1976)

j. Multiplicative Power Model.

3.1 Univariate Models

Univariate learning curves involve a dependent variable described by an independent variable. The most notable univariate models are listed above and are explained below.

3.1.1 The Log-linear Model.

There are two basic forms of the log-linear model: the average cost function and the unit cost function. The average cost is more popular than the unit cost function. The relationship indicates that cumulative cost per unit will decrease by a constant percentage as the cumulative production volume doubles. The average cost model can be expressed as:

$$C_x = C_1 x^b$$

where

C_x - cumulative average cost of producing x units

C_1 - cost of the first unit

x - cumulative production count

b - the learning curve exponent (slope of the curve on log-log paper)

- ln (learning percentage) / ln 2.

When linear graph paper is used, the log-linear learning curve is a hyperbola. On log-log paper the curve is represented by a straight line equation:

$$\log C_x = \log C_1 + b \log x$$

where b is the constant slope of the line.

The percent of productivity gain is computed as:

$$p = 2^b$$

The expression for computing Unit Cost is:

$$UC_x = C_1 x^{(b+1)} - C_1 (x-1)^{(b+1)}$$

The Marginal Cost is found when taking the derivative:

$$MC_x = \frac{d[C_x]}{dx} = (b+1) C_1 x^b$$

The Total Cost is:

$$TC_x = (X) C_x = C_1 x^{(b+1)}$$

3.1.2 Other Univariate Models

Carr proposed an S-shaped learning curve function based upon an assumption of a gradual startup. The function has the slope of the cumulative normal distribution function. The gradual startup is based upon the fact that the early stages of production consist of transient states with normal changes in tooling, methods, materials, and even workers. The basic form of the S-Curve is:

$$MC_x = C_1 \left[M + (1-M)(x+B)^b \right]$$

where

M - incompressibility

B - equivalent experience units

The disadvantage associated with this model is that it requires determining the value of four parameters that can be found by fitting a cubic curve on a log-log plot. An example of such a cubic function is:

$$\log MC_x = A + B(\log x) + C(\log x^2) + D(\log x^3)$$

The Stanford-B Model was researched by the Department of Defense at Stanford University:

$$Y_x = C_1(x + B)^b$$

where

Y_x - direct cost of producing the xth unit;

C_1 - cost of the first unit when $B = 0$;

b - slope of the asymptote;

B - constant ($1 \le B \le 10$);

- equivalent units of previous experience at the start of the process;

- equivalent to the number of units produced prior to first unit acceptance;

When $B = 0$, this model becomes the log-linear model. Hoffman (1934) found that the inclusion of the B parameter resulted in smaller sums of squared deviations.

DeJong presented a power function that incorporates parameters for the proportion of manual activity in a task. DeJong's formula introduces an incompressible factor, M, into the log-linear model to account for the man-machine ratio.

$$MC_x = C_1\left[M + (1 - M)x^{-b}\right]$$

where when $M = 0$, this becomes the log-linear model, which implies a manual operation. In machine dominated operations, $M = 1$. One of the drawbacks associated with DeJong's model is the difficulty in estimating the parameters C_1 and b. M is assumed known.

Levy proposed the following model to account for the leveling of the production rate and the factors that may influence learning.

$$MC_x = \left[\frac{1}{\beta} - \left(\frac{1}{\beta} - \frac{x^b}{C_1}\right)k^{-kx}\right]^{-1}$$

where

β - production index for the first unit

k - constant used to flatten the learning curve for large values of x

The flattening constant, k, forces the curve to reach a plateau instead of continuing to decrease or turn in an upward direction.

Glover presented a model that incorporates a work commencement factor. The model is based on a bottom-up approach that uses individual worker learning results as the basis for plant wide learning curve standards.

$$\sum_{i=1}^{n} y_i + a = C_1\left(\sum_{i=1}^{n} x_i\right)^m$$

where

y_i - elapsed time or cumulative quantity

x_i - cumulative quantity or elapsed time

a - commencement factor

n - index of the curve (usually $1+b$)

m - model parameter

Pegel presented an alternate algebraic function for the learning curve. His model, a form of an exponential function, is represented as:

$$MC_x = \alpha a^{x-1} + \beta$$

where α, β, and a are parameters based on empirical data analysis. This is the marginal cost equation where Pegel assumes that the marginal cost of the first unit is known. Thus,

$$MC_1 = \alpha + \beta = y_0$$

This also leads to the equation for total cost expressed as:

$$TC_x = \frac{a}{1-b} x^{1-b}$$

where

x	- cumulative number of units produced
a, b	- empirically determined parameters

Knecht presented a modification to the functional form of the learning curve to analytically express the observed divergence of actual costs from those predicted by learning curve theory when units produced exceed 200. This permits the consideration of non-constant slopes for the learning curve model. If UC_x is defined as the xth unit, then it approaches 0 asymptotically as x increases. In the continuous case, the formula for cumulative average costs is derived as:

$$C_x = \int_0^x C_1 z^b dz = \frac{C_1 x^{b+1}}{(1+b)}$$

Knecht alters the expression for the cumulative curve to allow for an upturn in the learning curve at large cumulative production levels. He suggested the functional form below:

$$C_x = C_1 x^b e^{cx}$$

where c is a second constant. Differentiating the modified cumulative average cost expression gives the unit cost of the xth unit as shown below:

$$UC_x = \frac{d}{dx}\left[C_1 x^b e^{cx}\right] = C_1 x^b e^{cx}\left(c + \frac{b}{x}\right)$$

A model similar to Knecht's model is the plateau model described by Baloff. The plateau model assumes that production cost reaches a steady state at which point cost levels off.

Yelle proposed a combined product learning curve model for products by aggregating and extrapolating the individual learning curves of the operations making up a product on a log-linear plot. The model, which is similar to one of the cost estimating relationships presented by Waller and Dwyer, is expressed as:

$$C_x = k_1 x_1^{b_1} + k_2 x_2^{b_2} + \ldots + k_n x_n^{b_n}$$

where

C_x - cost of producing the xth unit of the product

n - number of operations making up the product

$k_i x_i^{b_i}$ - learning curve for the ith operation

Yelle's model contains several deficiencies, some of which are:

a. A learning curve formulated for aggregating several different learning curves (with different slopes) will not necessarily be a straight line on a linear plot.

b. A learning curve extrapolated from different learning curves will not necessarily be a straight line.

c. A product specific learning curve seems to be a more reasonable model than an integrated product curve.

3.2 Multivariate Models

Multivariate models have not received the attention they deserve in practice. This is perhaps due to the complexity of implementing the models for practical productivity assessment. Multicollinearity is also a major problem in implementing multivariate models. Some of the models are:

3.2.1 Popular Form

$$C_x = K\prod_{i=1}^{n} c_i x_i^{b_i}$$

where

C_x - cumulative average cost per unit for a given set of factor values.

K - model parameter (cost of first unit of the product)

x - vector of specific values of independent variables of the ith factor

x_i - specific value of the of the ith factor

n - number of factors in the model

c_i - coefficient for the *i*th factor

b_i - learning exponent for the *i*th factor

For simplicity and ease of analysis, the above model is often reduced to a bivariate model of the type:

$$y = \beta_0 x_1^{\beta_1} x_2^{\beta_2}$$

Alchian at RAND CORPORATION raised the following questions:

a. How long does the reduction in labor costs continue?

b. Can the reduction be represented by a linear function on double log scale?

c. Does the reduction fall at the same rate for all different airframe manufacturing facilities?

d. Can reliable predictions of marginal and total labor requirements be obtained for a given facility based on an average progress curve for all airframe manufacturers?

e. Can reliable predictions of labor requirements for a specific type of bomber be obtained from a curve fitted to the experience of all bomber production?

f. How reliable is a single plant's own experience for predicting later requirements for producing a particular type of airframe?

g. What are the potential consequences of errors in the predictions obtained from the progress functions?

Alchian considered progress functions containing other variables in addition to cumulative production *N*. This effort provided the first known report of experimentation with multivariate learning curves. The alternate functions presented below were used by Alchian to describe the relationships between *Direct Labor* per pound of airframe (*m*), *Cumulative Production* (*N*), *Time (T)*, and *Rate of Production* per month (ΔN).

a. $\log m = \alpha_2 + \beta_2 T$

b. $\log m = \alpha_3 + \beta_3 T + \beta_4 \Delta N$

c. $\log m = \alpha_4 + \beta_5(\log T) + \beta_6(\log \Delta N)$

d. $\log m = \alpha_5 + \beta_7 T + \beta_8(\log \Delta N)$

e. $\log m = \alpha_6 + \beta_9 T + \beta_{10}(\log N)$

f. $\log m = \alpha_7 + \beta_{11}(\log N) + \beta_{12}(\log \Delta N)$

Extensive study of the above equations failed to produce significantly better results than with the univariate model. The major reason for the lack of significant improvement with the multivariate functions is the fact that there is a high degree of correlation among the independent variables, *T*, *N*, and ΔN. The problem of multicollinearity (multiple correlation) is still a major concern in the use of multivariate learning curves. Multicollinearity normally implies that one or more of the correlated variables can be omitted without jeopardizing the fit of the model.

Another multivariate model is the Cobb-Douglas Multiplicative Power Model:

$$C = b_0 x_1^{b_1} x_2^{b_2} \ldots x_n^{b_n} \varepsilon$$

where

C - estimated cost

b_0 - model coefficient

x_i - ith independent variable ($i = 1,2,\ldots,n$)

b_i - exponent of the ith variable

ε - error term

In the above model, the disturbance term, ε, is defined as:

$$\varepsilon = e^u$$

where $u \sim N(0,\sigma^2)$ and independent of $X_1,\ldots,X_n$. Thus, the mean of the disturbance is given by $E(\varepsilon) = \exp(\sigma^2/2)$ while its median is $M(\varepsilon) = 1$. Waller and Dwyer presented a variation of the Cobb-Douglas function and an additive power model of the form:

$$C = c_1 x_1^{b_1} + c_2 x_2^{b_2} + \ldots + c_n x_n^{b_n} + \varepsilon$$

where c_i is the coefficient of the ith independent variable.

McIntyre introduced some general models to include a non-linear cost-volume-profit model for learning curve analysis. Incorporating a non-linear cost function that expresses the effects of employee learning affects the non-linearity in the model. McIntyre applied sensitivity analysis to the non-linear model to assess the impact of estimation errors in the learning rate and steady-state production time on estimated profit and breakeven quantities. Several forms of the non-linear model are presented.

a. Basic Profit Function

$$P = px - c\left(ax^{b+1}\right) - f$$

where

P - profit

p - price per unit

x - cumulative production

c - labor cost per unit of time

f - fixed cost per unit

b - index of learning

b. Multiprocess Model

$$P = px - nca\left(\frac{x}{n}\right)^{b+1} - f$$

where x is the number of units produced by n labor teams consisting of one or more employees in each.

c. Multi-skill Model

Different skill levels of employees produce different learning parameters between production runs.

$$P = p\sum_{i=1}^{n} x_i - c\sum_{i=1}^{n} a_i x_i^{b_i+1} - f$$

where a_i and b_i denote the parameters applicable to the average skill level of the ith production run and x_i represents the output of the ith run in a given time period. This is a model that can be used in modern manufacturing systems that call for simultaneous engineering.

Womer (1979) introduced a variable production rate model that incorporates cumulative production, production rate, and program costs. He presents a production function that relates output rate to a set of inputs with a variable utilization rate. The function is:

$$q(t) = AQ^{\delta}(t)x^{1/\gamma}(t)$$

where

A	- constant
$q(t)$	- program output rate at time t
$Q(t)$	- cumulative production at time t
δ	- learning parameter
γ	- returns to scale parameter
$x(t)$	- rate of variable resource utilization at time t

To optimize the discounted program cost, the cost function is defined as:

$$C = \int_0^t x(t)e^{-\rho t}dt$$

Womer (1984) also investigated the estimation of learning curves from aggregated monthly data. He cites the case in which production of an item(s) spans more than one month and involves labor in more than one month. In many instances, labor hours per unit may not be a previously tracked item. Thus, indirect labor hours can be mixed with direct hours. Womer describes this situation mathematically by looking at a lagged model. He begins with the model above and creates a special case of that model:

$$L_i = \alpha X_i^{\beta} U_i^{\gamma}$$

where γ is a parameter describing returns to scale in the production process. Womer prefers to think of α as the first unit production cost corresponding to a production rate of one per month. This produces a *Lagged* model of:

$$M_j = L_0 + \sum_{h=0}^{k} d_h X_{j+h}^{\beta} U_{j+h}^{\gamma}$$

This allows some of the labor hours incurred each month to be elements of fixed cost, independent of production or learning. The model has an interesting feature because as the model's variables begin to change the shape more the problems of the model are highlighted. Without these problems the model reduces to the ordinary unit learning curve.

Badiru (1991) introduced a bivariate model with cumulative production (x_1) and cumulative training time (x_2). The inclusion of training time as a second variable in the model seemed reasonable since many factors that influence learning can be expressed as training dependent variables. Thus, with the two-factor model, the expected cost of production can be estimated on the basis of cumulative production and cumulative training time. The mathematical expression for the learning curve was hypothesized as:

$$C_{x_1 x_2} = K c_1 x_1^{b_1} c_2 x_2^{b_2}$$

where

C_x = cumulative average cost per unit for a given set of factor values

K = intrinsic constant

x_1 = specific value of the first factor

x_2 = specific value for the second factor

c_i = coefficient for the ith factor

b_i = learning exponent for the ith factor

Badiru used a data set from a local manufacturing company. He took the log of the above cost equation to achieve a linear expression and then used statistical analysis software to fit the data to the model. The fit showed greater sensitivity to cumulative units than to training time. Additionally, the bivariate model provided a more accurate picture of transactions between the factors associated with the process. As a result, multivariate learning curves can provide mechanisms for the effective cost-based implementation of design systems, facilitate system integration of production plans, improve supervisory interfaces, enhance design processes, and provide cost information to strengthen the engineering and finance interface.

3.3 Additional Models

There are other models presented in the literature attempting to describe the learning process in an industrial environment. One of these is from Dar-El, Ayas, and Gilad (1995) where there are parallel phases presenting the learning process. They are cognitive and motor and represent the human information system that enables a person to perform a task.

Learning slope values published in the literature consequently apply to both cognitive and motor components of learning. What is observed in reality is the combined effect of these two processes. Thus, during the early repetitions. when cognitive processes dominate the learning curve, the learning constant b can be expected to have a high value; however, as experience is increased, the magnitude of b begins to drop towards a value nearer that associated with the motor process.

The following analysis shows that b, the learning constant, reduces monotonically as n, the number of repetitions, is increased.

Let b_c = the learning constant under pure cognitive conditions.

b_m = the learning constant under motor conditions

$y_c^{(1)}$ = the time for the first cycle under pure cognitive conditions

$y_m^{(1)}$ = the time for the first cycle under motor conditions

$b^*(n)$, b^* = the learning constant as observed in practice after n repetitions.

The value of b^* is a function of n. For convenience, $b^*(n)$ is simply written as b^*.

The observed learning curve becomes the sum of the pure cognitive and motor learning curves. Thus,

$$y(n) = \left\{y_c^{(1)} + y_m^{(1)}\right\} * n^{-b},$$

$$\left\{y_c^{(1)} + y_m^{(1)}\right\} n^{-b} = y_c^{(1)} * n^{-b_c} + y_m^{(1)} * n^{-b_m}$$

It is observed that the dual-phase learning model is described by four parameters: $y_c^{(1)}, y_m^{(1)}, b_c$, and b_m. Of these, b_c and b_m, once obtained, remain constant for all tasks, leaving two parameters to be determined, $y_c^{(1)}$ and $y_m^{(1)}$. Therefore,

$$b^* = b_c - \frac{\log\left(\left(R + n^{(b_c - b_m)}\right) / (R+1)\right)}{\log(n)}$$

The implication of the dual phase learning model is that b_c, the learning constant for the pure cognitive process, must be greater than b^*, which is the learning constant for both cognitive and motor learning. Solving for b_c:

$$b_c = -\frac{\log\left[\left(\frac{1}{R}\right)\left\{(R+1)n^{-b^*} - n^{-b_m}\right\}\right]}{\log(n)}$$

It should be noted that the proposed dual-phase learning model was developed after their experiments were completed. It was developed in order to explain the behavior of the observed data, because the simple power curve was totally inadequate as a theoretical predictor of the data. As a consequence, the experiments were not designed for testing the validity of the proposed model - rather, the actual data were checked to determine whether they supported the dual-phase learning model.

The two tasks used in the experiment by Dar-El, Ayas, and Gilad were basically a cognitive task (task 1) and a motor task (task 2). The following hypotheses were tested:

Hypothesis 1

The learning constant for task 1 is greater than the learning constant for task 2. The b_c learning constant is associated with pure cognitive activity whereas b_m represents the motor constant. The null hypothesis ($b_1 \leq b_2$). was rejected.

Hypothesis 2

The average learning constant for the first eight observations should exceed the average learning constant for the subsequent observations. Here again, this null hypothesis was rejected.

The dual-phase learning model requires the estimation of four parameters of which two, b_c and b_m, once estimated, remain constant for all tasks. The estimates of two parameters still remain; these are $y_c^{(1)}$ and $y_m^{(1)}$. In this respect, the dual-phase model retains the advantage of the power model, by requiring the estimates of only two parameters. All other learning models require three or more parameters to be estimated. In all fairness, the researchers pointed out that perhaps some other single learning model could better explain the behavior of the experimental results. Thus, the implication is that the learning constant cannot have a fixed value because it gradually reduces as experience is gained.

More recently, Zangwill and Kantor (1998) have addressed learning in a continuous improvement (CI) industrial environment. They attempted to connect the concept of learning with the concept of CI since the two appear to be symbiotic. They acknowledge the formality of the CI process by theorizing that CI can effect the process each cycle. As a cycle ends, management reviews the process for possible improvements and evaluates any improvements made previously. Thus, the process is not stable and worker learning can occur erratically with each cycle due to the changes made by management to the process. All previous attempts to measure learning assumed that the only thing that could result in a decreased production rate is forgetting. It is assumed that as a worker becomes more familiar with the process, the worker will make changes that will affect the production rate, discarding those that do not increase the resultant rate. This has been characterized as individual learning. Additionally, management and other workers could be studying the actual process in operation and suggest improvements as they observe worker performance. This has been termed organizational learning. CI introduces another longer term variable through a more formalized management input into the process so that methods and equipment may change constantly during the life cycle of the product. Work breaks and forgetting aside, this will produce some degradation in the over all production rate and produce new learning effects. All improvement becomes a function of management opportunity and effectiveness. Zangwill and Kantor noted the following summarized postulates:

Postulate I. *Other things being equal, for any metric $M(q)$, the rate of improvement is proportional to the non-value-added (NVA) component of the metric, $N(q)$. Specifically,*

$$M(q) = (otherfactors) x N(q)$$

Further, the NVA is specified as

$$N(q) = M(q) - Z^{*}$$

NVA is frequently construed as *any work* that does not contribute value to the final customer. NVA is defined as $N(q)$ above so that NVA is any "work" not required by the ideal operating process Z^{*}. Typically, NVA includes rework, changes, delays, erroneous information, defects, wastes, preparation time, transportation, idle time, and inspection. Also, NVA would be the work in making any items not sold. If $M(q) - Z^{*}$ is large, there are many opportunities to improve, so that the rate of improvement should be large.

Postulate II: *The rate of improvement is proportional to the effectiveness of management in a Volterra-Lotka form.*

Zangwill and Kantor rewrote the Volterra and Lotka observations as the rate of improvement in the metric that is proportional to:

(i) the effectiveness of management, $E(q)$.

(ii) the amount of metric remaining to be eliminated, $N(q)$.

The continuous improvement equation assumes that low values of the metric are better, and that Z^{*} is a constant. Following Postulate II, with c as the coefficient of proportionality:

$$dN/dq = -cE(q)N(q).$$

To measure improvement in any practical application, express this equation in finite differences:

$$\Delta N / \Delta q = -cE(q)N(q)$$

The last two equations are known as the "continuous improvement differential (or finite difference) equation" (CIDE). As $M(q) - Z^*$ gets small, $E(q)$ increases.

Postulate III. *The effectiveness of the CI effort depends upon the amount of the improvement remaining, according to a power law. Specifically, for a parameter, κ, and coefficient, K,*

$$E(q) = KN(q)^{\kappa}$$

As improvements are made, $N(q) = M(q) - Z^*$ is decreasing. If the parameter $\kappa > 0$, the effectiveness of management is decreasing, while $\kappa < 0$ represents increasing effectiveness, and $\kappa = 0$ is constant effectiveness of management.

Postulate IV. *Additive Decomposition of Metric. The metric M(q) of a process can be decomposed into sub-metrics $M_i(q)$, which are metrics for steps of the process, and whose sum is the metric M(q).*

$$M(q) = \Sigma M_i(q)$$

Utilizing Postulate IV, the exponential and the power law become intimately related. The power law of improvement can be represented by a sum of exponentials, or in the limit, as the integral of exponentials.

$$(1 + q_{\min})^p (q + q_{\min})^{-p} = (1 + q_{\min})^p \int_0^{\infty} \frac{1}{\Gamma(p)} e^{-rq_{\min}} r^{p-1} e^{-rq} dr$$

This equation suggests that, at least in principle, the total metric $M(q)$ could improve as a power law, while the component parts of that metric, $M_i(q)$, improve as exponentials.

It is also true that the exponential can be approximated as a sum of finite forms (Kantor and Zangwill, 1996). The exponential can be written

$$e^{-rq} = \int_q^{\infty} (1 - q / q_{\max})^d \frac{1}{\Gamma(d+1)} x q_{\max}^d r^{d+1} e^{-rq_{\max}} dq_{\max}$$

The three functional forms (exponential, power, and finite) appear to compose a hierarchy. Sums of the finite form can well approximate the exponential form, and sums of exponentials can well approximate a power law. However, the reverse is not true. Since the finite form can generate the other two, but not conversely, the finite form seems the most fundamental.

With regard to the coexistence of the power law and exponential forms of learning, the differential equation shows that these may belong to the same family as a new law of learning, the finite form. In industrial situations, the power and exponential laws might be sums of finite forms, suggesting again that the finite law may be more basic (Zangwill and Kantor, 1998).

4.0 Forgetting

There is simply no generally accepted theory of forgetting to which psychologists can appeal for understanding or even for organizing the facts associated with forgetting. Psychologists use the term forgetting in a descriptive sense to refer to any decrement in performance that occurs in a retention test. Notice that this says nothing about the integrity of memory. If a memory is retrieved and so manifested in behavior, experts assert that the memory is in storage, but they can never really assert that a memory is not in storage when retrieval does not occur.

There is a set of ideas about memory retrieval that can provide a general conceptual framework as to how remembering is governed. The conceptual framework consists of a major assumption and three control ideas. The assumption is that whatever is learned - all memories acquired by an individual - is permanent as long as he or she remains neurophysically intact. The first control idea is that a memory is most likely expressed when the circumstances in which it is to be remembered are most like those in which it was originally learned. The second idea is that a multi-dimensional memory which is acquired when learning an episode, consists of a group of attributes that represent specific stimuli and responses of a central, target task as well as features of the context external and internal to the learner. The final idea states that a memory will be retrieved and manifested in behavior when a sufficient number, kind, or percentage of the attributes of a memory are aroused by the occurrence of events sufficiently similar to those that had been represented by the attributes. Retrieval of a memory is the process through which memory attributes are taken from storage and become active with the potential of affecting behavior.

Herman Ebbinghaus (1913) was one of the first to study retention and record his findings for mathematical interpretation. He learned lists of "nonsense syllables" and tested his own retention and relearning rate. He noted that retention was "negatively accelerated", that is the rate of forgetting slows over time.

Numerous studies have proven that during a break there is a loss of recall of previously learned information. Also, there appears to be a direct correlation between recall loss and the length of the break. It has been theorized by many including Womer that recall is enhanced with recall loss minimized if activities during a break are similar to the previously learned information. Some researchers have also shown that when learning cognitive activities, recall or performance after a break may be enhanced. This phenomenon, it is assumed, results when cognitive activity regarding the task does not cease, but continues during the break, sometimes resulting in improved understanding of the task and an improved application when the task is resumed.

4.1 Reminiscence and Incubation

Psychologists have determined that a break does not always result in a loss of recall. They have used the term reminiscence for many years to refer operationally to cases in which performance on a retention test is better after long rather than after short break intervals. In other words, reminiscence is increased recall of previously forgotten items, i.e., items that previously were not recalled. Hypermnesia is a term denoting improved retention beyond the normal levels of original learning, a consequence of a specific treatment.

In some studies there has been an interesting phenomenon appear in which retention initially decreases and then begins to increase over time with an eventual decay. This is known as the Kamin effect and is extremely rare. Some cases of this kind seem best attributed to factors that exert a relatively trivial performance-based impairment on retention shortly after learning so that retention later is better only relatively speaking. None the less, there is no conclusive body of evidence to date that proves that memory strength progressively grows during a period of relative inactivity. But the case is not closed, particularly for the poorly understood effects of arousal on retention.

The general lesson is that however appealing the concept of reminiscence might be, it is unwise to believe that an acquired memory will ever become more permanent by the mere passing of a retention interval. General Hypermnesia of the kind discussed earlier can be traced either to a special impairment in retention performance after short intervals (as opposed to enhanced retention after longer intervals) or to active processing during longer periods. This is not true of all memory as is discussed in reference to incubation. Incubation is much like reminiscence except that it is applied in instances involving fear or a heightened emotional response where the normal decline of recall does not occur. In fact, as time passes the recall of an event may become more pronounced and even distorted because of the loss of actual event memory without the loss of the emotion evoked by the memory.

4.2 Sources of Forgetting

A source of forgetting is an event that either intervenes between the time of learning and the test or occurs prior to the point of learning and reliably is accompanied by forgetting. It has been found that allowing a long interval to elapse between learning and retrieval normally results in forgetting. Thus, a long interval can be a source of relational forgetting.

Many authors have seen a variety of instances in which the expression of learning depends on faithful reproduction of the context of that learning, even though that context might seem irrelevant to the particular associations or relationships that were learned. Said another way, a change in contextual circumstances frequently results in forgetting.

Another major determinant of retention and forgetting is interference. Interference can be thought of as the effect produced by conflicting memories. Memories that share attributes conflict if the attributes contain opposing information critical to behavior. The two primary types of interference are retroactive and proactive. Retroactive refers to forgetting caused by conflicting memories acquired between acquisition and testing of the critical memory. Proactive interference is less intuitively apparent as a source of forgetting since the loss of the memory of interest is based upon previous learning. Interference is based upon similarity of memory attributes. The more separate the occurrence attributes, the less the opportunity of interference.

4.3 Previous Research in Forgetting

While extensive literature abounds on the subject of learning, few rigorous analytical studies of the effect of forgetting can be found. It only seems reasonable that if there is an interruption in the learning process, a break in the production process occurs after learning has plateaued, or some forgetting occurs during the learning process, forgetting must be modeled in conjunction with the learning process itself.

One of the first works performed in forgetting was done by Sule (1963) who studied the effect of alternate periods of learning and forgetting upon the Economic Manufacturing Quantity. He assumed that whenever job interruption occurs, it results in some forgetting. He interpreted this as a subsequent drop in the production rate at the beginning of the next production cycle depending on the length of the interruption and production rate when the interruption occurred. The relationship is:

$$Y_F = X_F R_F^{B_F}$$

where

Y_F - number of units that could be produced on the Rth day

X_F - equivalent production on the first day of the forget curve

R_F - cumulative number of days in forget cycle and $B_F = 3.32 \log (\% \text{ forgetting}) / 100$.

While the effect of learning on EOQ has been considered in the literature, the effect of forgetting has been largely ignored. Sule gives a procedure based on the above discussion for determination of the optimal order quantity for the finite production model that incorporates both effects. This generalized extension for learning and forgetting, however, does not exclude the traditional case of no learning and no forgetting, in which case, B_L and B_F have zero value. The approach taken is to adjust the model for continuous production by using the performance at the point of resumption of production as the restart value. Sule uses this methodology to find the optimum value of Q that minimizes yearly cost.

Sule presumes that learning occurs during production and that forgetting ensues as soon as production breaks. Forgetting then begins to erode the production rate achieved at the end of production until a new production cycle is initiated. He uses the forgetting rate for the non-production period to compute production rate loss and subtracts that from the rate at the end of the last production cycle. This resulting value becomes the production rate for the beginning of the next production cycle. In Sule's words, "the production rate on the last day of the first forget cycle is equal to the production on the day production resumes in the second production cycle." Sule iterates this way for subsequent cycles. He then uses this derivation of production rates and re-addresses the EMQ cost formulas applying the learning and forgetting information. He also assumes that at some point a steady state occurs at which the time to produce Q units is t. Thus, he assumes that at some point learning and forgetting will have no effect upon the production rate and the cost to produce Q items. His generalized formulas for computing the time of forgetting are:

$$X_{Fn} = X_L\left(t^{n-1} + t_{1n}\right)^{-(B_L + B_F)}$$

$$t_{1n} = \sqrt[1-B_L]{\frac{(1-B_L)Q}{X_L} + \left(t^{n-1}\right)^{1-B_L}} - t^{n-1}$$

$$t^n = \sqrt[B_L]{\frac{X_L}{X_{Fn}\left(t^{n-1} + T\right)^{B_F}}}$$

Sule then applies his cost model against various Q values and computes the annual costs. He shows that production time steadily decreases and converges to a steady state value. Thus, he shows that for a given learning rate, as the forget rate decreases, the time required to produce Q units decreases.

Several other models have been suggested, most of them based on theoretical assumptions not supported by empirical experiments regarding the forgetting process. Steedman (1970) and Carlson and Rowe (1976) claimed that mainly the length of the break and the performance time just before the break affects the forgetting process. Globerson and Levin (1995) argued that forgetting is a complex process affected by several factors including turnover, communication, and documentation. Womer (1984) agreed that production breaks can cause forgetting. He argued that breaks can cause forgetting if during the breaks the workers are involved in non-complementary operations, the learning process of which counteracts the preceding learning process. Thus, interference can occur resulting in loss of recall.

Carlson and Rowe (1976) used the Learn-Forget-Learn model to assess the cost of forgetting. They describe learning in three phases: the incipient phase in which very little learning occurs, the second phase during which real growth occurs, and then the final phase in which the learning rate really slows and even becomes asymptotic in the limit. Carlson and Rowe stated that forgetting, or the interruption portion of the learning cycle, can be described by a negative decay function comparable to the decay observed in the electrical loss in condensers. Some of the more notable conclusions drawn by Carlson and Rowe were:

1) The initial rate of learning is a function of the amount and proximity of prior experience.
2) Some forgetting may always be expected. Total forgetting does not occur within short periods of interruption.
3) Forgetting curves show rapid initial decrease in performance followed by a gradual leveling off as a function of the interruption interval period.
4) The rate and amount of forgetting decreases as an increased number of units are completed before an interruption occurs.

The amount of forgetting and the corresponding level of performance are thus a function of both the performances at the time the project was interrupted and the length of the interruption. Of course, motivation should be added to the list of factors because the rate of learning is correlated with worker attitude and cooperation.

Cochran (1968) concluded that retrogression is a function of the quantity produced to date and the length of the interruption with all other variables remaining the same.

The Learn-Forget-Learn analysis procedure can be summarized as follows:

1. The original learning curve data can be fitted with an appropriate model and a table is generated.

2. Interruptions can be assumed to take place at any point and the quantity at this point together with an assumed forgetting slope yields a model for forgetting. A table can be generated showing the performance degradation for retrogression as a function of the interruption interval.

3. At the point of resumption, the performance or unit time can be used to determine the restart point on the original experience curve (called back-up by Bailey).

The model for learn-forget-learn is:

$$T_1 = (S / P) * Z^L$$

where:

T_1 - Intercept for a given learning curve

S - Standard Time

P - Performance Level

Z - Reference Quantity

U_i - Units at Q_i

and

$$L = \log(learning\_rate) / \log(.5).$$

The expected time per unit is thus:

$$T(X) = T_1 / X^L$$

where X goes from unit 1 to Q_1 the point at which the interruption begins. If a forgetting slope of 80% is used this would mean that ultimately only 80 percent of what was learned would be retained.

$$T_2 = U_1 / Q_1^F$$

where

$$F = \log(.80) / \log(.5)$$

Summarizing:

Learn:

$$T(X) = T_1 / X^L \qquad \text{for} \quad 1 \le X \le Q_1$$

Forget:

$$T(X) = T_2 * X^F \qquad \text{for} \quad Q_1 \le X \le Q_2$$

where:

$$T_2 = U_1 / Q_1^F$$

Resume Learn:

$$T(X) = T_1 / (X - Q_2 + Q_3)^L \quad \text{for} \quad Q_2 \le X \le Q_3$$

where

$$Q_3 = (T_1 / U_2)^{1/L}$$

If the cubic curve is used, the notion that the rate of learning approaches a plateau after a large number of repetitions could be incorporated. The cubic curve appears to be more appropriate than the log-linear curve because it conforms more to psychological learning theory. The three curves used in the study were:

$$Y = A + BX + CX^2$$

$$Y = A + BX + CX^2 + DX^3$$

$$Y = A + B\log X + C(\log X)^2 + D(\log X)^3$$

Bailey (1989), in his review of the relevant psychological literature, pointed out that according to psychological research: (a) the most important factor affecting forgetting is the level of learning achieved before any break in the learning process; (b) the learning rate and the complexity of the task do not influence the forgetting rate; (c) the time required to achieve the original performance level when relearning is much less than the original learning period; and (d) the learning rate is highly correlated with the time taken to complete the first repetition. Bailey also found that there is a high correlation between amount learned and the learning curve rate which weakens the finding that forgetting is a function of amount learned and not of learning rate.

Bailey noted that there is a distinction between "procedural tasks" versus "continuous control" tasks when performing them manually. Procedural tasks consist of a series of discrete motor responses such as pressing buttons or loading computer tapes. Continuous tasks are those repetitious movements with no clear beginning or end such as swimming, bicycling, or inspecting the same items coming off a production line one after another. Bailey found that procedural tasks involve deciding what to do, not just learning a motion and thus over short retention intervals forgetting occurs. The same was not true of continuous tasks that show little or no forgetting, even over extended retention intervals.

Bailey and McIntyre (1997) realized that the best fit to a set of data may not always result in the best predictor in the future and that there is little evidence of the problem or the size of the problem in learning curves. They used three learning curves and nine relearning curves, especially emphasizing the log-log-linear model that they introduced in 1992. Backing-up was applied to each of the learning curves by using the original learning curves to find the value of x that corresponds to the actual restart time (the time required for the first post-break iteration). The three learning curves used were:

$$A = ax^b$$

$$M = ax^b$$

$$\log M = a[\log(x+1)]^b \qquad \text{(log-log linear model)}$$

The third equation is the log-log-linear model introduced in 1992. The researchers then looked at each of the learning curves as relearning curves by varying their application of the parameters. All were reviewed under the following conditions:

1) Setting new parameters.
2) Backing-up and using new parameters.
3) Backing-up and using the original parameters.

The third category back-up was the same as category two in that the same values of x were used but the original learning curve parameters were used to predict post-break performance.

The three methods were compared using goodness-of-fit and the mean absolute percentage error (MAPE) for marginal times. In one of their previous writings, Bailey and McIntyre discovered that the goodness-of-fit of relearning curves varied with the degree of forgetting. They used the same measure of forgetting as earlier, namely:

$$PROLST = \sum_{i=1}^{5}\left(MT_i - ET_i\right)/\left(ET_1 - ET_{m+1}\right)$$

where

MT_i = actual marginal time of the ith post-break iteration

ET_i = learning curve estimate of marginal time for the ith iteration based on one pre-break performance, and

m = number of iterations before the break

The log-log-linear curve was the best predictor in the pre-break period. Results showed that the length of the break was not as significant as a factor affecting the level of forgetting. Thus, Bailey and McIntyre concluded that factors other than the passage of time are the primary causes of forgetting in long term memory. This result was consistent with the interference theory of forgetting explained by Greene (1992).

During Bailey and McIntyre's laboratory experiment involving data from 55 students, it was seen that goodness-of-fit was not a reliable indicator of predictive ability for relearning curves. After the analysis of relearning curves it was concluded that relearning curves that start anew are better predictors than those that back-up a learning curve. Lastly, Bailey and McIntyre, considering that the log linear average time model and the log-log-linear marginal time model are both convex in a plot of marginal time against the log x, recommend that the log-log-linear model be tested as well as the classical average time model (Bailey and McIntyre, 1997).

Sparks and Yearout analyzed the learning process after the break and the amount of forgetting during the break. The results revealed that traditional curves developed for mechanical tasks do not reflect the forgetting that occurs in more typical modern industrial tasks.

Goonetilleke et al. (1995) argued that the learning process of complex tasks is divided into two stages: the mental model development stage and the optimization stage. In the mental model development stage, the operator learns the dynamics of the task. During this stage large fluctuations in performance are observed. The optimization stage begins when the operator acquires a good mental model of the task, at which time the fluctuations decrease markedly and a significant performance improvement begins. The performance fluctuations are small when a perfect understanding of the task is acquired. According to Goonetilleke, the performance curve of the optimum stage is the known learning curve. They concluded that to learn a task well one should 1) develop a good mental model of the task, and 2) understand the optimization factors of the task.

Arzi and Schtub (1997) looked at the differences between learning and forgetting in mental and mechanical tasks. They tested the hypothesis that there was no difference between the mental group and the mechanical group in forgetting due to length of break. They could not prove that there was a statistical difference. Thus, the break length has no significant effect on the coefficient's values in the mental or mechanical forgetting functions. They also hypothesized that there is a difference between the learning process before the break and the relearning process following the break. The conclusion of this test was that the learning process and the relearning process are significantly different. In all tests of break duration, H_o could not be rejected at a 95% confidence level. Hence, in the mental task, the effect of a break on performance in the steady state part of the learning process was not significant. In addition, although not significant, the intensity of forgetting decreased as the break length increases. This unexpected result, which is a consequence of the improvement in performance during the break, is in line with the lack of correlation between the break length and the intensity of forgetting in mental tasks. In tasks measured in their work, mechanical and mental, no linear correlation was found between each of the measures of forgetting intensity and the parameters of break length and the coefficient of the learning process before the break. Hence, in both task types, as the learning rate decreased the forgetting intensity increased. This result points out that the ability of an individual to remember (to forget less) is correlated with learning capability (Arzi and Shtub, 1997).

In both tasks, mechanical and mental, the cumulative exponential model was a good model for the learning process. The break during the learning process of a mechanical task caused a significant forgetting effect. The forgetting intensity is affected significantly only by the rate of learning before the break.

On average, the forgetting effects of the mental tasks were larger than those of the mechanical tasks (about 25%). Forgetting in mental tasks appears to be a complex process, which is affected by factors related to an individual's capabilities and motivation. An important and unexpected finding is that for mental tasks, the break duration has an unpredictable influence on the forgetting intensity. For many individuals, the break caused an improvement in performance with the longer breaks producing the most improved performance.

The above discussion may lead to practical applications in industry. First, following the conclusions of Goonetilleke *et al.*, it seems that a teaching program of a mental task must concentrate on the method of decision making associated with the task and not on the technical aspects of the task. Secondly, in mental tasks related to emergencies (e.g., tasks performed by soldiers, firemen, hazardous process controlling), where long breaks often exist between repetitions, it is essential to use the break for training. Without a proper training program, forgetting will occur and the performance in the first, and the only required repetition after a break may be low. By using the breaks for training, the performance may improve, especially if the mental task must concentrate on the method of decision making with the task and not all aspects of the task.

Elmaghraby (1990) defines his model as the variable regression to invariant forgetting (VRIF) model that assumes there is a unique forgetting function with a single intercept point. The rationale for this hypothesis is that the intercept and the forgetting slope are system-dependent parameters similar to the same variables in the normal power function. Elmaghraby defines the VRVF (variable regression to variable forgetting) model where the intercept of the forgetting function varies for each lot. These are the models used by Sule as well as Carlson and Rowe. The VRIF and VRVF models adopt a fixed forgetting slope. Refer to Fig. 1 for a visual description of the model. The Jaber and Bonney (1997) learn-forget curve model (LFCM) modifies the VRVF model by using a forgetting slope that is not fixed. The VRVF model is defined by equating the time required to produce the $q_i th$ unit on the learning curve to time required on the forgetting curve. It is assumed that after making q_i units, the process is interrupted for a period of length t_{bi}, during which additional s_i units would have been produced. Thus, the time required to produce unit number $q_i + s_i$ by the end of the break period on the forget curve is expressed as:

$$\hat{A}_{q_i+s_i} = \hat{A}_1 (q_i + s_i)^f$$

where $\hat{A}_1$ is the equivalent time for the first unit (intercept) of the forget curve.

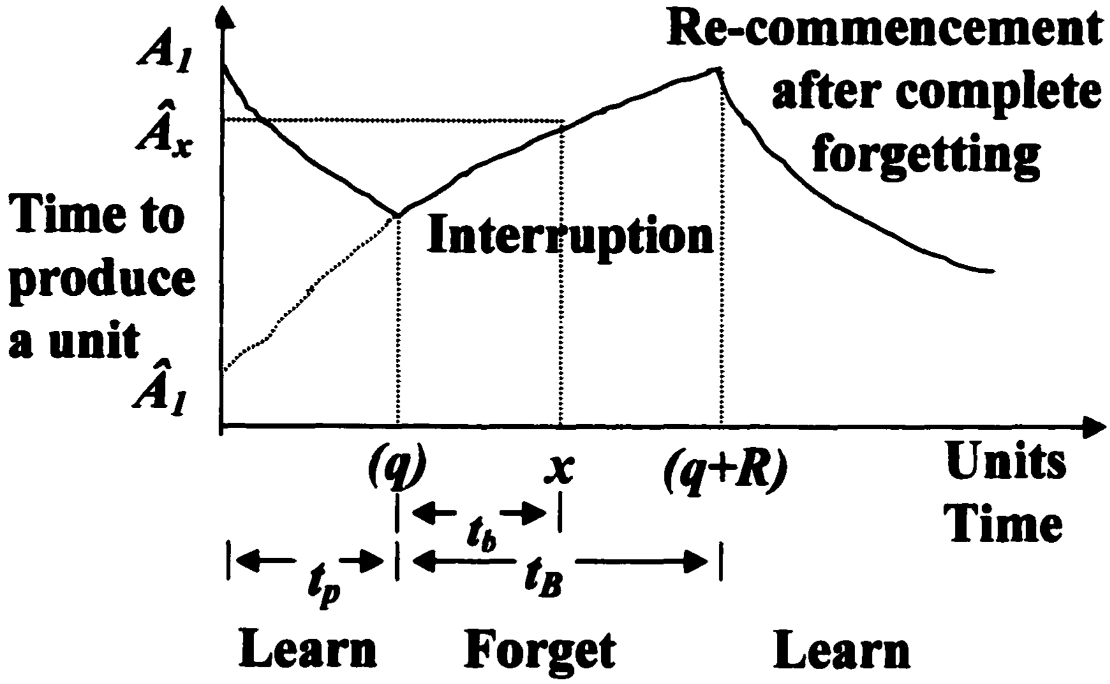


Fig. 4-1. Time to produce a unit versus units of output for an interrupted operation.

t_p is the time in production to produce q units

t_B is the minimum time for total forgetting.

R is the potential additional quantity that would be produced if no interruption occurred.

(q), $(q + R)$ are the number of units produced or which can potentially be produced in times t_p and $t_p + t_B$ respectively.

x is the number of units produced or which can potentially be produced in times t_p and $t_p + t_b$, where $x \leq q + R$ and $t_b \leq t_B$

The variable regression to variable forgetting (VRVF) model was devised by Carlson and Rowe who assumed that the intercept of the forget curve varies after each interruption based on the number of units produced in that cycle.

$$\hat{A}_{q_i+s_i} = \hat{A}_{1i}(q_i + s_i)^f$$

Elmaghraby argued that the value of $\hat{A}_{q_i+R_i}$ does not converge to a unique value A_1 when total forgetting is assumed in the VRVF model, given that t_B represents the minimum interruption to which total forgetting is assumed. If the production process is interrupted for t_B units of time, then the time to produce the first unit after the interruption is A_1 (i.e., $t_{bi} = t_B$ and $s_i = R_i$). In the case of Elmaghraby who assumed that $\hat{A}_1$ and f are constant for all cycles i:

$$\hat{A}_{q_i+R_i} = \hat{A}_1(q_i + R_i)^f$$

while in the case of Carlson and Rowe who assumed that $\hat{A}_{1i}$ is calculated for each cycle i while f is fixed for all cycles, then:

$$\hat{A}_{q_i+R_i} = \hat{A}_{1i}(q_i + R_i)^f$$

Jaber and Bonney expressed the forgetting rate in their learn-forget curve model (LFCM) as:

$$f_i = \frac{\ell(1-\ell)\log q_i}{\log(C_i + 1)} \text{ ; where } i = 1,2,3, \ldots$$

and f_i, which varies in every cycle, is the forgetting slope after interruption in cycle i, l is the learning slope, and $C_i = t_B / t(q_i)$ is the ratio of t_B, the minimum time for total forgetting, to $t(q_i)$, the amount of time required to produce q_i units. $t(q_i)$ is determined by migrating from the power function and C_i is represented as:

$$C_i = t_B \left[\frac{T^1}{1-\ell} q_i^{\,1-\ell} \right]^{-1}$$

Elmaghraby as well as Carlson and Rowe did not explain why they selected a doubling factor for the forgetting slope in their VRIF and VRVF models, and it was assumed fixed for all cycles. The LFCM does allow for an estimation of the learning and forgetting slope.

The VRIF, VRVF, and the LFCM hypothesized two relationships in defining the learning-forgetting relationship. The first hypothesis indicates that when total forgetting occurs, the performance time on the forgetting curve reverts to a unique value equivalent to the time required to produce the first unit with no prior experience. The second hypothesis is that the performance time on the learning curve equals that on the forgetting curve at the point of interruption. The LFCM was the only one that satisfied both hypotheses. None of these models have been validated with field data (Jaber & Bonney, 1997).

In Badiru's (1994) studies, there are three potential cases for the occurrence of forgetting:

1. Forgetting occurs continuously throughout the learning process.

2. Forgetting occurs only over a bounded time interval.

3. Forgetting occurs over intermittent time intervals where the time of occurrence and the duration of forgetting are described by some probability distributions.

If learning and forgetting start at a particular performance level, the resultant performance function, $r(t,u)$, may be modeled as

$$r(t,u) = (l(t,u) - \frac{[l(t,u) - f(t,u)]}{2} = \frac{l(t,u) + f(t,u)}{2}$$

where $l(t,u)$ is the learning function and $f(t,u)$ is the forgetting function. This is a point-by-point average of the learning and forgetting functions. Thus, the forgetting curve causes a downward pull on the learning curve that results in the resultant function.

In the Badiru research it is noted that the two functions, $l(t,u)$ *and* $f(t,u)$, do not necessarily start at the same performance level since the performance level at $t = 0$ for $l(t,u)$ is much higher than the expected performance at $t = 0$ for $f(t,u)$. Thus, alternate resolution approaches need to be investigated for this type of function. Badiru presents the definition:

> *Coincident Learning and Forgetting Functions:* A learning function and a forgetting function are said to be *coincident* if both functions originate at the same performance level.

Badiru presents an example wherein the two functions were not considered to be coincident. In that case, the forgetting function was raised to a point equivalent to the lowest point of the learning curve. This common point was referred to as the incipient performance level, P_i. The forgetting function was merely adjusted vertically to the incipient point.

5.0 Experimental Model and Methodology

As mentioned in the introduction, this research is a look into the effects of forgetting on learning in the scenario where forgetting occurs intermittently during the learning process for various lengths of time. This is later applied in a cyclic production system. Numerous other investigators have studied scenarios associated with the effects of forgetting only during breaks in the learning period. This study addresses forgetting that occurs in conjunction with learning. Thus, the total amount of information available for recall is a resultant of the interaction of learning and forgetting acting simultaneously. With forgetting assumed to be an intermittent process during learning, there will be times when only learning is occurring and times when the resultant of learning minus forgetting will occur. Borrowing from Badiru, relative bivariate functions describing learning, $l(t,u)$, and forgetting, $f(t,u)$, were used in the study. The variables were all representative and did not reflect a fit to an actual data sample. They were:

$$l(t,u) = 20t^{0.09} + u^{-0.05} \tag{5-1}$$

and

$$f(t,u) = t^{-0.20}u^{-0.30} \tag{5-2}$$

for $0 \leq t \leq 2000$ and $10 \leq u \leq 1500$ where t represents time and u represents production units. It was assumed that learning and forgetting begin at a particular performance level, the threshold point, at which the resultant function, $r(t,u)$, may be modeled as:

$$r(t,u) = l(t,u) - f(t,u)$$

$$= 20t^{0.09} + u^{-0.05} - \left[t^{-0.20}u^{-0.30}\right] \quad (5\text{-}3)$$

The resultant function, $r(t,u)$, represents the performance due to learning with forgetting. A continuous function is used in this scenario to describe learning over time with no production breaks interfering with the learning process. Intermittent forgetting was introduced throughout the process. Thus, this research did not initially have to be concerned with the learning rate to be utilized at the beginning of each production cycle when learning would be resumed after a break and some forgetting had occurred.

In order to determine when forgetting would begin and end, a single arrival single server queue was simulated. The arrival of entities was input as a range and a random, uniformly distributed number was generated within that range as the interarrival rate for that entity. The time between arrivals was assumed to be distributed exponentially. The service time for the server was input as a mean of an exponentially distributed service. This made the system an M/M/1 queue. The system was run and when a queue formed behind the server, forgetting began and continued as long as there were any entities in the queue. When the queue was cleared, forgetting was terminated until a queue began to form again.

In order to provide some smoothing during the transition from one state, learning, to another, learning plus forgetting, a weighted moving average was applied where:

a) α = relative contribution to the resultant at τ time units earlier

b) β = relative contribution to the resultant at ψ time units earlier

c) γ = relative contribution to the resultant at the present time t

with τ and ψ varying depending upon the time sample increment, and the restriction that

$$\alpha + \beta + \gamma = 1.0 \tag{5-4}$$

then,

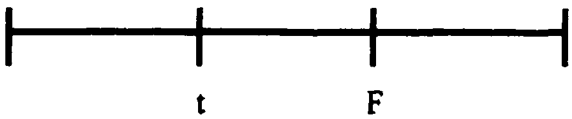
$$\begin{aligned} R(t,u) = {} & \alpha\left[l(t-t_2,u) - f(t-t_2,u)\right] + \beta\left[l(t-t_1,u) - f(t-t_1,u)\right] \\ & + \gamma\left[l(t,u) - f(t,u)\right] \end{aligned} \tag{5-5}$$

Substituting provides,

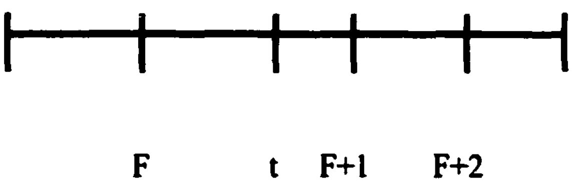
$$\begin{aligned} R(t,u) = {} & \alpha\left[20(t-t_2)^{0.09} + u^{-0.05} - (t-t_2)^{-0.20}u^{-0.30}\right] \\ & + \beta\left[20(t-t_1)^{0.09} + u^{-0.05} - (t-t_1)^{-0.20}u^{-0.30}\right] \\ & + \gamma\left[20(t)^{0.09} + u^{-0.05} - (t)^{-0.20}u^{-0.30}\right] \end{aligned} \tag{5-6}$$

This presented an interesting situation because not all time computations would involve the same resultant curve. The smoothing computation regressed one and two incremental periods backward. Where the points fell as it backed up would determine whether forgetting had to be applied or there was just learning continuing. Thus, three separate scenarios had to be modeled and tested to ensure the applicable function was employed during smoothing. The forgetting periods in the display below are the queue beginning and ending times. Prior to F, there is no forgetting occurring. After F, forgetting occurs as long as the queue is in place. When there is no longer a queue, then forgetting terminates until there is a queue established once again. The simulation program generates queue initiation and termination times that were used as the beginning and ending of a forgetting period. The objective was to generate enough points to adequately describe the overall learning process. The three scenarios mentioned above were:

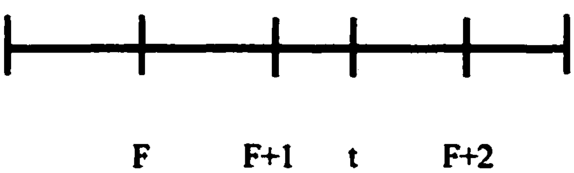
Scenario One. No forgetting is occurring. Forgetting does not begin until time F.



Scenario Two. Forgetting begins at time F and continues until time F + 1. Thus, at time t, forgetting is occurring and times t-10 and t-20 must be evaluated to determine the proper resultant function (Assuming $\tau = 20$ and $\psi = 10$).



Scenario Three. Similar to scenario two, times t-10 and t-20 must be evaluated to determine the correct resultant function to apply. At F + 2, forgetting begins again.



Thus, the resultant must be computed for t, t-10, and t-20 with smoothing applied to each time element for its contribution to the total resultant value.

Runs and *runs above and below the mean* tests were executed on the random number generator (Appendix A) for the simulation. In both cases, the null hypothesis of independence of sequential variables could not be rejected. The analysis is shown in Appendix A along with the computer code. The library routine used was that of Borland C++ version 4.52.

The simulation was also reviewed to determine an appropriate warm-up period to be used in the generation of the queue times. In order to study the resultant curves it is necessary to develop various queue length distributions of data. Thus, the effect of short compared to long queues could equate to short and long periods of forgetting. A review produced graphs as shown in Fig.s B-1 through B-6 in Appendix B. From the graphs of various queue lengths it appears that system steady state was reached after 400 time units. At that point, all graphs showed that the average queue length had been reached and that the length of queues had high variability. At no point did the system appear to go into a non-stable state. Thus, for all subsequent runs producing queues to be used in the analysis, a warm-up period of 400 units was used to ensure post initiation behavior.

Before commencing the computer portion of the study, an analysis was made of the resultant function and how it reflected learning as a result of time, units produced, and forgetting. The first two factors of Eq. 5-3 relate to learning. The first factor shows the contribution of time and the second factor the contribution of units. The exponents of each of these variables were essential in describing the rate of contribution of each. Referring to Table 5-1 below, it appears that time is the significant contributor to learning with learning approaching a near resultant asymptote of 50 with the greatest amount of learning occurring in the first 400 units (Table 5-1 and Fig. 5-1).

Time	Units	Time Only	Units Only	Resultant
10	10	24.605	0.8505	25.497
20	20	26.189	0.847	27.050
40	40	27.875	0.845	28.706
60	60	28.911	0.844	29.726
80	80	29.669	0.843	30.784
100	100	30.271	0.842	31.065
150	150	31.396	0.841	32.175
200	200	32.219	0.839	32.987
250	250	32.874	0.839	33.632
300	300	33.417	0.838	34.169
400	400	34.293	0.837	35.035
500	500	34.989	0.836	35.722
1000	1000	37.241	0.834	37.949
2000	2000	39.639	0.831	40.322
4000	4000	42.191	0.824	42.851
6000	6000	43.759	0.827	44.406
8000	8000	44.906	0.826	45.455
10000	10000	45.817	0.825	46.448

Table 5-1. Contributions of Time and Units to Resultant

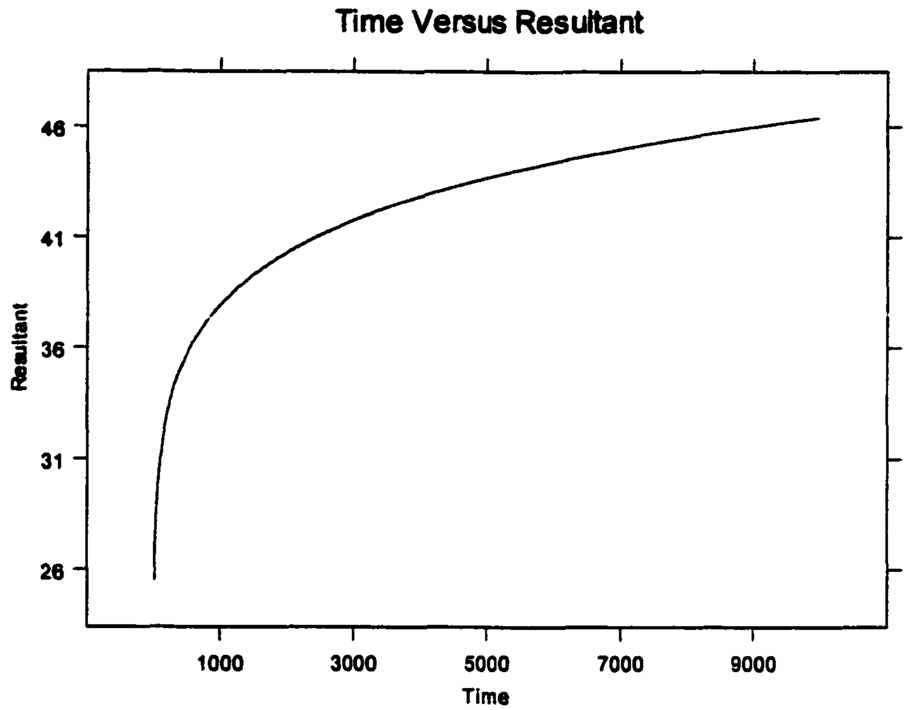


Fig. 5-1. Resultant as a Factor of Time Only

An analysis of the coefficients of the forgetting portion of the resultant curve, the third factor in Eq. 5-3, was also conducted. The third factor represents the forgetting portion of the equation with units and time contributing as a combined entity. Of course, the powers of the variables determine the overall rate of forgetting. The coefficients of -0.2 and -0.3 respectively produce a forgetting function that is negative exponential with forgetting rates lessening over time as Fig. 5-2 demonstrates.

Exponents -0.2/-0.3

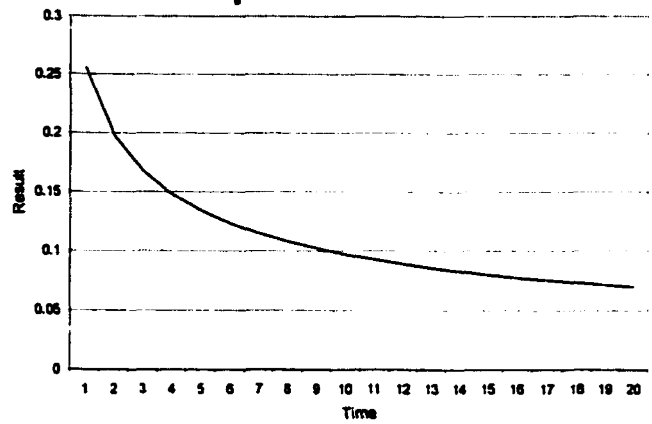


Fig. 5-2. Exponents of Forgetting -0.2/-0.3

A series of coefficients was then reviewed to determine which would produce a significant degree of forgetting. The following sets of coefficients were examined: 0.2/0.2, -0.1/0.2, 0.1/0.2, -0.1/-0.2, -0.2/-0.3, and -0.1/0.1. Only two of the combinations inspected showed a positive exponential curve, -0.1/0.2 and 0.1/0.2 respectively. The one producing the higher degree of forgetting was 0.1/0.2 (Figs. 5.3 and 5.4).

Exponents -.1/.2

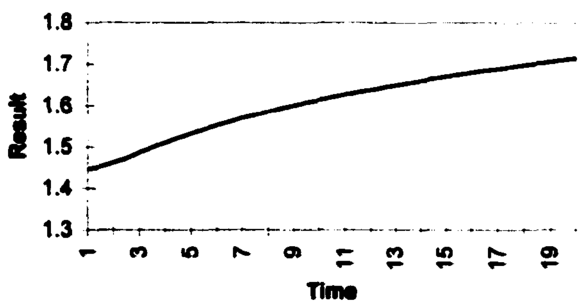


Fig. 5-3. Forgetting With Exponents -.1/.2

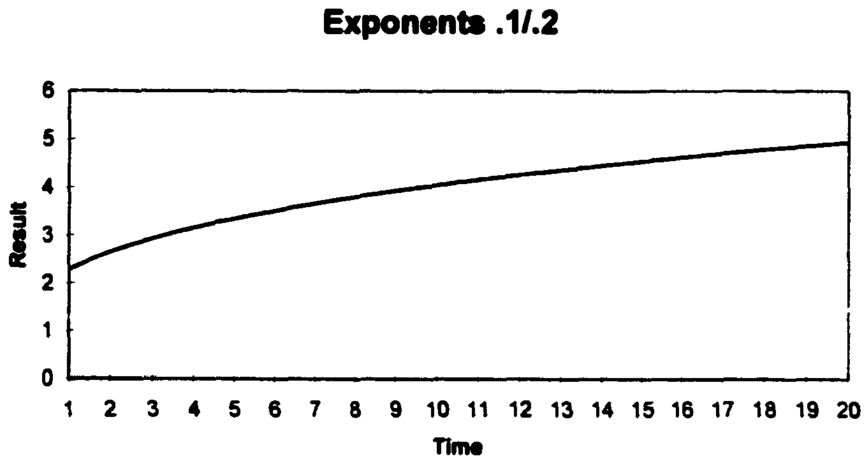


Fig. 5-4. Result With Exponents .1/.2

Thus, from Fig. 5-2 it is apparent that within 200 time units of the initiation of forgetting, the effects of forgetting during the learning process have been minimized. This assumes that the maximum amount of forgetting occurs initially and decreases over time.

The simulation and calculation of the resultant function was then run to obtain and analyze the results. The initial forgetting coefficients used were -0.20 and -0.30 in Eq. 5-3. The initial run of the simulation, shown in Fig. 5-5, collected data every 30 time increments and resulted in a very smooth curve with virtually no visible impact due to forgetting.

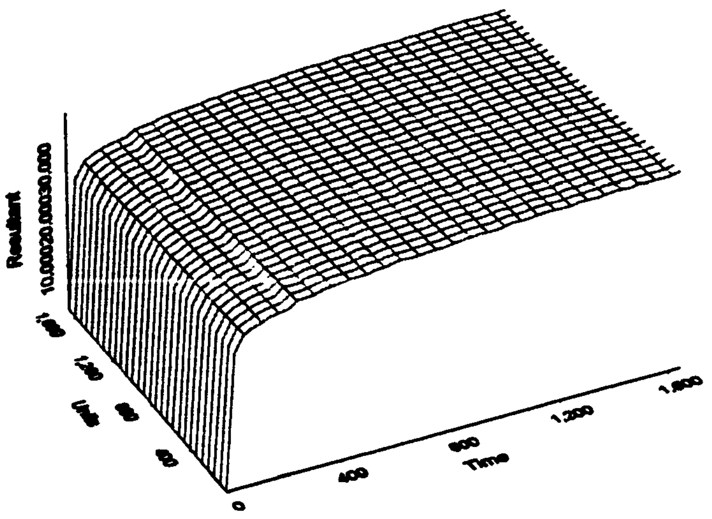


Fig. 5-5. Resultant Curve - Full Scale with Small Time Increments

Fig. 5-6 depicts the same curve with an expanded resultant scale. Still very little effect due to forgetting is visible.

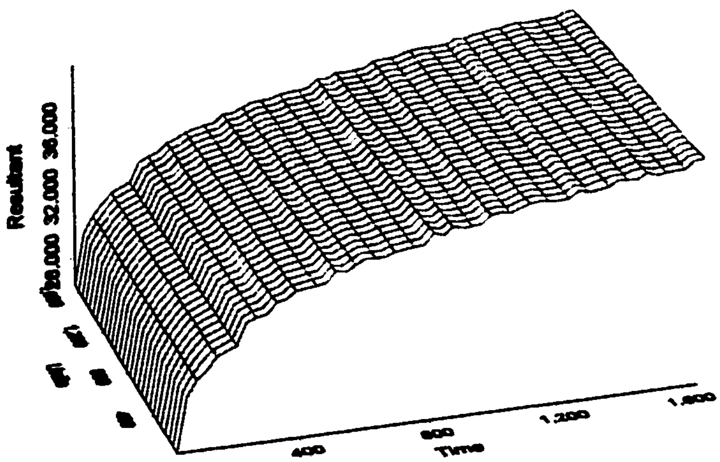


Fig. 5-6. Resultant Curve with Reduced Scale

Fig. 5-7 was then created using the same parameters except the time increment between samples was extended to 50. Still, the resultant curve demonstrated little effect due to forgetting.

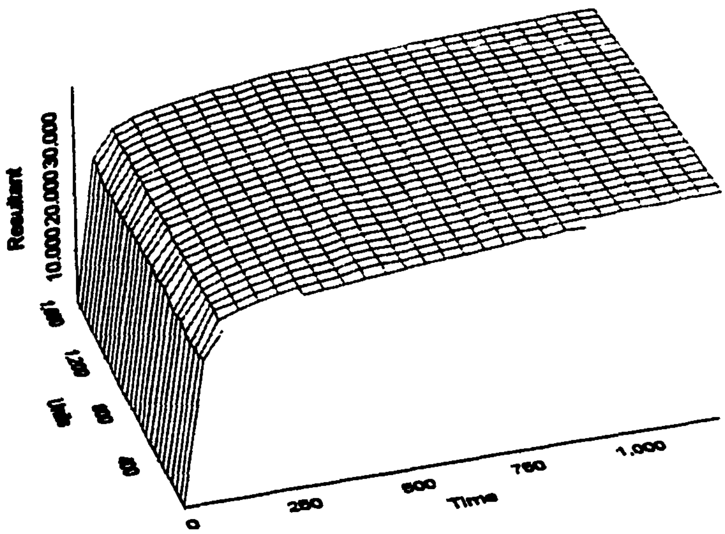


Fig. 5-7. Resultant Curve with Larger Time Increment

It was apparent that the plotted data would have to be taken using larger time increments in order to discern differences and that the coefficients affecting forgetting must be changed to determine visible forgetting effects. Since the coefficients of 0.1/0.2 produced the higher rate and degree of forgetting, they were substituted into Eq. 5-3 and the simulation produced the graph in Fig. 5-8.

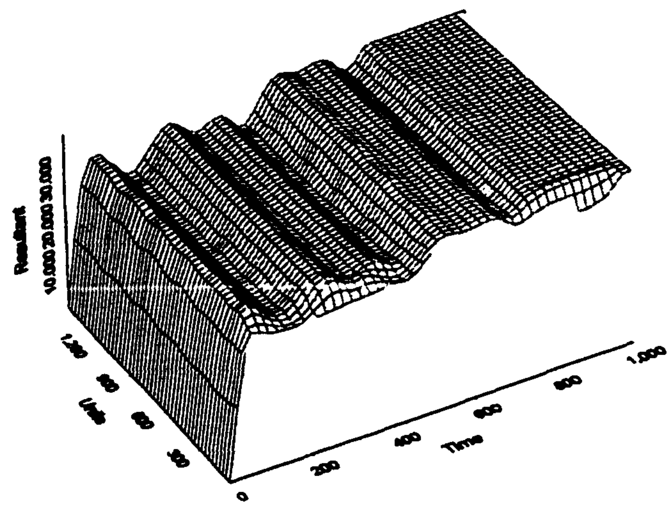


Fig. 5-8. Resultant Curve with Large Degree of Forgetting

This representation showed the effects of forgetting when forgetting was significant and continued over time. Minimal smoothing was applied in this view. Increasing the smoothing interval results in Fig. 5-9.

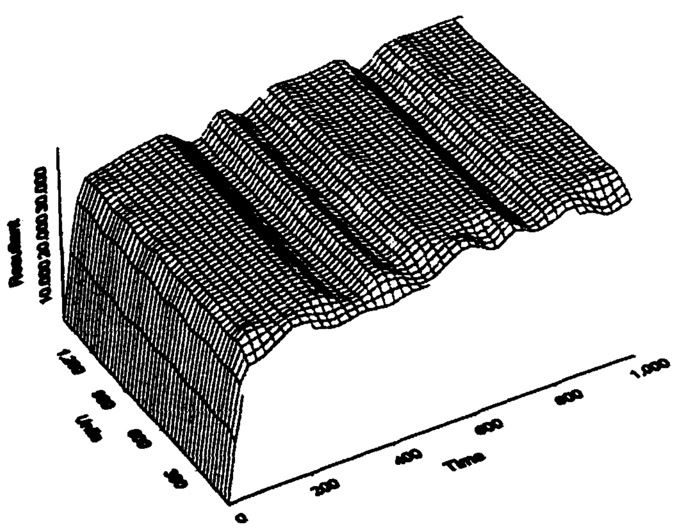


Fig. 5-9. Smoothing Applied to High Forgetting

Increasing the smoothing weights from .2, .3, and .5 respectively, to .4, .3, and .3, provides more weight on older values and produced Fig. 5-10.

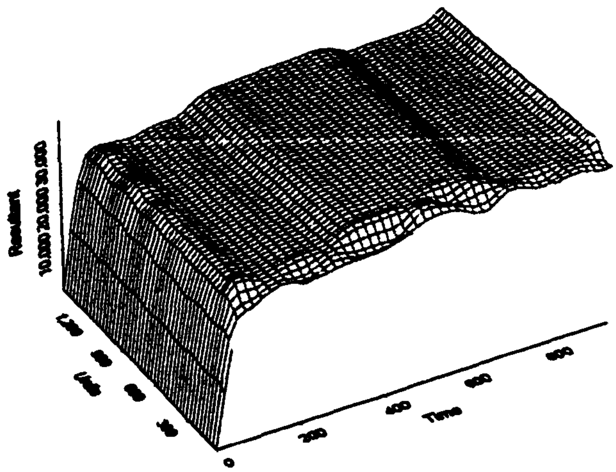


Fig. 5-10. Long Smoothing Interval with Higher Weights (.4, .3, .3) on Older Data

Thus, to view various graphical representations of a varying number and lengths of periods of forgetting, the smoothing parameters placing the higher emphasis on recent times and producing minimal smoothing were chosen. The objective was to view varying numbers and lengths of forgetting periods by making runs using different degrees of service levels with the same average arrival rates. The parameters used are in Table 5-2.

Number of Replications	5
Number of Entities	800
Warm-up Period	400
Maximum Arrival Input	6
Minimum Arrival Input	4
Moving Average Weighting	.2,.3,.5(Most Recent)
Maximum Time in Units	4000
Maximum Units	1500
Increment Between Samples	50

Table 5-2. Parameters Used in Generating Comparative Forgetting Graphs

The service level was varied as shown in Table 5-3 to produce comparative runs of various times and lengths of periods of forgetting.

Figure Number	4-12	4-13	4-14	4-15	4-16
Service Input	3	3.5	4	4.5	4.8
Avg. Queue Length	2.554	2.978	4.320	9.028	13.404
Avg. Time Between Qs	13.83	13.973	12.11	10.416	12.757
Avg. Queue Time	6.700	10.411	12.880	75.628	145.600

Table 5-3. Queue Size as a Function of Service Input

As can be seen in Fig. 5-11, using a mean service input in excess of 4.8 would lead to an infinite queue size implying continuous forgetting.

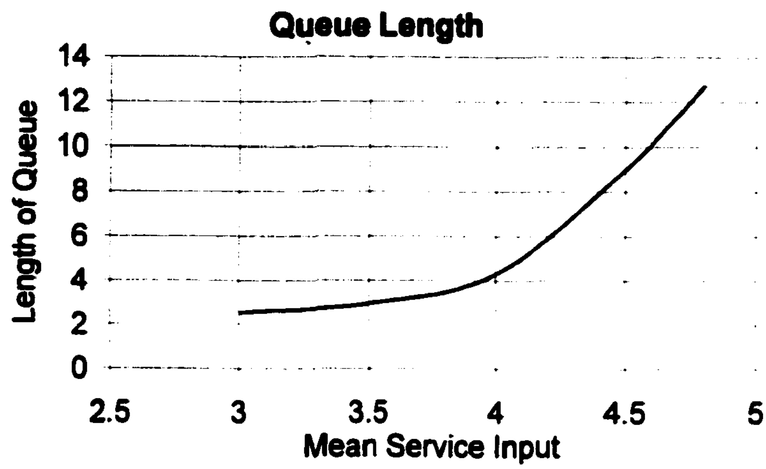


Fig. 5-11. Queue Size as a Function of Service Input

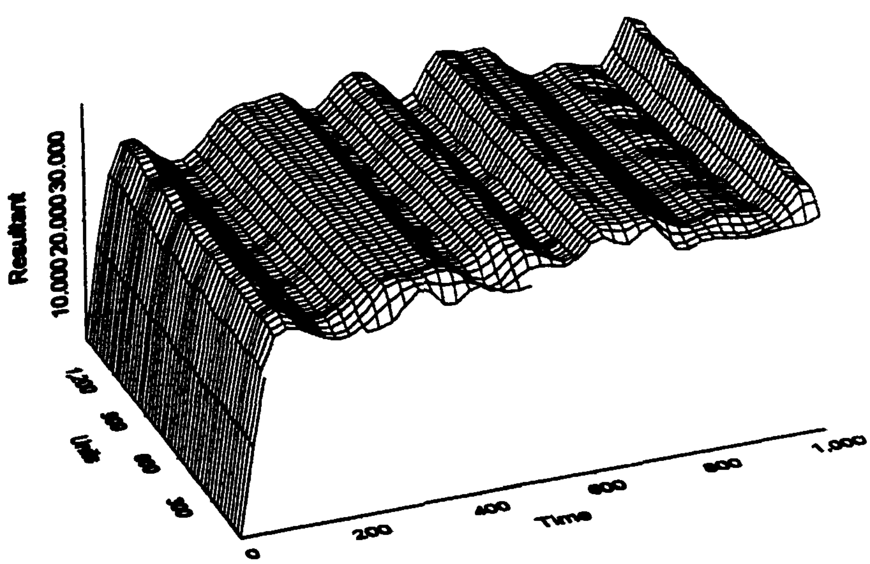


Fig. 5-12. Resultant Curve With Average Service Input of 3

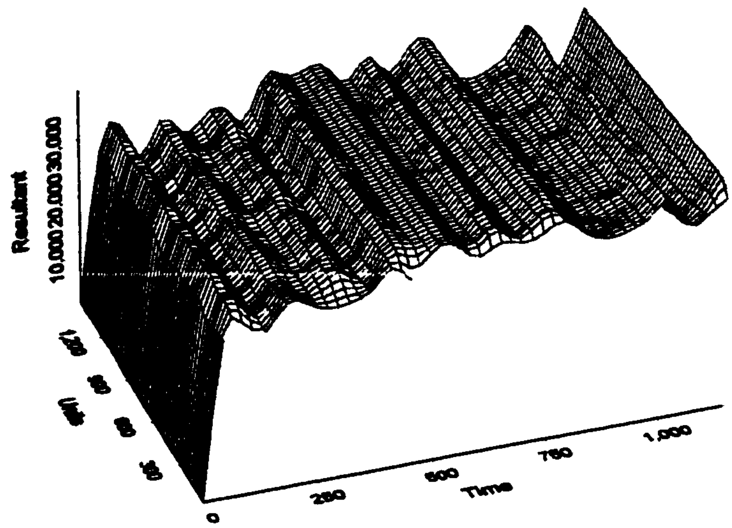


Fig. 5-13. Resultant Curve With Average Service Input of 3.5

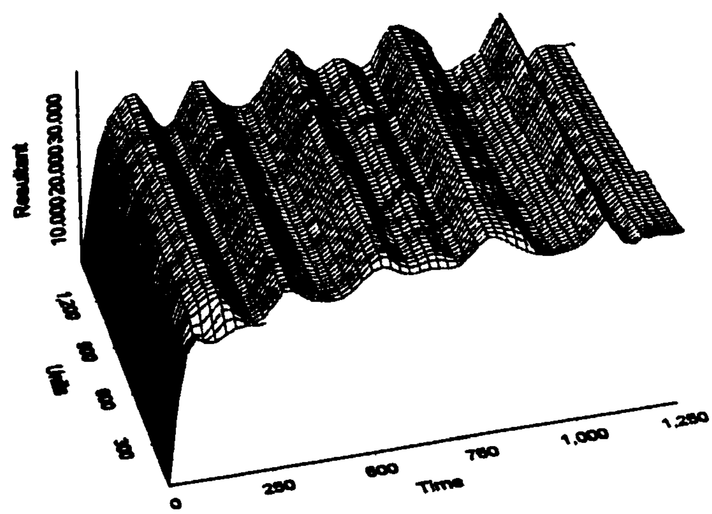


Fig. 5-14. Resultant Curve With Average Service Input of 4.0

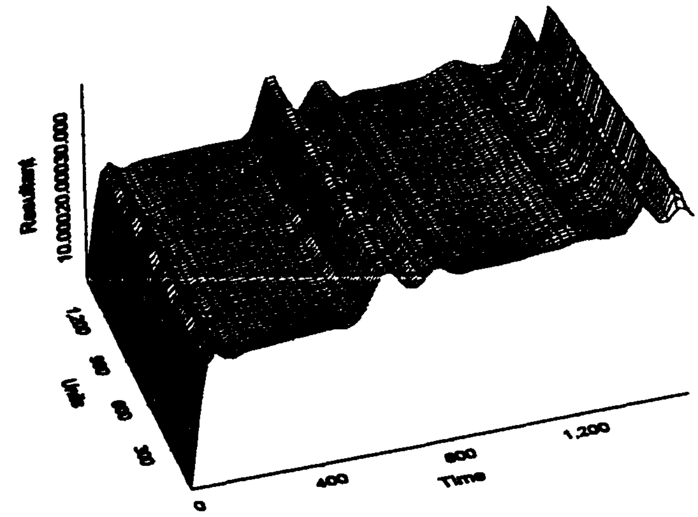


Fig. 5-15. Resultant Curve With Average Service Input of 4.5

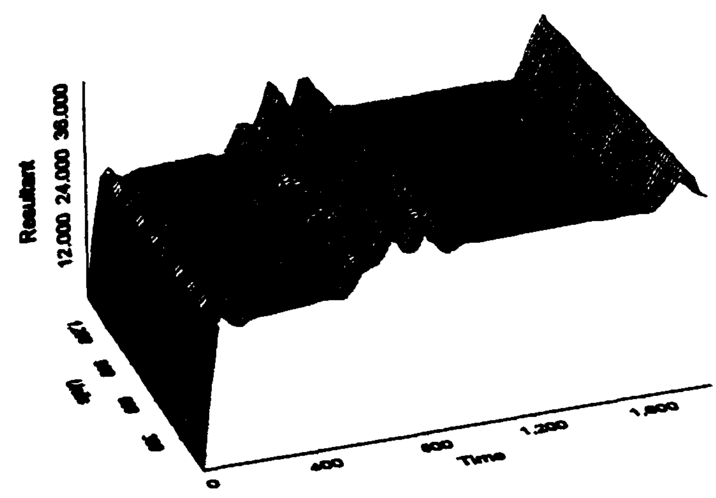


Fig. 5-16. Resultant Curve With Average Service Input of 4.8

As the service input became larger, the service rate decreased and the probability of a queue as well as the length of queues increased showing nearly continuous forgetting as in Fig. 5-16. Even though forgetting was occurring more and more during the process of learning, it only retarded the overall learning rate with learning continuing upward.

These runs were made with a program that did not reset time and units to zero each time forgetting was initiated. The result, since the exponents of forgetting rapidly drove it to the asymptote, provided maximum forgetting that continued at the same amount per unit time whenever forgetting was initiated. The computer program was modified to review the effect of resetting the time and units to zero each time forgetting ensued in order to determine if a ramping up of forgetting was visible. The initial run, shown in Fig. 5-17, was made with all of the above parameters remaining the same and a service input of 4.0.

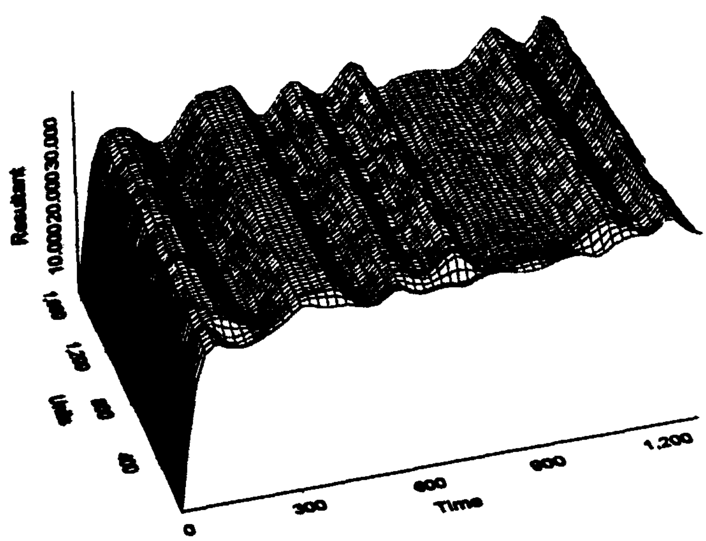


Fig. 5-17. Resultant Curve with Forgetting Reset to Zero Each Initiation

A smoothing effect is the result of setting "t" to zero each time forgetting begins. This is because the exponents used to compute forgetting in Eq. 5-3 provide a positive exponential curve, thus initially, the degree of forgetting is not as severe as when forgetting reaches a near asymptote. The same held true when increasing the service input to 4.8 in order to increase the length of forgetting periods. This is shown in comparing Fig. 5-18 below with Fig. 5-16.

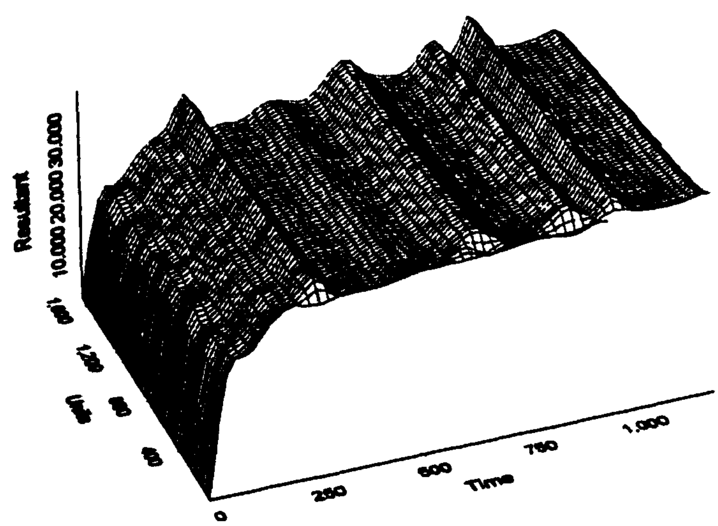


Fig. 5-18. Resultant Curve with Forgetting Reset Using A Service Input of 4.8.

Changing the exponents for forgetting to create a negative exponential curve would result in the opposite effect.

All of the above run scenarios assumed that production was continuous, with learning continuous and forgetting occurring intermittently for various lengths of time. But what would happen if a break in production would occur? This situation has been researched by many as presented and discussed earlier in this document.

Obviously, unless activities during the break were very similar and enhanced learning, forgetting would ensue during the break and the resultant ability to perform would be increasingly degraded over time. Some researchers assumed that eventually all that had been learned would be forgotten. Human experience has shown this not to be true since once one learns to ride a bicycle they do not completely forget the manual skills to ride again. Performance has, however, been degraded and must be rehearsed to achieve previous skill levels. While it has been shown that cognitive skills are lost more completely than manual skills, a total memory loss is not likely if a level of proficiency had been reached before the onset of forgetting.

To account for a break in production, the previous model was modified to account for a recurring production cycle where only forgetting occurred during the production breaks. With continuous learning, forgetting had to be accelerated functionally using a positive exponential curve to visibly see the effects of forgetting while learning continued. This would produce equally bad results when only forgetting was occurring. The model was modified so that the resultant curve was computed during the break period with only forgetting occurring. The forgetting parameters were changed to reflect a negative exponential forgetting curve as in Fig. 5-2. The initial run was made with a Cycle Time of 400 and a Production Run time of 250. The resultant graph is in Fig. 5-19 below.

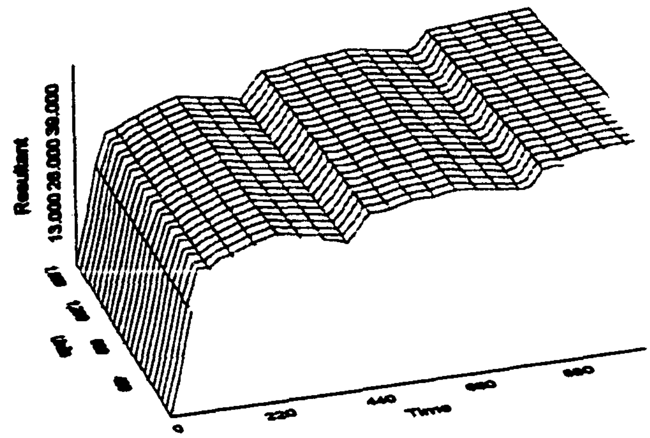


Fig. 5-19. Resultant Curve with Short Forgetting Periods During Production and a Break in Production

A run using longer periods of forgetting was made with the production parameters remaining the same. Refer to Fig. 5-20.

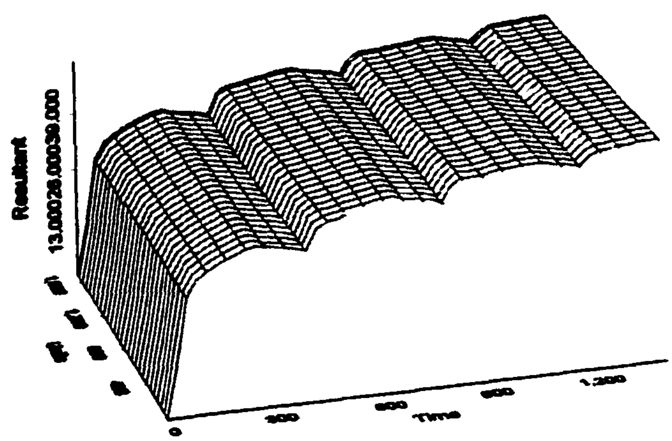


Fig. 5-20. Longer Periods of Forgetting - Production Model

The next views of interest are those associated with the same model run with a positive exponential forgetting function with both short and long forgetting periods. Refer to Figs. 5-21 and 5-22.

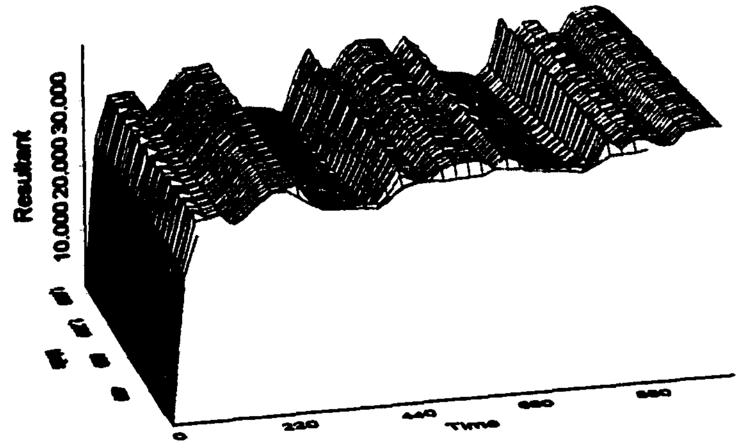


Fig. 5-21. Production Cycle with Short Forgetting Periods

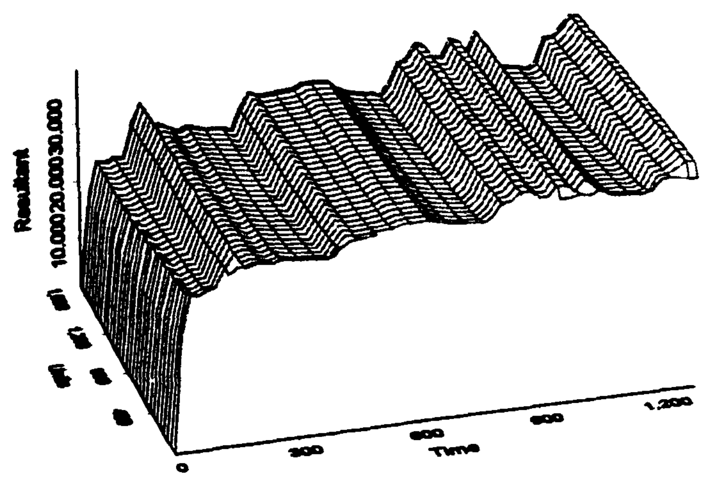


Fig. 5-22. Production Cycle with Long Forgetting Periods

The scales on the X axes between the graphs in Figs. 5-21 and 5-22 are different and must be acknowledged in order to compare the two figures.

6.0 Analysis

Analysis of the graphs and data runs shown in the previous chapter is an attempt to gain insights into the learning and forgetting scenarios in which forgetting occurs intermittently for varied lengths of time. One of the first observations was that the learning portion of the resultant equation showed a decrease in learning with an increase of units. This did not seem reasonable and the co-efficient for units (u) was changed to 0.5 from -0.5. This produced Fig. 6-1 showing a steady increase of the resultant learning as time and units increased.

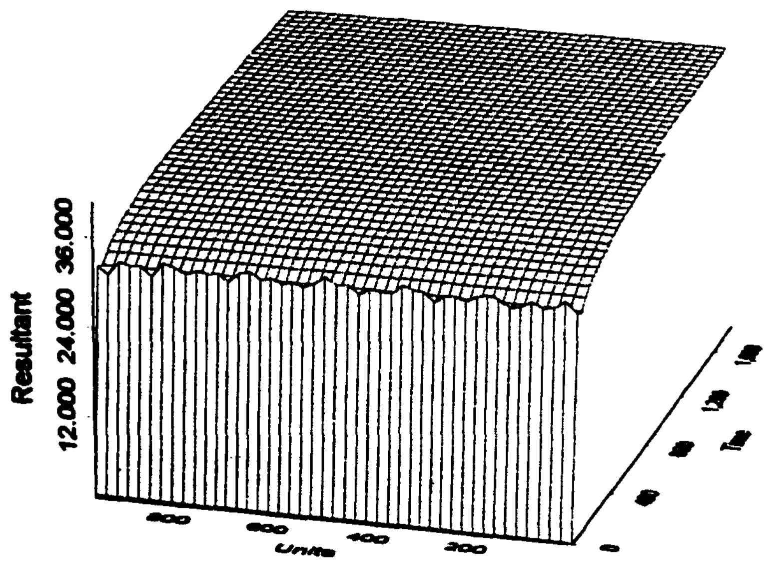


Fig. 6-1. No Forgetting with Positive Powers for t and u

One view in which forgetting is compared at both high and low levels produced Fig. 6-2. This figure shows the decrease with respect to time, the largest contributor to the resultant in Eq. 5-3, of the high forgetting curve to be significant compared to low forgetting.

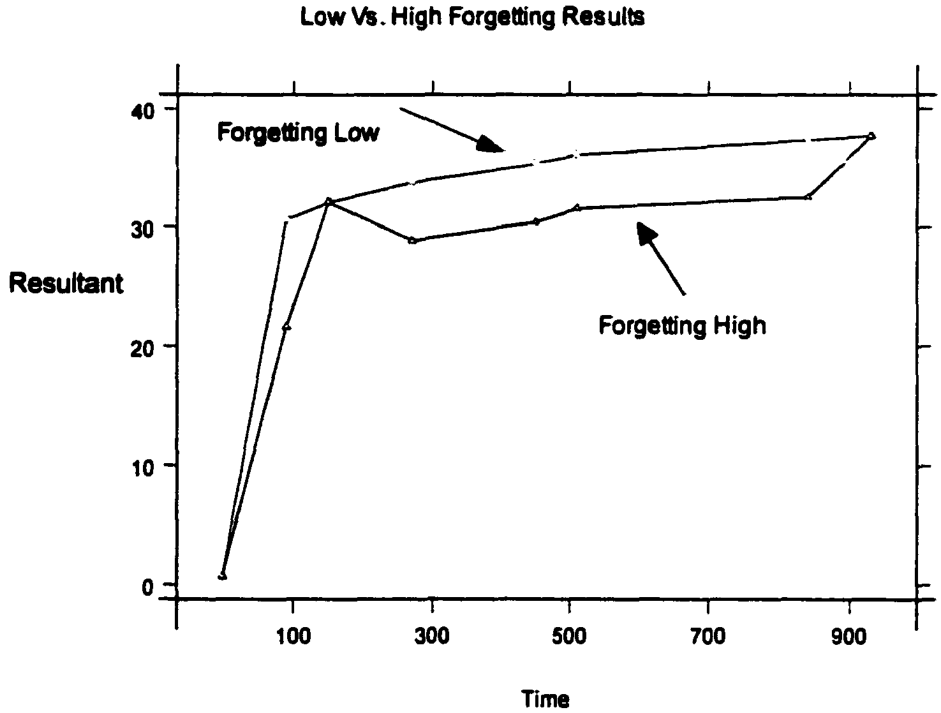


Fig. 6-2. Resultant versus Time - Various Forgetting Degrees

A comparison was made between high and low forgetting as well as comparing when there is no forgetting and when there is continuous forgetting. The low forgetting scenario was sampled 10 times with forgetting occurring a mean of 36.07 percent of available time. An alpha of .05 produced a confidence interval of 32.26 to 37.89 percent. Summary statistics of the samples are in Table 6-1.

Sample	1	2	3	4	5	6	7	8	9	10
Min.	1.689	1.300	0.641	0.675	0.761	1.346	0.742	1.755	0.397	0.972
Mean	7.825	7.483	6.696	7.625	7.119	6.447	7.339	7.538	7.408	7.270
Median	5.867	6.319	5.245	4.875	4.861	6.019	6.381	5.681	4.895	5.667
Max.	25.49	18.39	21.20	43.33	52.83	25.65	37.09	43.01	32.56	34.70
Total N	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
Std.Dev.	5.861	4.569	5.149	7.578	8.139	4.401	6.291	6.947	7.173	6.543

Table 6-1. Sample Summary Statistics

As can be seen, the mean lengths of forgetting periods are very close with an average of the means of 7.275 and a standard deviation between the means of 0.422. A review of the histograms of the data sets produced nearly normally distributed functions as in Fig. 6-3.

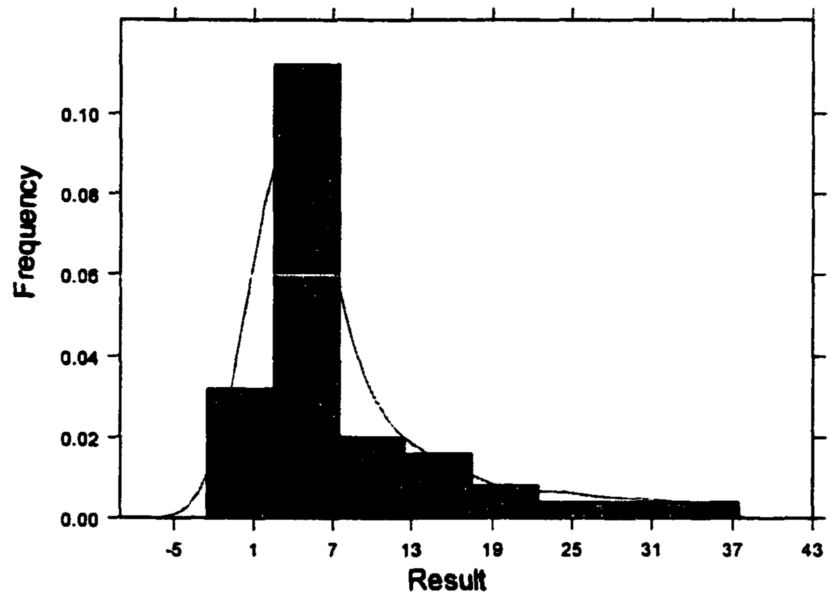


Fig. 6-3. Typical Short Queue Time Distribution

Since the histogram in Fig. 6-3 is not reflective of a normal distribution, the Kruskal-Wallis rank sum test was run to test the equivalence of the means of each of the samples of low forgetting. The samples were each of size 50. The test was an inference test in which the null hypothesis was:

$$H_0 : \mu_1 = \mu_2$$

and alternate

$$H_a : \mu_1 \neq \mu_2$$

These were tested using the T statistic:

$$T = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{[R_i - (1/2)n_i(N+1)]^2}{n_i}$$

The T value for the last two samples taken was computed as:

T = .312582

The result of Kruskal-Wallis was a chi-square distribution with 49 degrees of freedom and a p-value = 0.4731. Since T does not exceed the .95 quantile of the chi-square with df = 49 and $\alpha = .05$, the null hypothesis that the samples have the same mean cannot be rejected.

The data set was also fit to a series of functions to determine if the series could be described by a distribution. The distribution fit models in the Input Analyzer of the Arena simulation software package were utilized. The results were:

Distribution Summary

1. Distribution: Lognormal
2. Expression: LOGN(7.57, 8.59)
3. Square Error: 0.001381
4. Chi Square Test
    a. Number of intervals = 4
    b. Degrees of freedom = 1
    c. Test Statistic = 0.605
    d. Corresponding p-value = 0.457
5. Kolmogorov-Smirnov Test
    a. Test Statistic = 0.0737
    b. Corresponding p-value > 0.15
6. Data Summary
    a. Number of Data Points = 50

b. Min Data Value = 0.4

c. Max Data Value = 32.6

d. Sample Mean = 7.41

e. Sample Std Dev = 7.17

7. Histogram Summary

   a. Histogram Range = 0 to 33

   b. Number of Intervals = 7

The fit against all distributions using Kelton, Sadowski, and Sadowski appeared as:

Function	Sq Error
1. Lognormal	0.00138
2. Exponential	0.00331
3. Erlang	0.00331
4. Weibull	0.00737
5. Gamma	0.00892
6. Beta	0.0115
7. Normal	0.0957
8. Triangular	0.104
9. Uniform	0.182

Thus, this low forgetting data set produced forgetting periods most closely distributed according to the lognormal distribution with the best least squares fit. However, when all sample sets were fitted there were other distributions showing best fit. In all cases, the variances for all distributions were very large. Runs tests were made on the data and the tests showed independence (Appendix C). Additionally, autocorrelation tests showed low autocorrelation between sample values. Thus, with independent values, low autocorrelation, variable distribution fits and high variance, the data shows the characteristics desired of a consistent mean with high variance.

The high forgetting scenario was then sampled and produced forgetting an average of 79.26 percent of total available time with a .05 confidence interval of 76.90 to 81.61 percent. The same treatment was applied to this data set as the data set with low forgetting. A histogram of one sample is in Fig. 6-3 with all samples showing variation.

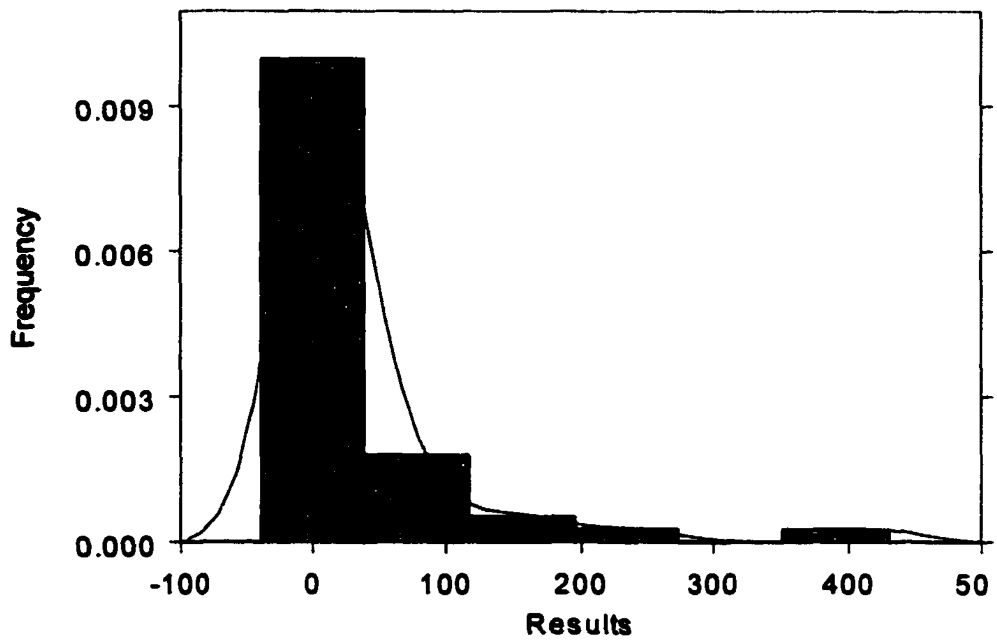


Fig. 6-4. Typical Long Queue Time Distribution

The data for high forgetting generation using queue length produced the same results as with low forgetting discussed earlier - same mean with high variability.

An analysis of the time between queues of the same sample data revealed the same conclusion (Appendix C). As a result, the sample data generated exhibited the same desired qualities of intermittence between forgetting periods with varying forgetting period lengths. By varying the input range between arrivals, various degrees of forgetting could be generated.

Once the quality of the sample data had been analyzed, the analysis of the four scenarios was pursued. The four data streams reviewed for the three forgetting scenarios studied were:

1. No forgetting
2. Low forgetting
3. High forgetting
4. Continuous forgetting.

A review of the summary statistics for the four scenarios is in Table 6-2.

*** Summary Statistics ***

	NoForget	LowForget	HighForget	ContForget
Min:	1.120000	1.120000	1.120000	1.120000
1st Qu.:	34.082500	32.052500	29.665000	28.245000
Mean:	34.756907	33.304367	31.668164	28.899804
Median:	36.440000	34.305000	32.860000	30.100000
3rd Qu.:	37.760000	37.220000	36.200000	31.187500
Max:	38.710000	38.610000	38.510000	35.250000
Total N:	1881.000000	1881.000000	1881.000000	1881.000000
Std Dev.:	6.350943	6.488334	6.567526	5.307088

Table 6-2. Summary Statistics

The means of the resultants as shown in the table above reflects a decreasing trend as forgetting continues to increase. Forgetting was computed with the time reset to zero whenever forgetting was initiated. This was for the computation of the forgetting portion of the resultant function only. The time used to compute the learning portion of the resultant was the current cumulative time. Table 6-3 reflects the results of 1122 data points observed where:

Instance	Forgetting (% of Total)	Resultant Reduction (%)
Low Forgetting	36.08	4.12
High Forgetting	79.26	8.85
Continuous Forgetting	100	16.36

Table 6-3. Forgetting Time Producing Resultant Reduction

low forgetting resulted in a 4.12% reduction in the overall resultant, high forgetting an 8.85% reduction, and a 16.36% reduction with continuous forgetting. Thus, reducing forgetting time by just over 20 percent can result in one-half the reduction in the resultant.

When forgetting is occurring during a production cycle, there are times when forgetting is occurring and when it is not. If a sample is taken at a time when forgetting is not occurring, it may be the same as that of a sample of a scenario in which no forgetting were occurring. It has been shown by other researchers that when forgetting stops and learning begins again, learning of the same information is at an increased rate. Thus, the increased rate can catch a process up to that of no forgetting in a given period of time and then the production per unit of time will be the same for both processes. When forgetting occurs intermittently, however, the cumulative number of units will be affected throughout the production period. The resultant function denotes the processes capability at any point in time and number of units produced. The forecast resultant capability must be adjusted for the periods of forgetting in which productivity continues to increase but at a reduced rate.

Consider the case in which a production cycle occurs with a period of no production. In this instance, forgetting is occurring during the learning process while in production, followed by a break in production during which a change must be made. Some researchers believe that with cognitive tasks, breaks in learning can produce a positive learning result and that during the break in production, learning may be enhanced as though production had not stopped. Other researchers such as Womer believe that performance of a similar task during the break can result in no loss of previous learning and that learning resumes at the same state when production begins again. Most believe, from their own experience, that when there is a break in learning, with or without concurrent forgetting, the result is a loss of some of what was previously learned. The rate of loss, or rate of forgetting, seems to vary depending upon the type and complexity of the task. The model was run in this scenario with forgetting occurring intermittently in a production cycle. Forgetting was computed as before with time and units reset to zero at the onset of forgetting. During the break in production, forgetting occurred with the amount learned remaining fixed in time. Thus, with the forgetting parameters set to reflect a positive exponential as before, forgetting would begin rapidly and slow to a near asymptote, but it would continue through time. The resultant curve would decrease and flatten as the forgetting near asymptote was reached. The result is not in closed form, but an indication of the amount of lost time and production.

The results of the samples taken using a production cycle showed a further decrease in the resultant as the amount of forgetting increased due to a break in learning. There were reductions due to the break in production and to intermittent forgetting. Five degrees of forgetting were considered plus that of no forgetting. The results are in table 6-4.

Instance	**Resultant Reduction (%)**
No forgetting	5.69
Low Forgetting	7.36
Medium Low Forgetting	7.89
Medium High Forgetting	8.51
High Forgetting	13.50
Continuous Forgetting	17.94

Table 6-4. Production and Forgetting Resultant Reduction

Table 6-4 reflects data collected in a production cycle operation with intermittent forgetting occurring at different levels and the affect upon overall production capability. If during a production process and while production is stopped, forgetting is mitigated by management action through the use of training or assignment of workers to similar tasks, the gain in production can be significant.

6.1 An Example

The resultant function can be used as an expression of performance. Thus, it can compute the number of units or assemblies that can be produced in a cycle where u represents the cumulative number of units produced and t represents the cumulative hours of operation. The data was output with u ranging from 10 to 1500 and t ranging from 0 to 900. When the output list is contrasted for any targeted production level, depending upon whether forgetting is occurring at the time or not, it is always better to reduce forgetting to a minimum. Thus, the importance of forgetting in the production process shows that it should not be ignored but managed for best overall results, considering the cost of training and the cost of forgetting. Knowing that forgetting cannot be stopped during the learning process or during breaks allows managers to take action to ensure the most difficult tasks, either manual or cognitive, are practiced often enough to reduce the effects upon production and the rate at which relearning may occur. Refer to the table 6.5 below for presentation of some computed points.

Forgetting Runs

Time	Units	No Forget	Low Forget	High Forget	Cont Forget
0.00	280.00	1.33	1.33	1.33	1.33
30.00	220.00	28.22	24.22	24.19	24.13
30.00	820.00	28.31	23.11	23.06	22.99
60.00	640.00	30.16	28.4	24.74	24.71
90.00	910.00	32.09	32.09	26.24	26.22
120.00	850.00	32.72	32.72	26.85	26.84
240.00	460.00	34.37	33.2	31.66	29.38
300.00	280.00	35.07	32.97	32.66	32.22
420.00	880.00	36.22	36.22	36.22	29.54
570.00	790.00	37.08	37.08	32.39	30.43
690.00	760.00	37.64	37.64	37.64	30.95
750.00	580.00	37.86	37.86	33.84	31.71
810.00	760.00	38.13	33.44	38.13	31.34
870.00	340.00	38.23	38.23	38.23	33.59
930.00	610.00	38.54	38.54	34.06	32.14
960.00	940.00	38.68	31.36	33.17	31.34

Table 6-5. Generated Production With Varying Levels of Forgetting

Thus, by taking actions to reduce forgetting a manager may achieve an increase in individual production capability ranging from approximately 13 to 23 percent. The amount of increased production will be directly related to the contributions of the variables, time and units, to the learning and forgetting processes.

7.0 Conclusions

It has become clear during the background and research for this study and through the study itself, that the acquisition and recall of knowledge must be treated as a life cycle management issue. Everyone acquires and recalls information at different yet similar rates. This knowledge, if not managed, will be forgotten to some degree at some point in time. The degree of knowledge loss depends upon the task and the level of knowledge attained before learning was discontinued. The results of the analysis lead to the conclusion that forgetting at all levels and times during the process of the knowledge life cycle has an effect upon production capability. The more knowledge that is forgotten, the greater the impact. In fact, there appears to be an exponential effect. Thus, if a person or a manager can take action, reducing the amount and length of forgetting by even one half can significantly change the results.

The bivariate model used in this study is representative of the power function developed by Wright (1936). While Carlson and Rowe (1976) demonstrated the cubic "S" curve with a good data fit, they did not dispute the worth of the Wright function in describing the learning process. The "S" curve had an initial stage in which the learning rate was slow, a growth stage in which learning was maximized, and then a mature stage in which the learning rate slowed once again resulting in a leveling of the curve. Carlson and Rowe also showed that a power function is as good for relearning as it is for initial learning.

In this study, a relearn period was not included as in Ash and Smith-Daniels (1999). This relearn period addresses the relearning of that which was forgotten during the break and it assumes that this occurs during a repetitive task cycle and not activity specific learning. The relearn period is normally short because the rate of relearning is much higher than the initial learning rate. Once the relearn period ends and the worker's proficiency level has returned to that obtained prior to the break, the learning rate returns to the original learning rate used at the time of the most recent break. From there, higher levels of knowledge may be achieved. In this model, the rate of attaining the previous level was very rapid with the powers of the affected variables producing significantly small differences and thus, it was not considered. Additionally, since the relearn rate decreases as the length of break increases (Ash and Smith-Daniels, 1999), this may be even less a factor in determining resultant learning.

A review of this work showed that the rates of learning and forgetting determine the most critical item in the computation - slope of the curve. The slope of the curves, determined by the powers of the variables, determines their effect upon the resultant curve showing overall learning. Learning is best described as a power function that will produce a rapidly increasing result in learning rate as time increases. The use of a bivariate function showed that time as well as units produced will contribute to task learning with time the more important of the two. This was additionally shown by Brazel (1972), Hirshman (1972), and, Sahal (1979). Using a forgetting curve with a positive exponential slope reveals a resulting forgetting curve describing the loss of recall as described by other researchers, initially rapid loss followed by a near leveling as time increases. It is acknowledged that an argument similar to that of Carr and Carlson and Rowe for the "S" shaped learning curve could also be waged in applying forgetting curves. Smoothing, initially theorized as necessary to reduce rapid change for ease of viewing and reflecting that actual change is not instantaneous, proved to have an insignificant effect overall even though the curves were slightly more smooth.

During multiple runs of the model to contrast input results, while the queues (forgetting periods) continue to grow in size implying an exponential lengthening of the forgetting period, the time between queues only decreased slightly. Of course, with longer queues and longer forgetting periods, there were fewer and fewer times between queues. With faster service times, there were shorter queues, but the number of queues was greater.

The multiple runs with varying degrees of forgetting showed that as intermittent forgetting became more prevalent, the decreasing effects upon the resultant learning were exponentially higher until culminating at slightly over 15 percent with continuous forgetting. Further, even a small decrease in the amount of intermittent forgetting produced a much smaller decrease in the resultant.

When production was introduced, the intermittent forgetting was accompanied by forgetting that ensues when a production break occurs. When learning was resumed following a break, it was not assumed that only learning was occurring. If intermittent forgetting was on-going at that moment, it continued. In this instance, more scenarios of intermittent forgetting were reviewed and the results showed a result similar to that as above. As forgetting approached continuous, the reduction in the resultant learning appeared to increase exponentially until reaching nearly 18 percent (continuous forgetting), dependent upon process parameters.

8.0 Future Research

In order to address continued research, a thorough review of known facts must be presented. There appears to be some confusion among the various types, modes, and methods of learning. One area is the difference between individual and organizational learning and forgetting. Individual learning applies to an individual learning everything possible regarding the task under that individual's control. Organizational learning applies to the entire process supporting and performing an individual task, that is everything in the supply chain, the process itself, and the distribution of the results of the task. Additionally, there appears to be some confusion regarding learning and process improvement. During the early stages of the initiation of a new task, a great deal of learning and task or process improvement occurs that classically is called "learning" and results in more efficient task accomplishment. This begins after the threshold level of knowledge has been attained. During the life cycle of the task, learning of the initial task does not stop, as well as being affected by learning associated with changes in the surrounding process that will occur. Many of these changes are beyond the task owner's control because they relate to changes in input materials, the method of input, the method of task accomplishment, and the methodology of task result distribution from output. While the rate of learning a particular task may seem to decrease at some point, the task does not operate in a vacuum and must interrelate with the environment which is seldom static. The process of learning in an organization involves the reaction of the organization to the extended environment in which the task exists. Thus, the individual performing each task learns and the organization as an entity supporting the individual learns. This has been proven to occur in repetitive task operations (Alchian, 1963; Shtub, 1991) as well as with non-repetitive task operations (Wright, 1936; Alchian, 1963; Globerson, 1980). After the initial stages of process implementation, formal process improvement is then encouraged to further enhance what management may term a stable task or process by introducing changes, both task oriented and process oriented. All of these effects upon the task affect the learning of the task. There has been a great deal of effort made in the acquisition, improvement, and documentation of organizational and individual initial learning. But, what occurs when forgetting is introduced into the process called "learning" which must include the ability to gain (learn) as well as recall (not forget) knowledge, both organizationally and individually? Organizations traditionally have attempted to record learning using various media and methodologies with little thought to operator retention. And when the task must be performed again, seldom is relearn time allocated for the individual. Thus, any knowledge of how to affect the life cycle of information, to extend the ability to recall learned information, both individually and organizationally, should prove beneficial.

Some known facts resulting from learning and forgetting research:

a. Learning can be described by a power function (Wright, 1936).

b. The rehearsal of an event does not affect the slope of the forgetting function, but it does affect the asymptote (Peterson and Peterson, 1959).

c. Even after performing a long distracter task, the asymptote becomes very low but never reached absolute zero probability of recall (Peterson and Peterson, 1959).

d. The rate of forgetting was less when there were fewer items learned for recall (Murdock, 1961).

e. Learning takes place over time (Brazel, 1972; Hirschman, 1964; Sahal, 1979).

f. An intuitive model described by an "S" curve provided a good fit of data from a specific manufacturing environment (Carlson and Rowe, 1976).

g. A power function provides a good model for learning <u>and</u> forgetting (Carlson and Rowe, 1976).

h. Starting anew after a break with a relearning curve provides more accurate predictions than backing up an existing learning curve (Bailey and McIntyre, 1977).

i. Threshold learning is that amount of knowledge that is required before an individual can do a job without external assistance. The time to achieve that level of knowledge should be added to the time computed by learning curves. The learning curves only contain "conditional learning time", which is the time that it takes a worker to learn after he or she barely knows how to do the work (Hancock and Bayha, 1977).

j. For a given learning rate, as the forget rate decreases, the time required to produce Q units will increase (Sule, 1978).

k. Other models have been found to apply to certain situations, yet the Wright model remains robust throughout (Globerson,1980; Yelle, 1979).

l. Breaks can cause forgetting if during the break the workers are involved in non-complimentary operations (Womer, 1984).

m. Forgetting is a complex process affected by several factors including turnover, communication, and documentation (Globerson and Levin, 1987).

n. Forgetting may be a mirror image of learning (Globerson, Levin, and Shtub, 1989).

o. The degree of forgetting is a function of the break length and the level of experience gained prior to the break (Globerson, Levin, and Shtub, 1989).

p. The most important factor affecting forgetting is the level of learning achieved before the break (Bailey, 1989).

q. The learning rate and the complexity of the task do not influence the forgetting rate (Bailey, 1989).

r. The time required to achieve the original performance level after a break is much less than the original learning rate (Bailey, 1989).

s. Forgetting is a function of the amount learned and break duration (Bailey, 1989).

t. Forgetting does not relate to forgetting rate (Bailey, 1989).

u. The relearn rate is not correlated with the learning rate but is related to skill factors and the degree of original learning (Bailey, 1989).

v. Forgetting of a procedural task was a linear function of the product of the amount learned and the log of the elapsed time (Bailey, 1989).

w. The learning curve rate was highly correlated with the time required for the first unit, for both tasks tested (Bailey, 1989).

x. For continuous control tasks, there was no detectable forgetting, but for a procedural task there was a high degree of predictability (Bailey, 1989).

y. The relearning parameter is not equal to the initial learning parameter, even where total forgetting has occurred (Hewitt *et al,* 1992).

z. To reduce forgetting, breaks should be as short as possible and if possible, delayed as long as possible in the learning process (Globerson, Levin, and Shtub, 1993).

aa. The spacing of trials has little effect on learning, but the greater spacing of practice can markedly enhance retention (Spear and Riccio, 1994).

bb. While the idea of reminiscence seems to refute the affect of time in forgetting, this only has occurred in special instances or when active processing has persisted over longer periods (Spear and Riccio, 1994).

cc. The learning of complex tasks is divided into two stages: the mental model development stage and the optimization stage. During the mental model development stage the operator learns the dynamics of the task and experiences large performance fluctuations. The optimization stage begins after the operator has a good mental model of the task, at which time the fluctuations decrease and significant performance improvement is noted. This is the real learning curve. Thus, to learn a task well, develop a good mental model of the task and understand the optimization factors of the task (Goonetilleke *et al.* 1995).

dd. Forgetting increases as the learning rate decreases and the break length increases (Dar-El *et al.* 1995).

ee. A teaching program of a mental task must concentrate on the method of decision making associated with the task and not on the technical aspects (Goonetilleke *et al.*, 1995).

ff. A break in performing a mental task may be used for improvement (Goonetillke *et al,* 1995).

gg. When breaking learning into cognitive and motor tasks, the learning rate constant under pure cognitive conditions is actually a variable that is reduced as experience is gained (Dar-El, Ayas, and Gilad, 1995).

hh. The forgetting curve slope is dependent on the length of the interruption period, the equivalent accumulated output by the point of interruption, the maximum break to which total forgetting is assumed, and the learning rate (Jaber and Bonney, 1997).

ii. Forgetting has an adverse effect on the productivity of labor intensive manufacturing and on the total labor cost due to a drop in labor productivity (Jaber and Bonney, 1997).

jj. On average, the forgetting effect of a mental task is larger than that of a mechanical task by approximately 25 percent. It seems that forgetting in a mental task is a complex process, which is affected by factors related to individual capabilities and motivation (Arzi and Shtub, 1997).

kk. In mental tasks related to emergencies, it is essential to utilize any breaks for training to increase the probability of recall (Arzi and Shtub, 1997).

ll. The forgetting intensity, rate of forgetting, is significantly affected by the rate of learning before the break. As the learning rate decreases, the forgetting rate increases. Thus, there is a correlation between an individual's capability to learn and to forget. The better learner forgets less (Arzi and Shtub, 1997).

mm. Sometimes, during breaks of mental tasks there is an unpredictable influence on forgetting. For some, a break may even improve performance. Is then the improvement due to an improved physical state of the subject or because the subject, while on break, was not on a mental break from the task and consequently forgot little or even improved during the break?

nn. True learning occurs by associating known facts in a different manner (Lowery, 1998).

oo. The effect of forgetting is unimportant when the learning rate is assumed to be 50 percent or less (Jaber and Bonney, 1999).

With all that is known, a great deal remains unknown. Specifically relating to the ability to recall once known information. During learning, organizational and individual learning rates may be similar. However, the organization relearn rate can be much different than that of the individual because of the modern management methods used to capture and document knowledge for ease of relearning. The individual does not, on average, have this ability and thus must begin relearning at a different rate and begin at a different level. During relearning and continued learning and forgetting, forgetting will continue for the individual, but not for the organization which will proceduralize and automate what has been learned. Thus, future studies must make an effort to separate organizational and individual learning, relearning, and forgetting, as well as formal process improvement. With intermittent forgetting in particular, research should concentrate on the individual's forgetting rate associated with critical knowledge and the interval length between rehearsals that would significantly reduce the effects of forgetting critical task information. Finally, this must be integrated into production methods for the minimization of resultant learning loss due to forgetting. It is known that individuals cannot behave at peak levels throughout the learning process and the remainder of the life cycle of knowledge, all of which is controlled by many behavioral factors. It is at off-peak times when forgetting will likely occur since as the learning rate decreases, the forgetting rate increases. The challenge is to anticipate forgetting and provide rehearsals at intervals necessary to maximize the probability of recall for knowledge maintenance and possible enhancement. It is assumed throughout forgetting research that relearned information equates to re-attainment of both a level of knowledge and performance. It must be recognized that to re-attain previous performance levels that cognitive thought must occur as well as motor activities essential to task accomplishment.

Immediate extensions of this paper would be to examine the effects of the amount of forgetting (low, medium, and high) on labor-intensive tasks and determine parameter sensitivity upon resultant learning.

BIBLIOGRAPHY

1. Adler, G.L. and Nanda, R. (1974). Effects of Learning on Optimal Lot Size Determination - Single Product Case, *AIIE Transactions*, 6(1), 14-20.

2. Adler, G.L. and Nanda, R. (1974). Effects of Learning on Optimal Lot Size Determination - Multiple Product Case, *AIIE Transactions*, 6(1), 21-27.

3. Arzi, Y. and Shtub, A. (1997). Learning and forgetting in mental and mechanical tasks: a comparative study, *IIE Transactions*, 29, 759-768.

4. Ash, R. and Smith-Daniels, D.E. (Winter 1999). The Effects of Learning, Forgetting, and Relearning on Decision Rule Performance in Multiproject Scheduling, *Decision Sciences*, 30(1), 47-82.

5. Badiru, A. and Pulat, P.S. (1995). *Learning Curve Analysis*, Comprehensive Project Management, Englewood Cliffs, New Jersey.

6. Badiru, A. (1994). Multifactor Learning and Forgetting Models for Productivity and Performance Analysis, *The International Journal of Human Factors in Manufacturing*, 4(1), 37 - 54.

7. Badiru, A. (1992). Computational Survey of Univariate and Multivariate Learning Curve Models, *IEEE Transactions on Engineering Management*, 39(2), 176 - 188.

8. Badiru, A. (1991). Manufacturing Cost Estimation: A Multivariate Learning Curve Approach, *Journal of Manufacturing Systems*, 10(6), 431 - 441.

9. Bailey, C. (1989). Forgetting and The Learning Curve: A Laboratory Study, *Management Science*, 35(3), 340-352.

10. Bailey, C.D. and McIntyre, E.V. (1997). The relation between fit and prediction for alternative forms of learning curves and relearning curves, *IIE Transactions*, 29, 487-495.

11. Baloff, N. (1970). Startup Management, *IEEE Transactions, Engineering Management*, EM-17, 132-141.

12. Camm, J.D., Gulledge, T.R., and Womer, N.K. (1993), Forgetting: A Production Line Representation, TIMS/ORSA Joint National Meeting, Chicago, Illinois.

13. Carlson, J.G. and Rowe, A.J. (1976). How Much Does Forgetting Cost?, *Industrial Engineering*, 8(9), 40-47.

14. Cochran, E.B. (1968). *Planning Production Costs: Using the Improvement Curve*, San Francisco, Chandler Publishing Co.

15. Conway, R.W. and Schultz, A. (1959). The Manufacturing Progress Function, *Industrial Engineering*, 10, 39-54.

16. Dar-El, E.M., Ayas, K., and Gilad, I. (1995). A dual-phased model for the individual learning process in industrial tasks, *IIE Transactions*, 27, 265-271.

17. Dar-El, E.M., Ayas, K., and Gilad, I. (1995). Predicting Performance Times for Long Cycle Time Tasks, *IIE Transactions*, 27, 272-281.

18. Diamond, M. and Hopson, J. (1998). *Magic Trees of the Mind*, Harmondsworth, Middlesex, England, Dutton.

19. Elmaghraby, S.E. (1990). Economic manufacturing quantities under conditions of learning and forgetting (EMQ/LaF), *Production Planning and Control*, 1(4), 196-208.

20. Evans, J.B., (1988). *Structures of Discrete Event Simulation: An Introduction to the Engagement Strategy*, Chichester, England, Ellis Horwood Limited.

21. Fischer, K.W. and Rose, S.P. (1998). Growth Cycles of Brain and Mind, *Educational Leadership*, November, 56-60.

22. Fleischman, E.A. and Rich, S. (1963) Role of kinesthetic and spatial visual abilities in perceptual motor learning, *Journal of Experimental Psychology*, 6, 6-11.

23. Globerson, S. and Levin, N. (1987). Incorporating forgetting into learning curves, *International Journal of Operations Research*, 7(4), 80-94.

24. Globerson, S. and Levin, N. (1995). A learning curve model for an equivalent number of units, *IIE Transactions*, 27, 716-721.

25. Globerson, S., Levin, N., and Shtub, A. (1989). The Impact of Breaks on Forgetting When Performing a Repetitive Task, *IIE Transactions*, 21(4), 376-381.

26. Globerson, S., Levin, N., and Shtub, A. (1993). Learning and Forgetting Industrial Skills: An Experimental Model, *The Journal of Human Factors in Manufacturing*, 3(3), 293-305.

27. Globerson, S. and Seidmann, A. (1988). The Effects of Imposed Learning Curves on Performance Improvements, *IIE Transactions*, 20(3), 317-323.

28. Goonetilleke, R. S., Drury, C.G., and Sharit, J. (1995). What does an operator need to learn? In *Proceedings of the Human Factors and Ergonomics Society the 39th Annual Meeting,* 1284-1288.

29. Gottfried, B.S. (1984). *Elements of Stochastic Process Simulation*, Englewood Cliffs, New Jersey, Prentice-Hall, Inc.

30. Greene, R.L. (1992). Human Memory: Paradigms and Paradoxes, Hillsdale, Lawrence Erlbaum Associates.

31. Halpern, D.F. (1984). *Thought and Knowledge - An Introduction to Critical Thinking,* London, Lawrence Erlbaum Associates.

32. Hancock, W.M. and Bayha, F.H. (1977). *The Learning Curve,* Handbook for Industrial Engineering, 4.3.1 - 4.3.13.

33. Hancock, W.M. and Foulke, J.A. (1963). Learning curve research on short cycle operations: phase 1, laboratory experiments, *MTM Association for Standards and Research Report,* no. 112.

34. Hewitt, D. Sprange, K., Yearout, R., Lianershi, D., and Sparks, C. (1992). Effects of unequal relearning rates on estimating forgetting parameters associated with performance curves, *International Journal of Industrial Ergonomics,* 10(3), 217-224.

35. Hoffman, T.R. (1968). Effect of Prior Experience on Learning Curve Parameters, *The Journal of Industrial Engineering,* 19(8), 412-413.

36. Jaber, M.Y. and Bonney, M. (1999). The economic manufacture/order quantity (EMQ/EOQ) and the learning curve: Past, present, and future, *International Journal of Production Economics.* 59, 93-102.

37. Jaber, M.Y. and Bonney, M. (1997). A Comparative Study of Learning Curves with Forgetting, *Applied Mathematical Modeling,* AMM 2288.

38. Jensen, E. (1998). How Julie's Brain Learns, *Educational Leadership*, November, 41-45.

39. Kyllonen, P.C. and Alluisi, E.A. *Learning and Forgetting Facts and Skills,* Handbook for Human Factors, 124 - 153.

40. Liebman, J.S. (1998). Teaching Operations Research: Lessons from Cognitive Psychology, *Interfaces,* 28(2), 104-110.

41. Lowery, L. (1998). How New Science Curriculums Reflect Brain Research, *Educational Leadership*, November 26-30.

42. Mayhew, D.J. (1992). *Principles and Guidelines in Software User Interface Design*, Englewood Cliffs, New Jersey. PTR Prentice Hall.

43. Mazur, J.E. and Hastie, R. (1978). Learning as Accumulation: A Reexamination of the Learning Curve, *Psychological Bulletin*, 85, 1256 - 1274.

44. Mazzola, J.B. and McCardle, K.F. (1996). A Bayesian Approach to Managing Learning-Curve Uncertainty, *Management Science*, 42(5), 680-691.

45. Mazzola, J.B. and McCardle, K.F. (1997). The Stochastic Learning Curve: Optimal Production in the Presence of Learning-Curve Uncertainty, *Operations Research*, 45(3), 440-449.

46. Payne, J.A. (1982). *Introduction to Simulation: Programming Techniques and Methods of Analysis*, New York, McGraw-Hill Book Company.

47. Pegden, C.D. (1982). *Introduction to Siman*, State College, Pennsylvania, Systems Modeling Corporation.

48. Pegels, C.C. (1969). On Startup of Learning Curves: An Expanded View, *AIIE Transactions*, 1(3), 216 - 222.

49. Pegels, C.C., (1976). On Start Up of Learning Curves—Some New Approaches, *Decision Sciences*, 7(4), 705-713.

50. Peterson, L.R. and Peterson, M.R. (1959). Short-term retention of individual verbal items, *Journal of Experimental Psychology*, 58, 193-198.

51. Pidd, M. (1989). *Computer Modeling for Discrete Simulation*, New York, John Wiley & Sons.

52. Restle, F. and Greeno, J.G. (1970). *Introduction to mathematical psychology*, Reading, Addison-Wesley.

53. Sanders, M.S. and McCormack, E.J. (1993). *Human Factors in Engineering and Design* (Seventh Edition), New York, McGraw-Hill.

54. Spears, W.D. (1985). Measurement of Learning and Transfer through Curve Fitting, *Human Factors*, 27, 251 - 266.

55. Spear, N.E. and Riccio, D.C. (1994). *Memory: Phenomena and Principles*, Boston, Allyn and Bacon.

56. Sprenger, M. (1998). Memory Lane Is a Two-Way Street, *Educational Leadership*, November, 65-67.

57. Steedman, I. (1970). Some improvement curve theory, *International Journal of Production Research*, 8(3), 189-205.

58. Sule, D.R. (1978). The Effect of Alternate Periods of Learning and Forgetting on Economic Manufacturing Quantity, *AIIE Transactions*, 10(3), 338 - 343.

59. Sylvester, R. (1995). *A celebration of neurons*, Alexandria, VA:ASCD.

60. Towill, D.R. and Kaloo, U. (1978). Productivity Drift in Extended Learning Curves, *The International Journal of Management Science*, 6(4), 295-304.

61. Weisley, R.W. (1980). *Memory, Thought, and Behavior*, New York, Oxford University Press.

62. Wolfe, P. and Brandt, R. (1998). What Now, *Educational Leadership*, November, 8-13.

63. Wolfe, P. (1998). Revisiting Effective Teaching, *Educational Leadership*, November, 61-64.

64. Womer, K. (1979). Learning Curves, Production Rate, and Program Costs, *Management Science*, 25(4), 312-319.

65. Womer, K. (1984). Estimating Learning Curves From Aggregate Monthly Data, *Management Science*, 30(8), 982-992.

66. Wright, T.P. (1936). Factors affecting the cost of airplanes, *Journal of Aeronautical Sciences*, 3(4), 122-128.

67. Yelle, L.E. (1976). Estimating Learning Curves for Potential Products, *Industrial Marketing Management*, 5, 147-154.

68. Yelle, L.E. (1979). The Learning Curve: A Historical Review and Comprehensive Survey, *Decision Sciences*, 10, 302-328.

69. Yelle, L.E. (1980). Industrial Life Cycles and Learning Curves: Interaction of Marketing and Production, *Industrial Marketing Management*, 9, 311-318.

70. Zangwell, W.I. and Kantor, P.B. (1998). Toward a Theory of Continuous Improvement and the Learning Curve, *Management Science*, 44(7), 910-920.

APPENDIXES

Appendix A:	**Random Number Generator**
Appendix B:	**Warm-Up Period Analysis**
Appendix C:	**Queue Data Analysis Presentations**
Appendix D:	**Computer Program Flow Chart and Code**

Appendix A

Random Number Generator

With Runs Tests

The following is a description of the random number generator used in this report. It is from the Borland C++ library version 4.52.

<u>Random Number Generator</u>

(Mean = 1/r)

```
float r = 1.0/mean;
return = (1.0/r * log(1.0 - UNIF(0));
```

Runs Tests

(Data on following pages)

Runs Test and Runs Above and Below the Mean Test

Runs Test

Concerned with the number of runs and the length of the runs.

Definitions:

Run: A succession of similar events preceded by and followed by a different event.

Length: Number of events that occur in each run

Runs Up: A sequence of numbers each of which is followed by a larger number.

Runs Down: Just as Runs Up only the succeeding numbers decrease in size.

Let: N be the number of numbers in a sequence.

a is the number of runs in a sequence

then

$$\mu_a = (2N - 1)/3$$

and

$$\sigma^2 = (16N - 29)/90$$

For $N > 20$,

$$a \sim N(\mu_a, \sigma_a^2)$$

then,

$$Z = (a - \mu)/\sigma_a = (a - [(2N-1)/3])\sqrt{(16N-20)/90}$$

Hypotheses:

H_0: $X_{i's}$ ~ independent

H_1: $X_{i's}$ not independent

Test Statistic

$$Z_0 = (a - [(2N-1)/3])/\sqrt{(16N-29)/90}$$

Fail to Reject H_0 when $-Z_{\alpha/2} \le Z_0 \le Z_{\alpha/2}$.

The sample data run was analyzed to produce the following results:

$a = 171$

$N = 249$

$\mu = 165.6667$

$\sigma^2 = 43.9444$

$Z_0 = 0.804538$

With $\alpha = 0.05$, fail to reject H_0 if $-Z_{\alpha/2} \le Z_0 \le Z_{\alpha/2}$

$Z_{\alpha/2} = Z_{.025} = 1.96$

Thus, since $| Z_0 | \le 1.96 \Rightarrow$ Fail to Reject H_0 and consider the generated numbers independent.

Runs Above and Below the Mean Test

This also is a test for independence of observations. It is concerned with the number of runs above and below the mean.

Let n_1 = the number of observations above the mean

n_2 = the number of observations below the mean

b = the total number of runs

N = the total number of numbers in a sequence

thus,

$$\mu_b = \frac{2n_1n_2}{N} + \frac{1}{2}$$

and

$$\sigma_b = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)}$$

If either n_1 or $n_2 > 20$, then $b \sim N(\mu_b, \sigma_b^2)$

Then $Z_0 \cong \frac{b - \mu_b}{\sigma_b} \sim N(0,1)$

Hypotheses:

H_0: X_i's ~ independent

H_1: X_i's not independent

Test Statistic:

$$Z_0 = \frac{b - (2n_1n_2/N) - \frac{1}{2}}{\left[\frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)}\right]^{\frac{1}{2}}}$$

Fail to Reject H_0 criteria is the same as in the runs test.

The sample data showed:

$n_1 = 124$

$n_2 = 125$

$b = 122$

$\mu_b = 124.998$

$\sigma_b^2 = 61.99698$

$Z_0 = -0.38075$

Fail to Reject H_0 at the same α level since $|Z_0| \leq 1.96$

Thus, two separate tests unequivocally show that the numbers in the random number generator stream are independent.

Runs Test
Above and
Below the Mean

Random Number	Inc.	Dec.	Change	Above or Below Mean	Change
0.949991					
0.239998		1	1	1	1
0.489995	1		1	1	
0.98999	1			0	1
0.599994		1	1	0	
0.839992	1		1	0	
0.949991	1			0	
0.649993		1	1	0	
0.669993	1		1	0	
0.919991	1			0	
0.779992		1	1	0	
0.359996		1		1	1
0.949991	1		1	0	1
0.549995		1	1	0	
0.509995		1		1	1
0.079999		1		1	
0.899991	1		1	0	1
0.05		1	1	1	1
0.459995	1		1	1	
0.949991	1			0	1
0.319997		1	1	1	1
0.01		1		1	
0.409996	1		1	1	
0.779992	1			0	1
0.639994		1	1	0	
0.97999	1		1	0	
0.409996		1	1	1	1
0.879991	1		1	0	1
0.739993		1	1	0	
0.759992	1		1	0	
0.929991	1			0	
0.219998		1	1	1	1
0.169998		1		1	
0.399996	1		1	1	
0.269997		1	1	1	
0.909991	1		1	0	1
0.829992		1	1	0	
0.949991	1		1	0	
0.269997		1	1	1	1
0.519995	1		1	1	
0.589994	1			0	1
0.249997		1	1	1	1
0.349997	1		1	1	
0.149999		1	1	1	

0.729993	1		1	0	1
0.349997		1	1	1	1
0.269997		1		1	
0.309997	1		1	1	
0.599994	1			0	1
0.739993	1			0	
0.949991	1			0	
0.02		1	1	1	1
0.649993	1		1	0	1
0.399996		1	1	1	1
0.309997		1		1	
0.95999	1		1	0	1
0.849992		1	1	0	
0.159998		1		1	1
0.559994	1		1	0	1
0.059999		1	1	1	1
0.119999	1		1	1	
0.259997	1			1	
0.509995	1			1	
0.129999		1	1	1	
0.839992	1		1	0	1
0.199998		1	1	1	1
0.579994	1		1	0	1
0.339997		1	1	1	1
0.509995	1		1	1	
0.349997		1	1	1	
0.699993	1		1	0	1
0.089999		1	1	1	1
0.919991	1		1	0	1
0.259997		1	1	1	1
0.429996	1		1	1	
0.789992	1			0	1
0.099999		1	1	1	1
0.489995	1		1	1	
0.649993	1			0	1
0.349997		1	1	1	1
0.159998		1		1	
0.079999		1		1	
0.97999	1		1	0	1
0.449995		1	1	1	1
0.649993	1		1	0	1
0.819992	1			0	
0.299997		1	1	1	1
0.239998		1		1	
0.389996	1		1	1	
0.139999		1	1	1	
0.03		1		1	
0.579994	1		1	0	1
0.03		1	1	1	1
0.639994	1		1	0	1
0.609994		1	1	0	
0.849992	1		1	0	
0.619994		1	1	0	

0.429996		1		1	1
0.489995	1		1	1	
0.299997		1	1	1	
0.789992	1		1	0	1
0.559994		1	1	0	
0.279997		1		1	1
0.779992	1		1	0	1
0.659993		1	1	0	
0.819992	1		1	0	
0.239998		1	1	1	1
0.479995	1		1	1	
0.869991	1			0	1
0.279997		1	1	1	1
0.659993	1		1	0	1
0.259997		1	1	1	1
0.229998		1		1	
0.429996	1		1	1	
0.949991		1	1	0	1
0.539995		1		0	
0.089999		1		1	1
0.529995	1		1	1	
0.099999		1	1	1	
0.759992	1		1	0	1
0.509995		1	1	1	1
0.619994	1		1	0	1
0.489995		1	1	1	1
0.96999	1		1	0	1
0.729993		1	1	0	
0.449995		1		1	1
0.369996		1		1	
0.749992	1		1	0	1
0.809992	1			0	
0.939991	1			0	
0.199998		1	1	1	1
0.629994	1			0	
0.699993	1			0	
0.129999		1	1	1	1
0.459995	1		1	1	
0.209998		1	1	1	
0.169998		1		1	
0.289997	1		1	1	
0.519995	1			1	
0.639994	1			0	1
0.98999	1			0	
0.829992		1	1	0	
0.719993		1		0	
0.419996		1		1	1
0.339997		1		1	
0.169998		1		1	
0.719993	1		1	0	1
0.859991	1			0	
0.929991	1			0	
0.939991	1			0	

0.259997		1	1	1	1
0.179998		1		1	
0.639994	1		1	0	1
0.689993	1			0	
0.799992	1			0	
0.269997		1	1	1	
0.179998		1		1	1
0.329997	1		1	1	
0.96999	1			0	1
0.759992		1	1	0	
0.909991	1		1	0	
0.159998		1	1	1	1
0.879991	1		1	0	1
0.859991		1	1	0	
0.129999		1		1	1
0.729993	1		1	0	1
0.279997		1	1	1	1
0.699993	1		1	0	1
0.439996		1	1	1	1
0.619994	1		1	0	1
0.139999		1	1	1	1
0.389996	1			1	
0.709993	1			0	1
0.189998		1	1	1	1
0.919991	1		1	0	1
0.909991		1	1	0	
0.169998		1		1	1
0.529995	1		1	1	
0.329997		1	1	1	
0.289997		1		1	
0.419996	1		1	1	
0.709993	1			0	1
0.449995		1	1	1	1
0.569994	1		1	0	1
0.549995		1	1	0	
0.699993	1		1	0	
0.599994		1	1	0	
0.679993	1		1	0	
0.499995		1	1	1	1
0.309997		1		1	
0.909991	1		1	0	1
0.099999		1	1	1	1
0.949991	1		1	0	1
0.03		1	1	1	1
0.849992	1		1	0	1
0.149999		1	1	1	1
0.649993	1		1	0	1
0.339997		1	1	1	1
0.529995	1		1	1	
0.229998		1	1	1	
0.499995	1		1	1	
0.519995	1			1	
0.469995		1	1	1	

0.569994	1		1	0	1
0.389996		1	1	1	1
0.499995	1		1	1	
0.709993	1			0	1
0.059999		1	1	1	1
0.439996	1		1	1	
0.859991	1			0	1
0.599994		1	1	0	
0.96999	1		1	0	
0.319997		1	1	1	1
0.819992	1		1	0	1
0.599994		1	1	0	
0.629994	1		1	0	
0.939991	1			0	
0.329997		1	1	1	1
0.869991	1		1	0	1
0.659993		1	1	0	
0.209998		1		1	1
0.579994	1		1	0	1
0.05		1	1	1	1
0.339997	1		1	1	
0.409996	1			1	
0.659993	1			0	1
0.129999		1	1	1	1
0.97999	1		1	0	1
0.529995		1	1	1	1
0.359996		1		1	
0.869991	1		1	0	1
0.699993		1	1	0	
0.929991	1		1	0	
0.129999		1	1	1	1
0.559994	1		1	0	
0.95999	1			0	
0.329997		1	1	1	1
0.579994	1		1	0	1
0.719993	1			0	
0.96999	1			0	
0.749992		1	1	0	
0.669993		1		0	
0.279997		1		1	1
0.679993	1		1	0	1
0.96999	1			0	
0		1	1	1	1
0.419996	1		1	1	
0.849992	1			0	1
0.909991	1			0	
0.679993		1	1	0	
133.348673	128	121	171	125	122

Appendix B

Warm-Up Period Analysis

Welch's Method to determine a statistical state of equilibrium for the simulation and queue was not applied in this analysis. The simulation was used to generate times in which forgetting would be assumed to be occurring and unless the warm-up period is extremely erratic before reaching stead state, the times generated starting from time = 0 could be used. A visual analysis was done reviewing simulations in which the queues were long and short to determine any period of early extreme point generation. The graphs of time versus queue lengths are presented with the number of entities set at 800, maximum arrival input at 6, minimum arrival input at 4, and the server input varying from 3 to 4.8. When a server input of 5 was used the length of the queue grew without bound.

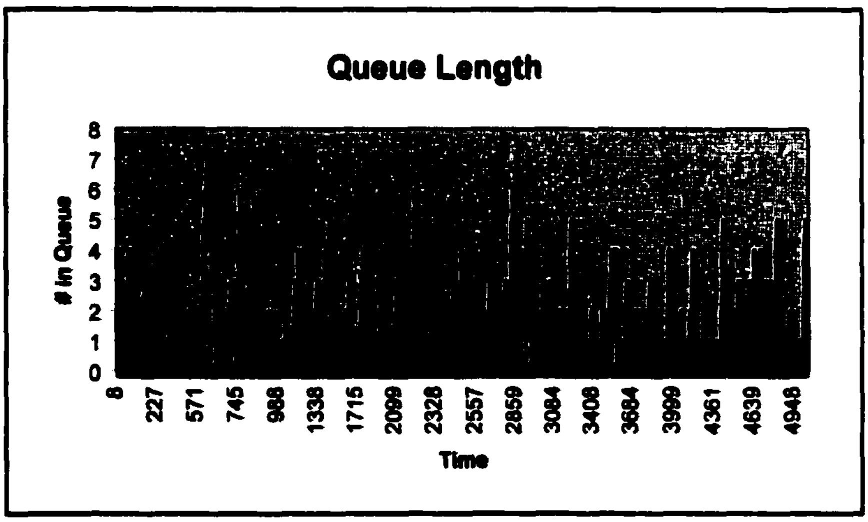


Fig B-1. Average Queue Length = 1.101,
Mean Input Rate = 3

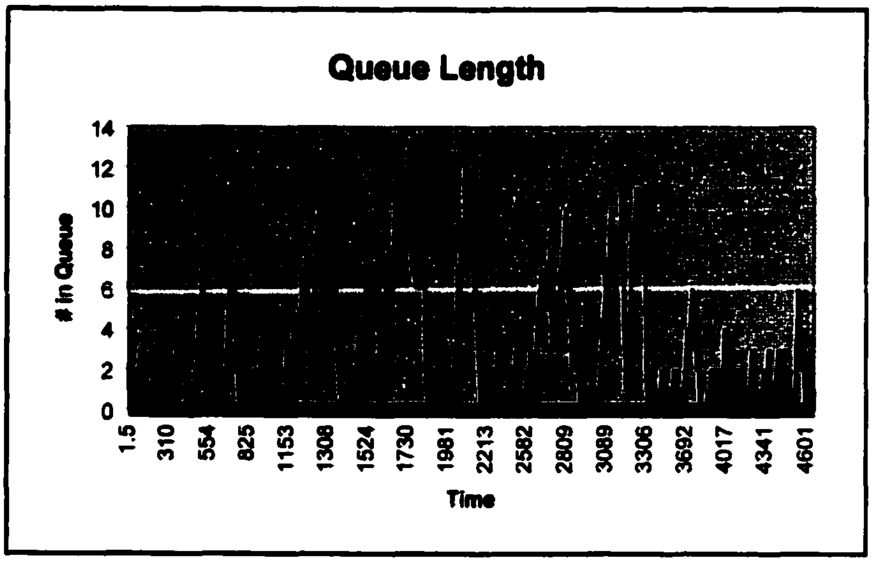


Fig. B-2. Average Queue Length = 2.746, Mean Input Rate = 3.5

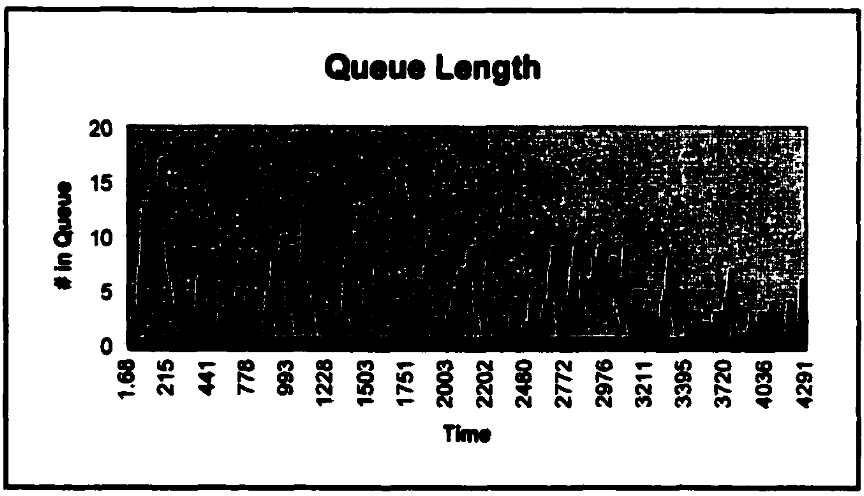


Fig B-3. Average Queue Length = 3.221, Mean Input Rate = 4

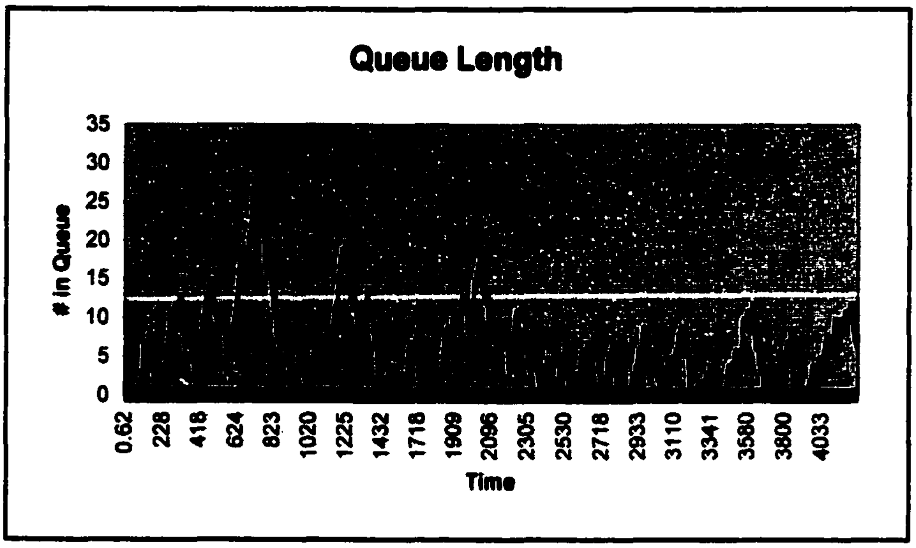


Fig. B-4. Average Queue Length = 6.064,
Mean Input Rate = 4.5

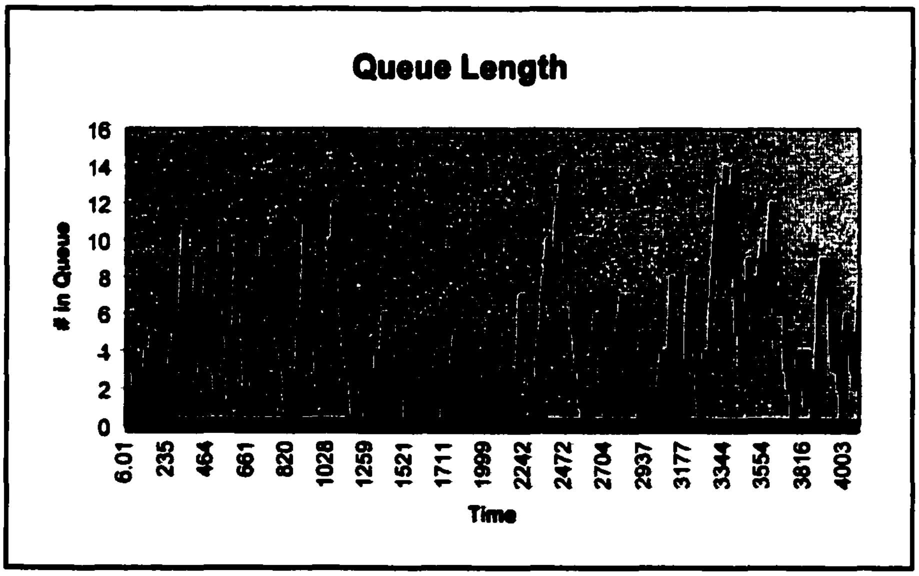


Fig. B-5. Average Queue Length = 4.102,
Mean Input Rate = 4.5

Appendix C

Queue Data Analysis Presentation

Data Review of Low Forgetting Queue Length Samples

(LFQL)

Runs and Runs Above and Below the Mean

Data	Runs	Above	Below	Mean = 7.4084
13.752533		Above		
3.21347	Minus		Below	
6.492462	Plus		Below	
6.605468	Plus		Below	
14.939209	Plus	Above		
3.380371	Minus		Below	
3.428711	Plus		Below	
12.568726	Plus	Above		
28.337829	Plus	Above		
5.253418	Minus		Below	
4.09375	Minus		Below	
10.092651	Plus	Above		
22.051758	Plus	Above		
2.693116	Minus		Below	
24.378418	Plus	Above		
6.007812	Minus		Below	
3.815552	Minus		Below	
6.932129	Plus		Below	
6.447815	Minus		Below	
9.049622	Plus	Above		
2.35968	Minus		Below	
2.687317	Plus		Below	
1.02478	Minus		Below	
5.089783	Plus		Below	
17.368591	Plus	Above		
2.005616	Minus		Below	
4.426087	Plus		Below	
2.50586	Minus		Below	
4.895569	Plus		Below	
10.280823	Plus	Above		
32.559754	Plus	Above		
4.894287	Minus		Below	
0.397217	Minus	Above		
2.599732	Plus		Below	
6.140259	Plus		Below	

7.086792	Plus		Below
0.64685	Minus		Below
1.976074	Plus		Below
5.534302	Plus		Below
9.230713	Plus	Above	
11.385987	Plus	Above	
3.574097	Minus		Below
1.475952	Minus		Below
2.633057	Plus		Below
1.986328	Minus		Below
3.13623	Plus		Below
17.505371	Plus	Above	
3.900268	Minus		Below
3.140137	Minus		Below
4.441529	Plus		Below

Runs Test

a = 30

N = 50

$\mu_a = (2N-1)/3 = 2(50-1)/3 = 32.6666$

$\sigma^2 = (16N - 29)/90 = (16(49) - 29)/90 = 8.3888$

$$Z_0 = \frac{a-\mu}{\sigma_a} = -0.9206$$

$\alpha = 0.05$

Since $Z_0 < 1.96$, Fail to Reject H_0: X_i's independent.

Runs Above and Below the Mean

$n_1 = 14$

$n_2 = 36$

$b = 20$

$$Z_0 = \frac{b - (2(n_1)(n_2)/N) - \frac{1}{2}}{\sqrt{\frac{2(n_1)(n_2)[2(n_1)(n_2) - N]}{N^2(N-1)}}}$$

$$Z_0 = \frac{20(20.16) - 1/2}{\sqrt{\frac{965664}{122500}}} = -0.235$$

Fail to Reject H_0 since $|Z_0| < Z_\alpha = 1.96$

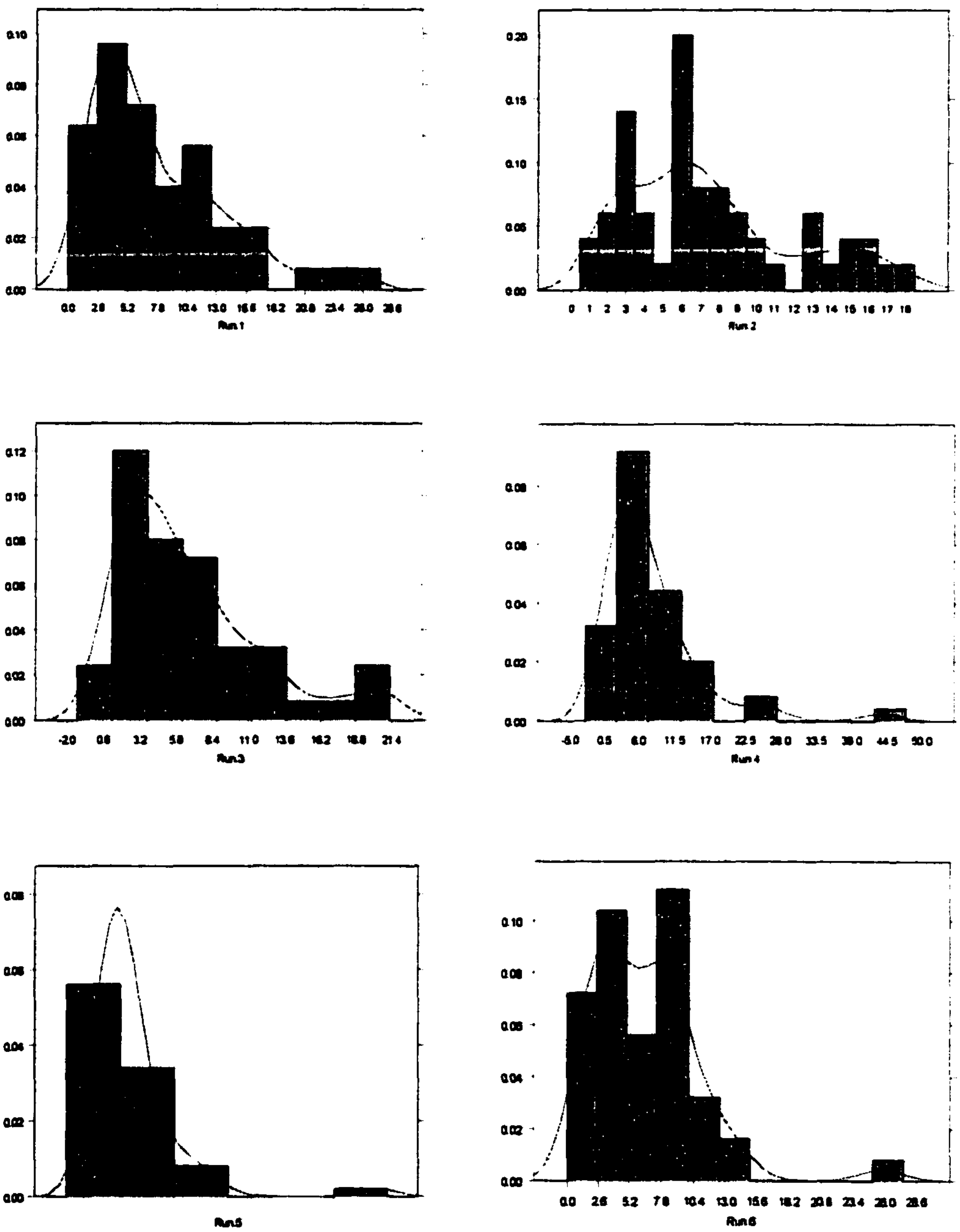
Thus, the data appears independent.

Autocorrelation matrix:

	lag	LFQL
1	0	1.0000
2	1	0.0172
3	2	-0.1434
4	3	-0.1267
5	4	0.1595
6	5	0.0377
7	6	0.1805
8	7	-0.0393
9	8	-0.1112
10	9	-0.1064
11	10	0.1271
12	11	0.0228
13	12	0.0282
14	13	-0.1918
15	14	-0.1446
16	15	-0.0172
17	16	0.4266

Run statistics are in table 6-1 of the report.

Sample Data Histograms



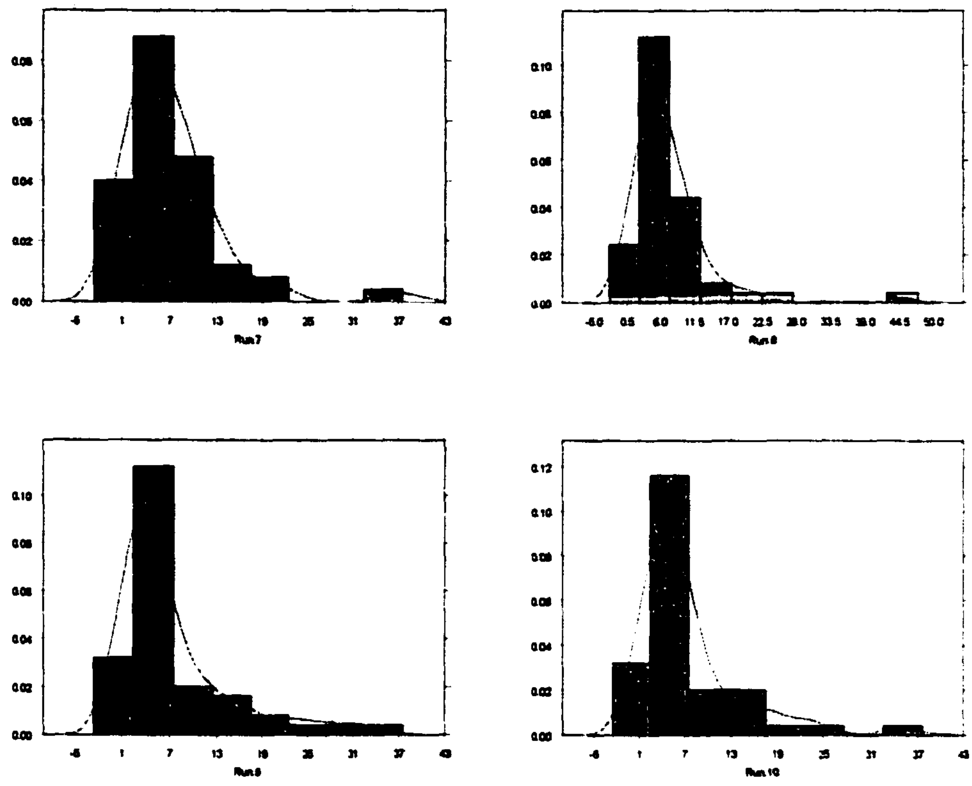


Fig. C-1. Graphs of Sample Data

Run	K-S Statistic	P value
1	0.1865	0.0002
2	0.1206	0.0668
3	0.1451	0.0102
4	0.2051	0
5	0.2968	0
6	0.1232	0.0557
7	0.1729	0.0007
8	0.225	0
9	0.2379	0
10	0.2285	0

Table C-1. One-Sample K-S Test for Composite Normality

Runs	K-S Statistic	P value
1,2	0.014	0.7166
3,4	0.06	1
5,6	0.24	0.1124
7,8	0.12	0.8693
9,10	0.12	0.8693
1,4	0.12	0.8693
1,8	0.14	0.7166
3,6	0.1	0.9667
3,9	0.1	0.9667
5,10	0.1727	0.3877

Table C-2. LFQL - Two-Sample K-S Test for Distribution Independence

Kruskal-Wallis Rank Sum Test for same data showed a p-value of 0.4731. The null hypothesis of all distributions being the same could not be rejected.

From the above data it can be concluded that the data is independent and has very low autocorrelation. Additionally, the one-sample K-S test for composite normality, sometimes referred to as the Lillefors test, shows that the data does not support an assumption of normality with rejection of the null hypothesis. The two-sample K-S test shows that the null hypothesis of equal means cannot be rejected. Since normality cannot be assumed with the data, the Kruskal-Wallis Two Sample tests showed that the null hypothesis of same distributions also could not be rejected.

A review of the distribution fit data for all ten sample runs is in table C-3.

Run 1		Run 2		Run 3	
Gamma	0.00426	Weibull	0.00621	Gamma	0.00426
Weibull	0.0046	Beta	0.00643	Erlang	0.00454
Lognormal	0.00713	Gamma	0.00648	Weibull	0.00482
Erlang	0.0104	Triangular	.00706	Lognormal	0.00837
Exponential	0.0104	Erlang	0.0105	Beta	0.0157
Beta	0.0107	Lognormal	0.0119	Exponential	0.0267
Triangular	0.0327	Exponential	0.0164	Normal	0.0282
Normal	0.0478	Normal	0.0273	Triangular	0.0497
Uniform	0.0971	Uniform	0.0411	Uniform	0.0819
Run 4		**Run 5**		**Run 6**	
Lognormal	0.00114	Beta	0.0113	Beta	0.0108
Exponential	0.00229	Lognormal	0.0267	Erlang	0.0166
Weibull	0.00318	Exponential	0.0382	Exponential	0.0166
Gamma	0.0038	Weibull	0.0543	Weibull	0.0167
Beta	0.00698	Gamma	0.0587	Gamma	0.0174
Erlang	0.0107	Erlang	0.0699	Lognormal	0.0216
Normal	0.0925	Normal	0.269	Normal	0.0556
Triangular	0.165	Triangular	0.424	Triangular	0.0776
Uniform	0.26	Uniform	0.514	Uniform	0.164
Run 7		**Run 8**		**Run 9**	
Erlang	0.00348	Lognormal	0.00186	Lognormal	0.00138
Gamma	0.00712	Exponential	0.00605	Erlang	0.00331
Weibull	0.00916	Beta	0.00664	Exponential	0.00331
Beta	0.0162	Weibull	0.0121	Weibull	0.00735
Lognormal	0.0222	Gamma	0.0131	Gamma	0.0089
Normal	0.03	Erlang	0.0207	Beta	0.0115
Exponential	0.0395	Normal	0.147	Normal	0.0957
Triangular	0.0943	Triangular	0.259	Triangular	0.104
Uniform	0.199	Uniform	0.353	Uniform	0.182
		Run 10			
		Lognormal	0.00423		
		Gamma	0.00686		
		Erlang	0.00838		
		Weibull	0.00869		
		Exponential	0.0179		
		Beta	0.0221		
		Normal	0.0682		
		Triangular	0.0994		
		Uniform	0.192		

Table C-3. Standard Error Data for Sample Fits

The sample fit data shows that a single distribution does not standout as dominant. The lognormal and the Weibull are, however, near or at the top of each of the fit listings. Thus, the data has the desired variability.

Data Review of High Forgetting Queue Length Samples

(HFQL)

Runs and Runs Above and Below the Mean

Data	Run	Above	Below	Mean =	36.97457
48.85367		Above			
29.81433	Minus		Below		
58.43802	Plus	Above			
25.84747	Minus		Below		
47.61499	Plus	Above			
47.56781	Minus	Above			
29.59149	Minus		Below		
2.054931	Minus		Below		
391.6483	Plus	Above			
4.034424	Minus		Below		
30.09436	Plus		Below		
18.1228	Minus		Below		
13.08557	Minus		Below		
18.38867	Plus		Below		
5.633179	Minus		Below		
65.9519	Plus	Above			
29.5354	Minus		Below		
17.34521	Minus		Below		
18.54236	Plus		Below		
10.31506	Minus		Below		
4.784424	Minus		Below		
56.66345	Plus	Above			
13.23779	Minus		Below		
58.89441	Plus	Above			
106.3256	Plus	Above			
28.16211	Minus		Below		
3.233154	Minus		Below		
9.400879	Plus		Below		
7.407226	Minus		Below		
12.76123	Plus		Below		
24.08301	Plus		Below		
30.83411	Plus		Below		
5.716675	Minus		Below		
14.56824	Plus		Below		

11.96314	Minus		Below
34.89014	Plus		Below
6.27295	Minus		Below
151.2922	Plus	Above	
4.270508	Minus		Below
30.97559	Plus		Below
11.36108	Plus		Below
79.36304	Plus	Above	
43.56958	Minus	Above	
15.1875	Minus		Below
15.70532	Plus		Below
7.427001	Minus		Below

Runs Test

a = 33

N = 46

$\mu_a = (2N-1)/3 = 2(46-1)/3 = 30.3333$

$\sigma^2 = (16N - 29)/90 = (16(46) - 29)/90 = 7.855$

$$Z_0 = \frac{a-\mu}{\sigma_a} = .33946$$

$\alpha = 0.05$

Since $Z_0 < 1.96$, Fail to Reject H_0: X_i's independent.

Runs Above and Below the Mean

$n_1 = 13$

$n_2 = 35$

$b = 18$

$$Z_0 = \frac{b - (2(n_1)(n_2)/N) - \frac{1}{2}}{\sqrt{\frac{2(n_1)(n_2)[2(n_1)(n_2) - N]}{N^2(N-1)}}}$$

$$Z_0 = \frac{-2.2826}{\sqrt{\frac{782600}{95220}}} = -0.796$$

Fail to Reject H_0 since $|Z_0| < Z_\alpha = 1.96$

Thus, the data appears independent.

Autocorrelation matrix:

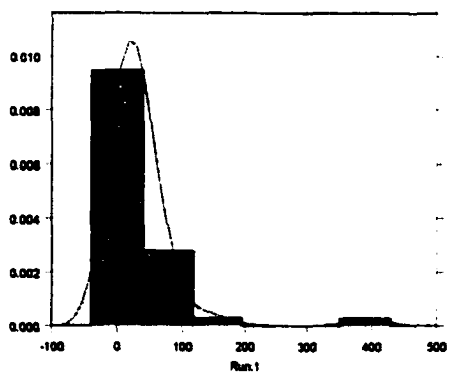
	lag	LFQL
1	0	1.0000
2	1	-0.2026
3	2	0.0250
4	3	0.0610
5	4	-0.1757
6	5	-0.0292
7	6	-0.0112
8	7	-0.1471
9	8	0.2226
10	9	-0.0966
11	10	-0.1628
12	11	0.1327
13	12	-0.1886
14	13	0.0021
15	14	0.0424
16	15	0.0093
17	16	0.0570

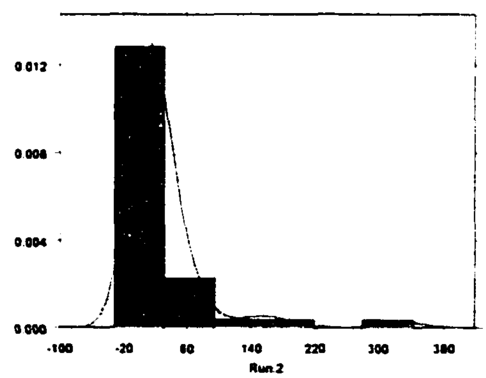
Sample	1	2	3	4	5
Min.	2.054	1.996	0.746	1.476	1.334
Mean	36.974	30.401	31.102	58.194	31.880
Median	18.466	15.899	10.002	22.184	16.562
Max.	391.648	324.983	266.103	423.514	354.395
Total N	50	50	50	50	50
Variance	3693.93	2836.32	3207.99	8184.98	2865.49
Std.Dev.	60.7	53.3	56.6	90.5	53.5

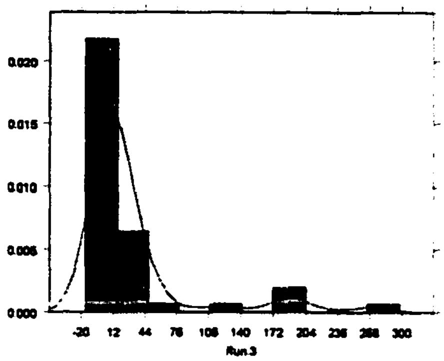
Sample	6	7	8	9	10
Min.	2.910	2.388	2.533	1.484	2.345
Mean	49.819	37.636	35.403	45.163	44.093
Median	25.915	12.498	17.522	17.056	11.978
Max.	303.913	388.025	178.999	622.663	528.129
Total N	50	50	50	50	50
Variance	5280.30	4200.03	2495.01	9385.48	8348.48
Std.Dev.	72.7	64.8	50.0	96.9	91.4

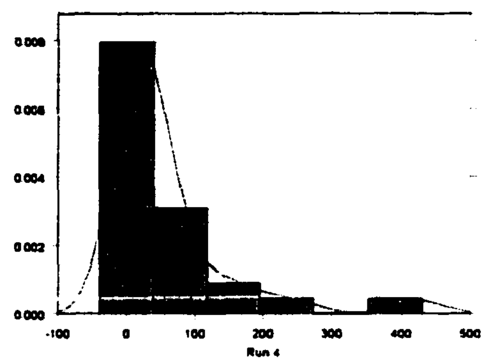
Table C - 4. HFQL Summary Statistics

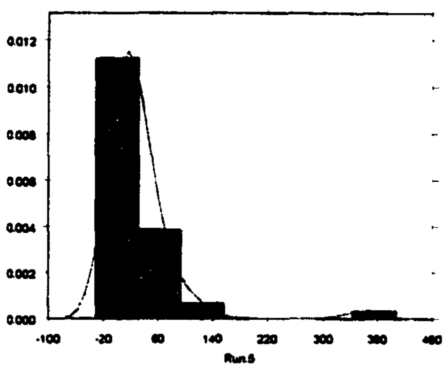
Sample Data Histograms

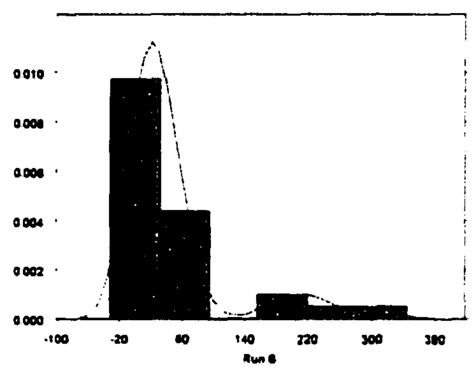


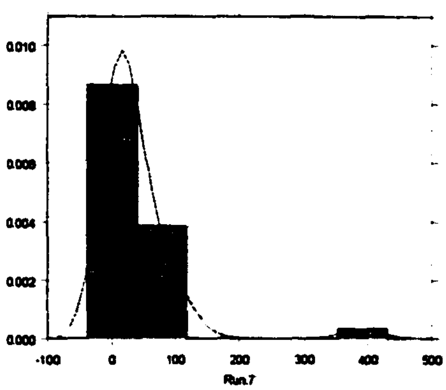


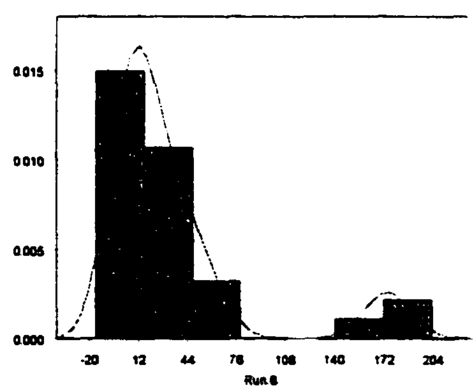

Run 3

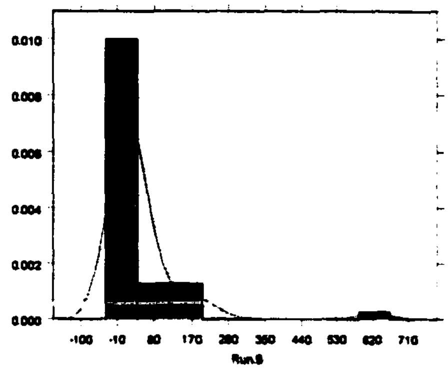

Run 4


Run 5


Run 6


Run 7


Run 8



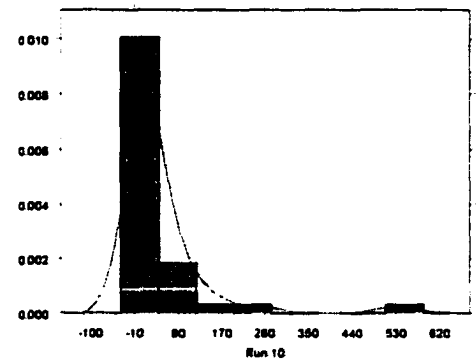


Fig. C - 2. Graphs of Sample Data

Run	K-S Statistic	P value
1	0.2617	0
2	0.3123	0
3	0.327	0
4	0.2757	0
5	0.2841	0
6	0.3139	0
7	0.2933	0
8	0.2553	0
9	0.326	0
10	0.3449	0

Table C-5. One Sample K-S Test for Composite Normality

Runs	K-S Statistic	P value
1,2	0.2183	0.1689
5,6	0.2158	0.2655
7,8	0.1667	0.6771
9,10	0.1514	0.6028
1,5	0.1591	0.5115
2,6	0.2752	0.0806
2,7	0.2	0.2974
3,9	0.24	0.1124
2,9	0.14	0.7166
2,10	0.1686	0.4732
3,10	0.1448	0.6601

Table C - 6. Two Sample K-S Test for Independent Distributions

Kruskal-Wallis Rank Sum test for the same data showed a p-value of 0.4731. The null hypothesis of all distributions being the same could not be rejected.

From the above data it can be concluded that the data is independent and has very low autocorrelation. Additionally, the one-sample K-S test for composite normality resulted with rejection of the null hypothesis. The two-sample K-S test shows that the null hypothesis of equal means cannot be rejected. Since normality cannot be assumed with the data, the Kruskal-Wallis Two Sample tests showed that the null hypothesis of same distributions also could not be rejected.

A review of the distribution fit data for all ten sample runs is in table C-7.

Run 1		Run 2		Run 3	
Lognormal	0.000696	Lognormal	0.00769	Lognormal	0.00769
Weibull	0.0027	Weibull	0.0198	Weibull	0.0198
Erlang	0.00712	Beta	0.0265	Beta	0.0265
Exponential	0.00712	Erlang	0.0466	Erlang	0.0466
Beta	0.0128	Exponential	0.0466	Exponential	0.0466
Gamma	0.0189	Gamma	0.0564	Gamma	0.0564
Normal	0.284	Normal	0.392	Normal	0.392
Triangular	0.484	Triangular	0.495	Triangular	0.495
Uniform	0.574	Uniform	0.571	Uniform	0.571
Run 4		**Run 5**		**Run 6**	
Weibull	0.00118	Lognormal	0.00132	Lognormal	0.0156
Exponential	0.00305	Weibull	0.00219	Beta	0.0262
Erlang	0.00305	Erlang	0.00492	Weibull	0.0265
Lognormal	0.00389	Exponential	0.00492	Erlang	0.0504
Gamma	0.0148	Beta	0.011	Exponential	0.0504
Beta	0.0308	Gamma	0.0142	Gamma	0.0587
Normal	0.21	Normal	0.254	Normal	0.353
Triangular	0.337	Triangular	0.484	Triangular	0.434
Uniform	0.45	Uniform	0.575	Uniform	0.526
Run 7		**Run 8**		**Run 9**	
Erlang	0.00144	Weibull	0.0141	Weibull	0.00174
Exponential	0.00144	Lognormal	0.0179	Lognormal	0.00203
Weibull	0.0033	Erlang	0.018	Erlang	0.00431
Lognormal	0.00634	Exponential	0.018	Exponential	0.00431
Gamma	.00995	Gamma	0.0224	Beta	0.0243
Beta	0.0267	Beta	0.0467	Gamma	0.029
Normal	0.209	Normal	0.225	Normal	0.307
Triangular	0.431	Triangular	0.252	Triangular	0.549
Uniform	0.537	Uniform	0.34	Uniform	0.637
		Run 10			
		Erlang	0.00259		
		Exponential	0.00259		
		Weibull	0.00398		
		Lognormal	0.00496		
		Gamma	0.0198		
		Beta	0.0795		
		Normal	0.228		
		Triangular	0.44		
		Uniform	0.543		

Table C - 7. Standard Error Data for Sample Fits

The sample fit data shows that a single distribution does not standout as dominant. The lognormal and the Weibull are, however, near or at the top of each of the fit listings. Thus, the data has the desired variability.

Data Review of Low Forgetting Time Between Queues

(LFTBQ)

Runs and Runs Above and Below the Mean

Data	Runs	Above	Below	Mean =	13.90102
7.933502			Below		
7.814972	Minus		Below		
2.519531	Minus		Below		
5.543885	Plus		Below		
12.77704	Plus		Below		
6.261078	Minus		Below		
12.22739	Plus		Below		
1.345734	Minus		Below		
4.039734	Plus		Below		
24.34125	Plus	Above			
24.55127	Plus	Above			
31.24915	Plus	Above			
2.360535	Minus		Below		
7.58374	Plus		Below		
20.38385	Plus	Above			
13.56915	Minus		Below		
9.287658	Minus		Below		
14.85638	Plus	Above			
6.240784	Minus		Below		
10.83167	Plus		Below		
7.988403	Minus		Below		
12.15009	Plus		Below		
2.937927	Minus		Below		
9.357178	Plus		Below		
15.3866	Plus	Above			
11.06421	Minus		Below		
30.20203	Plus	Above			
11.92523	Minus		Below		
22.61182	Plus	Above			
7.483521	Minus		Below		
13.79602	Plus		Below		
16.92798	Plus	Above			
13.07263	Minus		Below		

8.947754	Minus		Below
2.983764	Minus		Below
14.06775	Plus	Above	
39.95435	Plus	Above	
5.482422	Minus		Below
21.93762	Plus	Above	
15.97852	Minus	Above	
16.66541	Plus	Above	
19.35925	Plus	Above	
40.10498	Plus	Above	
16.98608	Minus	Above	
4.496826	Minus		Below
8.856933	Plus		Below
25.72253	Plus	Above	
9.366821	Minus		Below
29.61719	Plus	Above	

Runs Test

a = 32

N = 49

$\mu_a = (2N-1)/3 = 2(49-1)/3 = 32$

$\sigma^2 = (16N - 29)/90 = (16(49) - 29)/90 = 8.388$

$$Z_0 = \frac{a-\mu}{\sigma_a} = 0$$

$\alpha = 0.05$

Since $Z_0 < 1.96$, Fail to Reject H_0: X_i's independent.

Runs Above and Below the Mean

$n_1 = 19$

$n_2 = 30$

$b = 22$

$$Z_0 = \frac{b - (2(n_1)(n_2)/N) - \frac{1}{2}}{\sqrt{\frac{2(n_1)(n_2)[2(n_1)(n_2) - N]}{N^2(N-1)}}}$$

$$Z_0 = \frac{-1.765}{\sqrt{\frac{1243740}{115248}}} = -0.5372$$

Fail to Reject H_0 since $|Z_0| < Z_\alpha = 1.96$

Thus, the data appears independent.

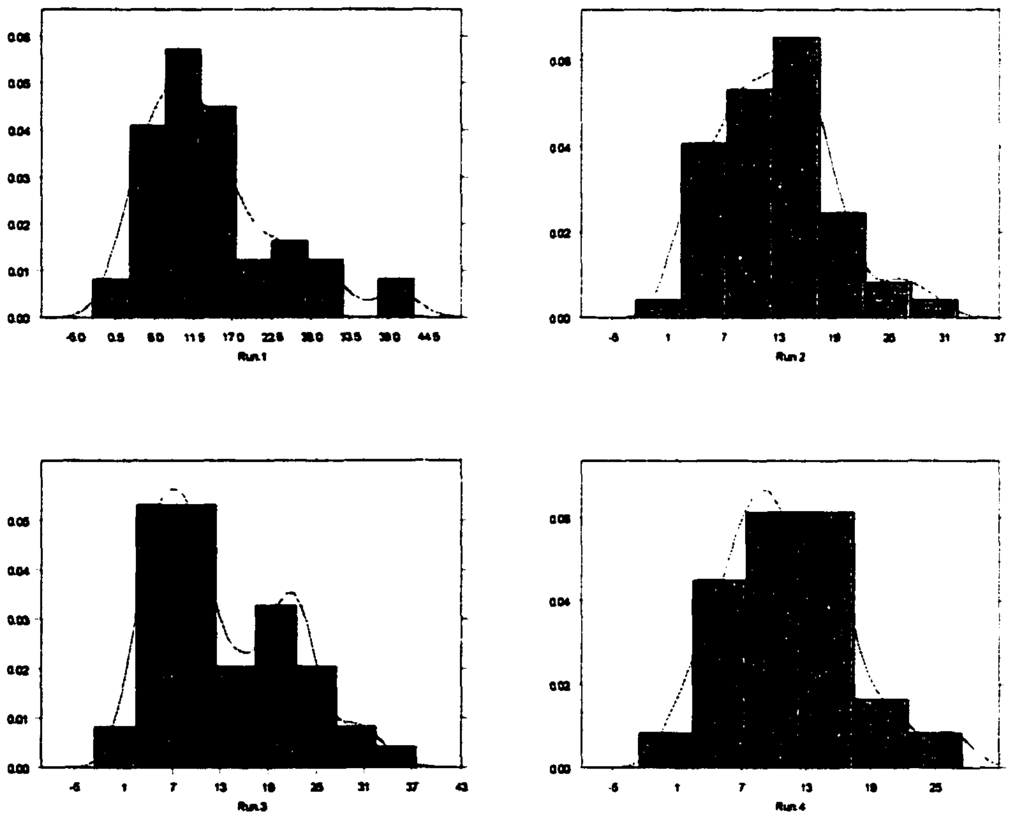
Autocorrelation matrix:

	lag	LFTBQ
1	0	1.0000
2	1	0.0507
3	2	0.0222
4	3	-0.0679
5	4	0.0598
6	5	0.0870
7	6	0.1723
8	7	-0.0386
9	8	-0.1740
10	9	-0.2127
11	10	0.2510
12	11	-0.0590
13	12	0.1451
14	13	-0.0592
15	14	-0.0238
16	15	0.1562
17	16	0.1044

Sample	1	2	3	4	5	6	7	8	9	10
Min.	1.345	2.413	1.730	0.862	3.290	2.067	2.592	2.596	1.271	2.581
Mean	13.90	12.61	13.19	11.39	14.43	13.42	13.07	12.79	10.99	13.02
Median	12.15	12.85	10.57	10.63	12.72	11.82	11.31	10.79	9.48	12.05
Max.	40.10	29.64	32.63	26.64	29.27	42.24	27.23	32.53	24.36	28.63
Variance	87.94	39.73	66.26	35.14	49.71	58.93	43.84	47.64	26.62	43.55
Std.Dev.	9.38	6.30	8.14	5.93	7.05	7.68	6.62	6.90	5.16	6.60

Table C - 8. Summary Statistics

Sample Data Histograms



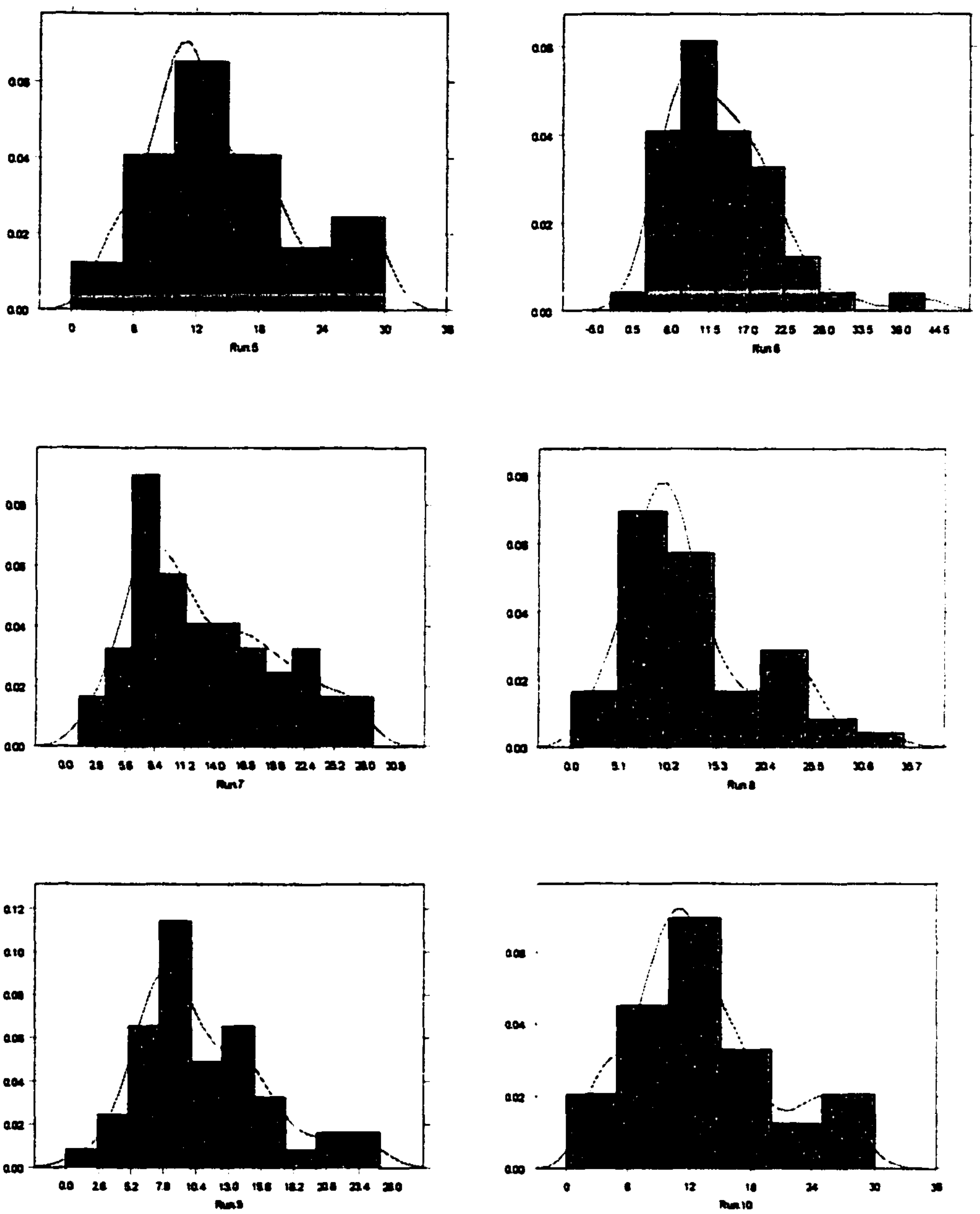


Fig. C - 3. Graphs of Sample Data

Run	K-S Statistic	P value
1	0.1262	0.0493
2	0.0596	0.5
3	0.1545	0.0051
4	0.0837	0.5
5	0.1371	0.022
6	0.0945	0.5
7	0.1176	0.0878
8	0.188	0.0002
9	0.1388	0.0192
10	0.1222	0.065

Table C - 9. One-Sample K-S Test for Composite Normality

Runs	K-S Statistic	P value
1,2	0.1429	0.7049
3,4	0.2245	0.17
5,6	0.2041	0.2611
7,8	0.1224	0.8613
9,10	0.2449	0.1059
1,5	0.2449	0.1059
1,8	0.1429	0.7049
2,6	0.102	0.9636
2,8	0.1837	0.3834
3,7	0.1837	0.3834
3,10	0.1633	0.5355
5,10	0.1224	0.8613

Table C - 10. Two-Sample K-S Test for Distribution Independence

Kruskal-Wallis Test for Same Data Showed a p-value of 0.4728 in all cases. The null hypothesis of all distributions being the same could not be rejected. From the above data it can be concluded that the data is independent and has very low autocorrelation. Additionally, the one-sample K-S test for composite normality resulted with rejection of the null hypothesis. The two-sample K-S test shows that the null hypothesis of equal means cannot be rejected. Since normality cannot be assumed with the data, the Kruskal-Wallis Two Sample tests showed that the null hypothesis of same distributions also could not be rejected.

A review of the distribution fit data for all ten sample runs is in table C -11.

Run 1		Run 2		Run 3	
Erlang	0.00337	Weibull	0.0042	Weibull	0.00298
Gamma	0.00442	Gamma	0.00535	Beta	0.0036
Weibull	0.005	Erlang	0.00598	Gamma	0.00441
Lognormal	0.0116	Beta	0.0101	Erlang	0.00446
Beta	0.0169	Normal	0.0119	Normal	0.0107
Normal	0.0287	Lognormal	0.0144	Lognormal	0.0123
Exponential	0.0304	Triangular	0.0186	Triangular	0.019
Triangular	0.0346	Exponential	0.0491	Exponential	0.0251
Uniform	0.0675	Uniform	0.073	Uniform	0.0478
Run 4		**Run 5**		**Run 6**	
Weibull	0.00229	Erlang	0.00212	Erlang	0.00162
Gamma	0.00242	Gamma	0.00221	Gamma	0.00166
Erlang	0.00247	Weibull	.00287	Weibull	0.00185
Beta	0.00416	Beta	0.00418	Beta	0.00267
Lognormal	.00733	Lognormal	0.00553	Lognormal	0.00496
Normal	0.00843	Normal	0.00953	Normal	0.00727
Triangular	0.0186	Triangular	0.0136	Triangular	0.0131
Exponential	0.0409	Exponential	0.042	Exponential	0.0384
Uniform	0.0509	Uniform	0.0484	Uniform	0.0469
Run 7		**Run 8**		**Run 9**	
Erlang	0.00111	Erlang	0.00322	Gamma	0.00238
Gamma	0.00115	Gamma	0.00322	Erlang	0.00247
Weibull	0.00148	Weibull	0.00407	Weibull	0.0034
Beta	0.00212	Beta	0.00453	Beta	0.00364
Lognormal	0.00398	Lognormal	0.00556	Lognormal	0.00437
Normal	0.00693	Normal	0.0101	Normal	0.00872
Triangular	0.016	Triangular	0.0172	Triangular	0.0145
Exponential	0.0378	Exponential	0.0398	Exponential	0.0382
Uniform	0.0446	Uniform	0.0468	Uniform	0.0446
		Run 10			
		Gamma	0.00118		
		Erlang	0.00131		
		Weibull	0.00209		
		Beta	0.00227		
		Lognormal	0.00318		
		Normal	0.00703		
		Triangular	0.0121		
		Exponential	0.0358		
		Uniform	0.0417		

Table C - 11. Standard Error Data for Sample Fits

The sample fit data shows that a single distribution does not standout as dominant. The lognormal and the Weibull are, however, near or at the top of each of the fit listings. Thus, the data has the desired variability.

Data Review of High Forgetting Time Between Queues

(HFTBQ)

Runs and Runs Above and Below the Mean

Data	Runs	Above	Below	Mean = 10.28519
5.854553			Below	
17.33453	Plus	Above		
11.07233	Minus	Above		
10.95953	Minus	Above		
11.21668	Plus	Above		
10.87268	Minus	Above		
10.03583	Minus		Below	
4.073609	Minus		Below	
9.188842	Plus		Below	
3.318054	Minus		Below	
7.088562	Plus		Below	
11.77216	Plus	Above		
8.862305	Minus		Below	
16.47357	Plus	Above		
3.811524	Minus		Below	
17.64575	Plus	Above		
10.67316	Minus	Above		
5.227051	Minus		Below	
2.229492	Minus		Below	
18.31531	Plus	Above		
11.85242	Minus	Above		
9.839844	Minus		Below	
12.31641	Plus	Above		
10.30676	Minus	Above		
7.680542	Minus		Below	
14.60815	Plus	Above		
11.64392	Minus	Above		
8.37854	Minus		Below	
9.290649	Plus		Below	
10.81079	Plus	Above		
12.75525	Plus	Above		
5.975708	Minus		Below	
9.824951	Plus		Below	

4.001343	Minus		Below
7.588379	Plus		Below
13.41773	Plus	Above	
15.08679	Plus	Above	
11.73804	Minus	Above	
24.44214	Plus	Above	
5.932129	Minus		Below
11.48608	Plus	Above	
15.34131	Plus	Above	
3.168701	Minus		Below
12.5824	Plus	Above	
6.548218	Minus		Below
7.045654	Plus		Below
13.71558	Plus	Above	

Runs Test

a = 31

N = 47

$\mu_a = (2N-1)/3 = 2(47-1)/3 = 30.6666$

$\sigma^2 = (16N - 29)/90 = (16(47) - 29)/90 = 8.0333$

$$Z_0 = \frac{a - \mu}{\sigma_a} = 0.0415$$

$\alpha = 0.05$

Since $Z_0 < 1.96$, Fail to Reject H_0: X_i's independent.

Runs Above and Below the Mean

$n_1 = 25$

$n_2 = 22$

$b = 23$

$$Z_0 = \frac{b - (2(n_1)(n_2)/N) - \frac{1}{2}}{\sqrt{\frac{2(n_1)(n_2)[2(n_1)(n_2) - N]}{N^2(N-1)}}}$$

$$Z_0 = \frac{-0.904}{\sqrt{\frac{1158300}{101614}}} = -0.268$$

Fail to Reject H_0 since $|Z_0| < Z_\alpha = 1.96$

Thus, the data appears independent.

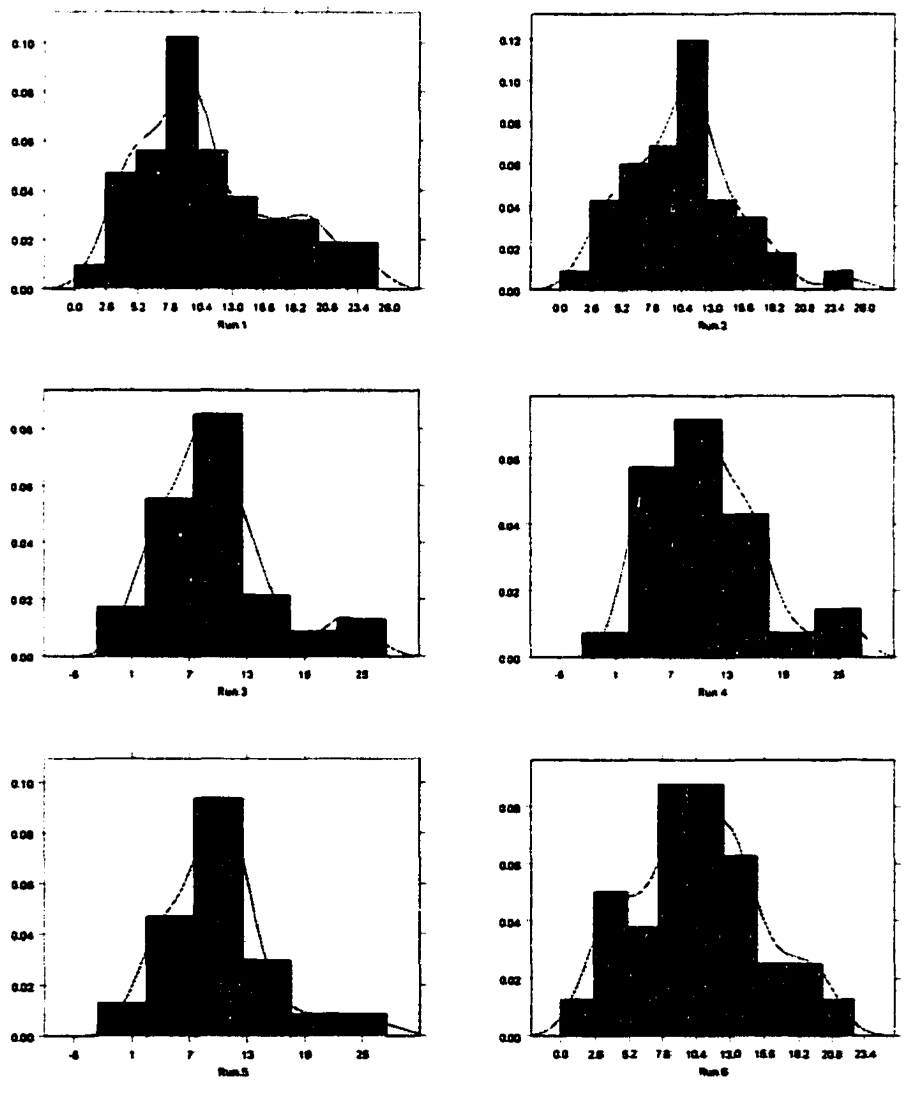
Autocorrelation matrix:

	lag	HFTBQ
1	0	1.0000
2	1	-0.1920
3	2	0.0509
4	3	0.0973
5	4	-0.1874
6	5	-0.1146
7	6	-0.0761
8	7	-0.1056
9	8	-0.0474
10	9	0.0374
11	10	0.0759
12	11	0.0749
13	12	-0.0537
14	13	-0.0420
15	14	0.0147
16	15	0.0209
17	16	0.0147

Sample	1	2	3	4	5	6	7	8	9	10
Min.	2.055	1.996	0.746	1.476	1.334	2.910	2.389	2.53	1.484	2.345
Mean	36.97	30.40	31.10	58.19	31.88	49.82	37.64	35.4	45.16	44.09
Median	18.46	15.89	10.00	22.18	16.56	25.91	12.5	17.5	17.05	11.98
Max	391.6	324.9	266.1	423.5	354.5	303.9	388.0	179.	528.1	251.7
Variance	3693.	2836.	3207.	8184.	2865.	5280.	4200.	2495	9385.	8348.
Std.Dev.	60.77	53.25	56.64	90.54	53.53	72.67	64.81	49.9	96.88	91.37

Table C - 12. Sample Summary Statistics

Sample Data Histograms



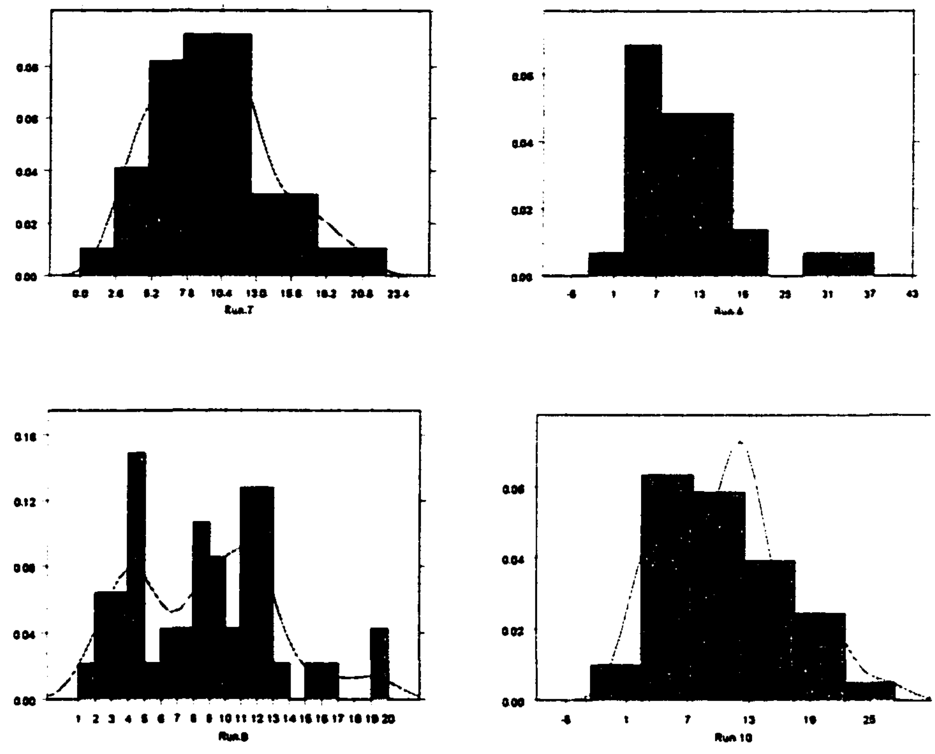


Fig. C - 4. Graphs of Sample Data

Run	K-S Statistic	P value
1	0.1647	0.005
2	0.0884	0.5
3	0.1196	0.09
4	0.0932	0.5
5	0.121	0.0824
6	0.0683	0.5
7	0.0731	0.5
8	0.2016	0.0039
9	0.2688	0.0673
10	0.2383	0.1923

Table C - 13. One-Sample K-S Test for Composite Normality

Runs	K-S Statistic	P value
1,2	0.1618	0.5297
3,4	0.1953	0.4471
5,6	0.1922	0.4209
7,8	0.1742	0.6202
9,10	0.2688	0.0673
10,11	0.2383	0.1923
1,4	0.1869	0.516
2,6	0.1509	0.7108
2,8	0.1665	0.6265
3,11	0.1526	0.668

Table C - 14. K-S Test for Independent Distributions

Kruskal-Wallis Rank Sum test for the same data showed a p-value of 0.4731. The null hypothesis of all distributions being the same could not be rejected.

From the above data it can be concluded that the data is independent and has very low autocorrelation. Additionally, the one-sample K-S test for composite normality resulted with rejection of the null hypothesis. The two-sample K-S test shows that the null hypothesis of equal means cannot be rejected. Since normality cannot be assumed with the data, the Kruskal-Wallis Two Sample tests showed that the null hypothesis of same distributions also could not be rejected.

A review of the distribution fit data for all ten sample runs is in table C-15.

Run 1		Run 2		Run 3	
Gamma	0.00215	Gamma	0.00124	Gamma	0.00122
Erlang	0.00228	Erlang	0.0014	Erlang	0.00134
Weibull	0.00329	Beta	0.00239	Beta	0.00244
Beta	0.00344	Weibull	0.00254	Weibull	0.00285
Lognormal	0.00398	Lognormal	0.00301	Lognormal	0.00311
Normal	0.00838	Normal	0.00704	Normal	0.00702
Triangular	0.0164	Triangular	0.0154	Triangular	0.0152
Exponential	0.0355	Exponential	0.0342	Exponential	0.0336
Uniform	0.042	Uniform	0.0411	Uniform	0.0408

Run 4		Run 5		Run 6	
Gamma	0.00127	Gamma	0.00215	Gamma	0.00192
Erlang	0.00138	Erlang	0.00227	Erlang	0.00205
Beta	0.00239	Beta	0.00372	Beta	0.00297
Weibull	0.00288	Weibull	0.00401	Weibull	0.00392
Lognormal	0.00331	Lognormal	0.00437	Lognormal	0.00399
Normal	0.00691	Normal	0.0075	Normal	0.00723
Triangular	0.0155	Triangular	0.0161	Triangular	0.0152
Exponential	0.0335	Exponential	0.0337	Exponential	0.0326
Uniform	0.041	Uniform	0.0417	Uniform	0.0404

Run 7		Run 8		Run 9	
Gamma	0.00194	Gamma	0.00128	Gamma	0.00177
Erlang	0.00212	Erlang	0.00146	Erlang	0.0019
Beta	0.00306	Beta	0.00232	Beta	0.00238
Lognormal	0.00395	Lognormal	0.00339	Lognormal	0.00401
Weibull	0.00433	Weibull	0.00352	Weibull	0.00423
Normal	0.00736	Normal	0.00617	Normal	0.00643
Triangular	0.0161	Triangular	0.0176	Triangular	0.0151
Exponential	0.0335	Exponential	0.0316	Exponential	0.031
Uniform	0.0417	Uniform	0.0398	Uniform	0.0395

Run 10	
Gamma	0.00174
Erlang	0.00183
Beta	0.00219
Weibull	0.00404
Lognormal	0.00418
Normal	0.00605
Triangular	0.0151
Exponential	0.0304
Uniform	0.0392

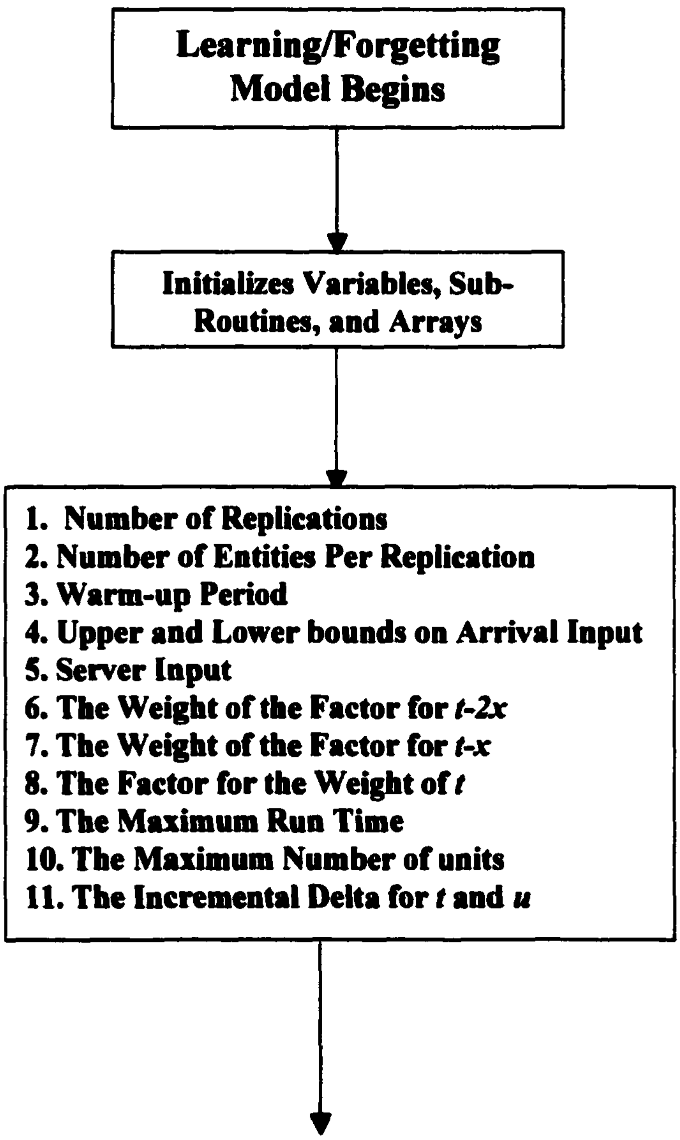
Table C - 15. Standard Error Data for Sample Fits

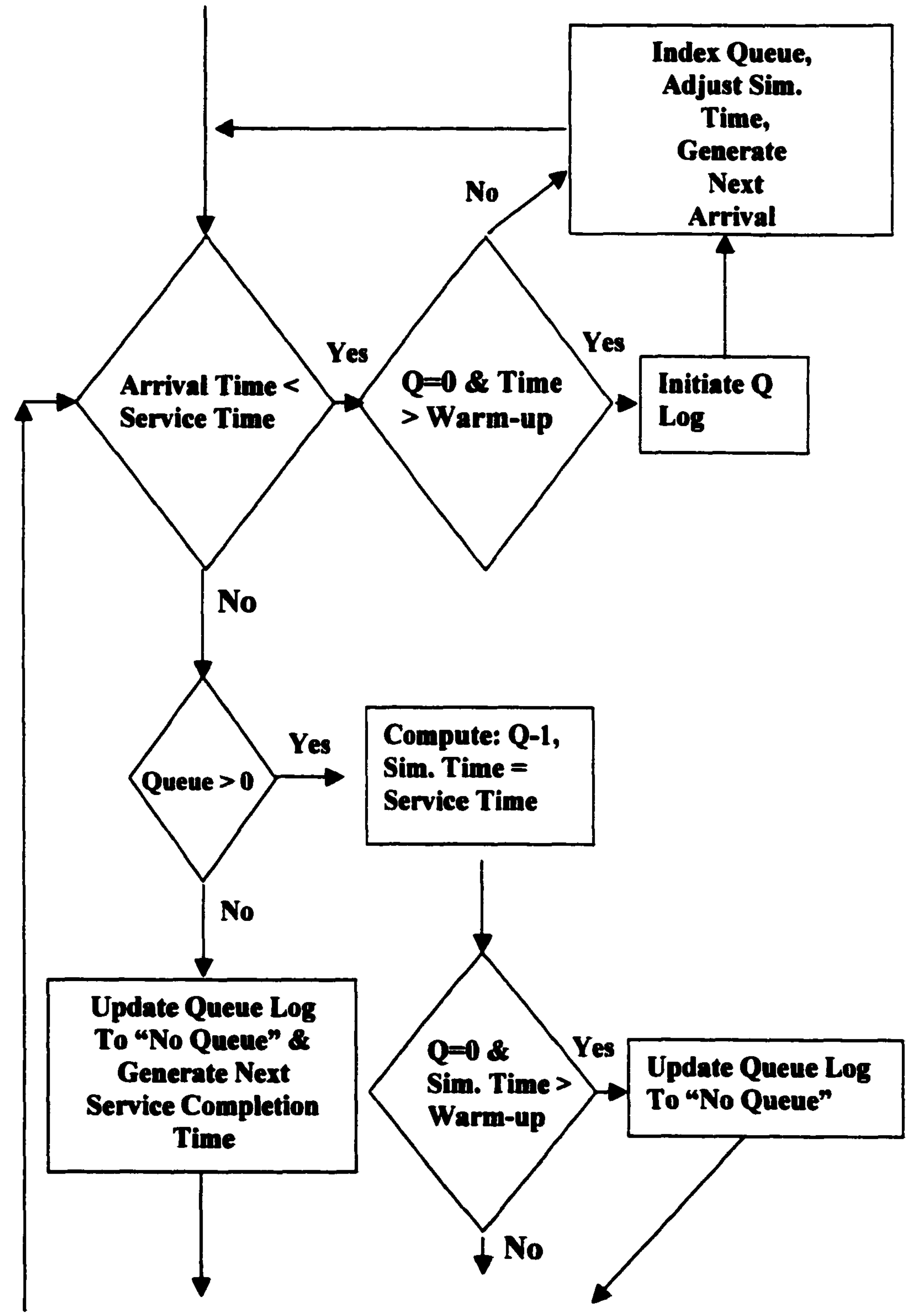
The sample fit data shows that a single distribution does not standout as dominant. The lognormal and the Weibull are, however, near or at the top of each of the fit listings. Thus, the data has the desired variability.

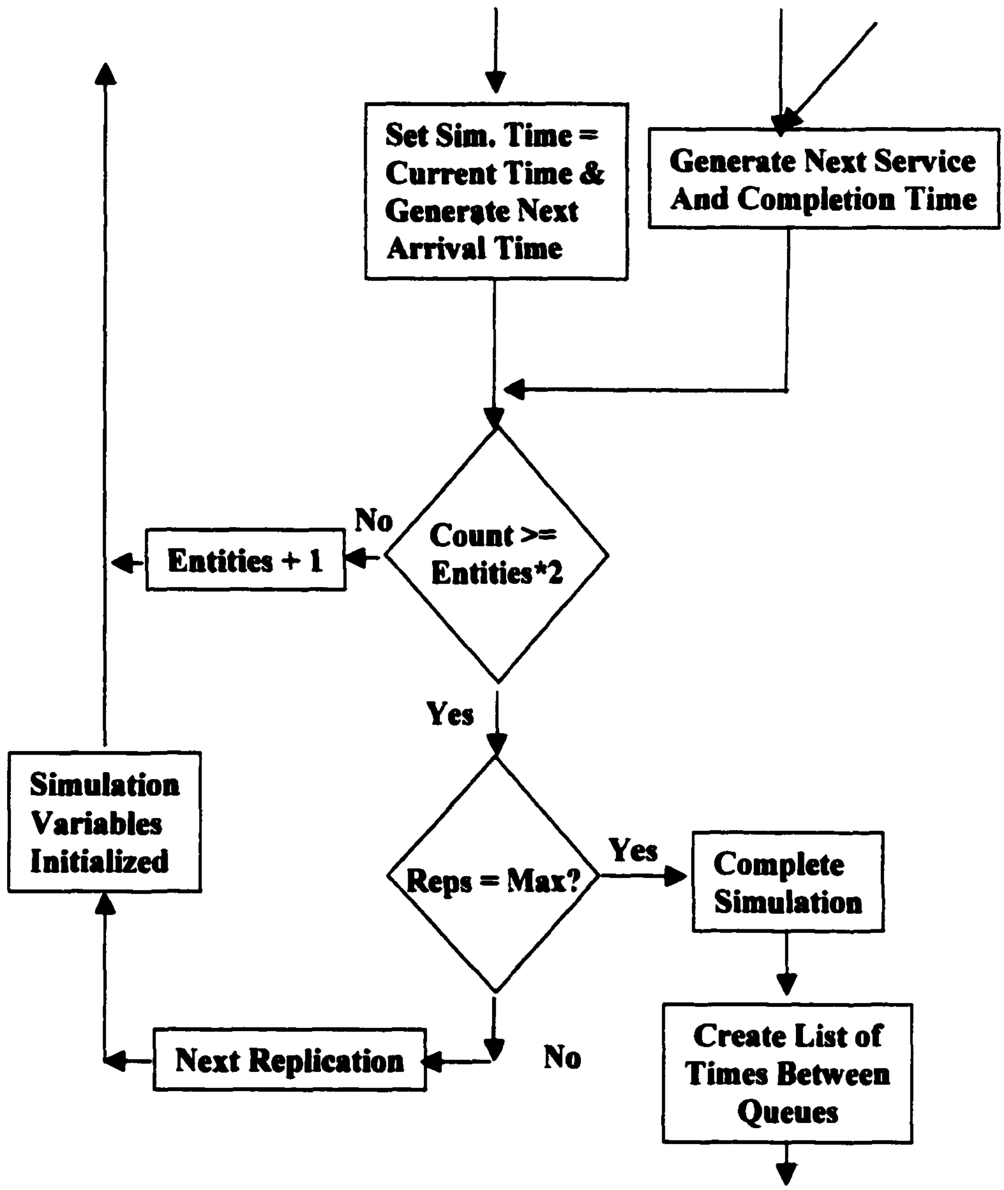
Appendix D

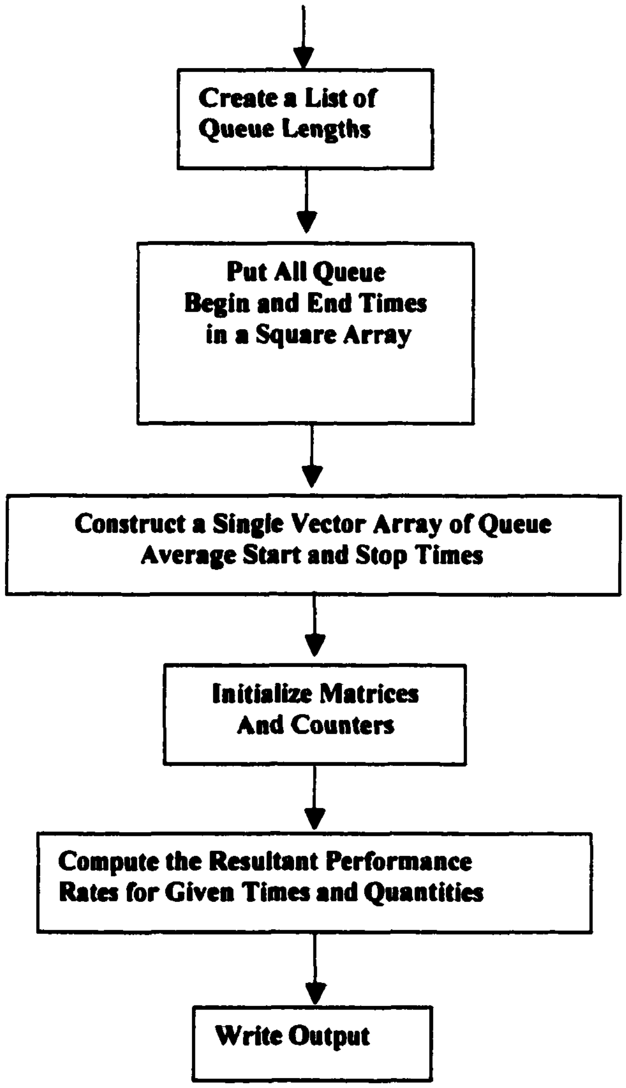
Computer Program Flow Chart and Code

Computer Program Flow Chart




Index Queue, Adjust Sim. Time, Generate Next Arrival
No
Yes
Arrival Time < Service Time
Yes
Q=0 & Time > Warm-up
Yes
Initiate Q Log
No
Queue > 0
Yes
Compute: Q-1, Sim. Time = Service Time
No
Update Queue Log To "No Queue" & Generate Next Service Completion Time
Q=0 & Sim. Time > Warm-up
Yes
Update Queue Log To "No Queue"
No


Set Sim. Time = Current Time & Generate Next Arrival Time
Generate Next Service And Completion Time
No
Count >= Entities*2
Entities + 1
Yes
Simulation Variables Initialized
Reps = Max?
Yes
Complete Simulation
Next Replication
No
Create List of Times Between Queues


Create a List of Queue Lengths
Put All Queue Begin and End Times in a Square Array
Construct a Single Vector Array of Queue Average Start and Stop Times
Initialize Matrices And Counters
Compute the Resultant Performance Rates for Given Times and Quantities
Write Output

Program Listing

```
/* Program to compute frequency of Forgetting and
      length of each Forgeting period. Methodology is
      to approximate the infrequent forgetting periods
      and length of the period with a simulation of a
      queue. The arrival rate to the queue is non-constant.
      Forgetting continues as long as entities are in the
      queue.                          */

/*  Production Run Included        */
/*  This model changes the learn u factor to +.05 versus -.05
                    as in the other models   */

      #include<iostream.h>
      #include<stdio.h>
      #include<stdlib.h>
      #include<time.h>
      #include<math.h>
      #include<malloc.h>
      #define Run_Size 500
      #define MULT1    24112.
      #define MULT2    26143.
      #define Run_Size2 1000



      double unif(long);
      void Randomize(void);
      double Exponential(double);
      float sim_time, diff, upperb,lowerb,mum=0.,QTMax;
      int count=1, QTime_Count, llcount, kkount, out_ct=0;
      float Server_c[1000];
      float huge arr_time[4000];
      float huge next_QTime[4000];
      long  MULTIPLIER=48271.;
      float Generate_Arrival(int, int);
      float Server_comp (float);
      float random_generator(float lowerb, float upperb);
      void Time_Bet_Queues(float,int);
      void QLengths(float,int);
      float huge betq[3000];
```

```
float huge qlen[3000];
float huge Rep_Avg_TBQ[3000];
float huge queue[1000][15];
float Go_Results(int,float,float,int,float,float,float,float,int,int,int);
float comp_learn_nf(float,int);
float comp_lf(float,int,int);
float huge lf_QF_u[200][50];
double first_factor,sec_factor,third_factor,resultant;
double QF;


float random_generator(float lowerb, float upperb)
	{
		float rn, diff, rnum;
		diff = (upperb-lowerb) + 1.;
		rn = random(diff);
		rnum = lowerb + rn;
		return rnum;
	}


/*  Returns a random number on [0,1].
	Call with 0 as the argument for U[0,1].
	Call with positive argument (integers <=9)
	to specify a seed.			*/

double unif(long seed)
{
	static long SEED;
	static long q, r; float newnum;
	long MULTIPLIER = 48271.;
	long MODULUS = 2147483647;
	long t;

	if (seed > 0)
	{
		q = MODULUS / MULTIPLIER;
		r = MODULUS % MULTIPLIER;
		SEED = seed;

}
	newnum = (SEED % q);
	t = MULTIPLIER * (SEED % q) - r * (SEED / q);
```

```
        if (t>0)
                SEED = t;

        else
                SEED = t + MODULUS;
        count = count + random(200);
        return ((float) SEED / MODULUS);
        }


        float Server_comp (float serv_mean)
        {
                float serv_time,r, unum;
                r = 1.0 / serv_mean;
                unum = ((rand() % 100) /100.001);
                serv_time = (-1.0 / r * log(1.0 - unum));
                return serv_time;
        }

        float Generate_Arrival(float lowerb,float upperb)
{
        float next_arr;      // generate arrival rate
        float r, rnumb;
        r = (1.0 / random_generator(lowerb,upperb));
        rnumb = ((rand() % 100) /100.001);
        next_arr = (-1.0/r * log(1 - rnumb));
        return next_arr;
}
 void Time_Bet_Queues(float QTMax,int out_ct)
        {
                        float betq[500], TBQ_sum,TBQ1_sum,TBQ2_sum,
                                TBQ3_sum, sum_ct=1, Rep_Avg_TBQ,
                                Rep_Avg_TBQ1, Rep_Avg_TBQ2,
                                Rep_Avg_TBQ3;
                        int lcount,l,llcount,rep_ct=0;
                        FILE *TBQ;
                        TBQ = fopen("TimeBetQ", "w+");
                        FILE *TBQ1;
                        TBQ1 = fopen("TimeBet1", "w+");
                        FILE *TBQ2;
                        TBQ2 = fopen("TimeBet2", "w+");
                        FILE *TBQ3;
```

```
                    TBQ3 = fopen("TimeBet3", "w+");
                    lcount = 0; llcount = 2;

TBQ_sum=0.,TBQ1_sum=0.,TBQ2_sum=0.,TBQ3_sum=0.;

                    for (l=1;l<(QTMax/2);l++){
                        if (next_QTime[llcount]== (-2.5))
                            {
                            fclose(TBQ):
                            fclose(TBQ1);
                            fclose(TBQ2);
                            fclose(TBQ3);
                            return;
                            }
                        if ((next_QTime[llcount] == (-1.5))
                            && (next_QTime[llcount+1]==(-2.5)))
                            {
                            fclose(TBQ);
                            fclose(TBQ1);
                            fclose(TBQ2);
                            fclose(TBQ3);
                            return;
                            }
                        if (next_QTime[llcount - 1] == (-1.5)){
                                llcount = llcount + 2;
                                 rep_ct++;
                                 sum_ct = 1.;
                            }
                        if (next_QTime[llcount - 1] == (-1.5)){
                                llcount = llcount + 2;
                                 rep_ct++;
                                 }
                        if (next_QTime[llcount] == (-1.5)){
                                llcount = llcount + 3;
                                rep_ct++;
                                sum_ct = 1.;
                            }
                        if (llcount < out_ct){
                            betq[lcount] = next_QTime[llcount] -
                                next_QTime[llcount-1];
                            if (betq[lcount] > 0.0){
                                if (rep_ct == 0){
```

```
                                        TBQ_sum = TBQ_sum +
                                        betq[lcount];
                                        Rep_Avg_TBQ =
                                        TBQ_sum/sum_ct;
                                                sum_ct++;
                                                }
                                        if (rep_ct == 1){
                                                TBQ1_sum = TBQ1_sum
                                                        + betq[lcount];
                                                Rep_Avg_TBQ1 =
                                                        TBQ1_sum/sum_
                                                        ct;
                                                sum_ct++;
                                                }
                                        if (rep_ct == 2){
                                                TBQ2_sum = TBQ2_sum
                                                        + betq[lcount];
                                                Rep_Avg_TBQ2 =
                                                        TBQ2_sum/sum_
                                                        ct;
                                                sum_ct++;
                                                }
                                        if (rep_ct == 3){
                                                TBQ3_sum = TBQ3_sum
                                                        + betq[lcount];
                                                Rep_Avg_TBQ3 =
                                                        TBQ3_sum/sum_
                                                        ct;
                                                sum_ct++;
                                                }
                                        }
                        }
                        llcount = llcount + 2;
                        lcount++;
                        }
                free(betq);
                }

void QLengths(float QTMax, int out_ct)
{
        float
                qlen[500],q_sum,q1_sum,q2_sum,q3_sum,Avg_q,Avg_
                q1,
```

```
        Avg_q2,Avg_q3, q_ct=1;
int k, kcount,kkcount, rep_ct = 0;
FILE *QTimes;
QTimes = fopen("QueueT", "w+");
FILE *QTimes1;
QTimes1 = fopen("QueueT1", "w+");
FILE *QTimes2;
QTimes2 = fopen("QueueT2", "w+");
FILE *QTimes3;
QTimes3 = fopen("QueueT3", "w+");
kcount = 0;   kkcount = 0;
q_sum=0.;
q1_sum=0.;
q2_sum=0.;
q3_sum=0.;
for (k=1;k<QTMax; k++){

        if ((next_QTime[kkcount]==(-1.5))&&(next_Qtime
                [kkcount+1]==(-2.5))){
                fclose(QTimes);
                fclose(QTimes1);
                fclose(QTimes2);
                fclose(QTimes3);
                free(qlen);
                return;
                }

        if ((next_QTime[kkcount+1]==(-1.5))&&(next_Qtime
                [kkcount+2]==(-2.5))){
                fclose(QTimes);
                fclose(QTimes1);
                fclose(QTimes2);
                fclose(QTimes3);
                free(qlen);
                return;
        }
        if (next_QTime[kkcount] == (-1.5)){
                kkcount = kkcount + 1;
                rep_ct++;
                q_ct = 1;
        }
        if (next_QTime[kkcount+1] == (-1.5)){
                kkcount = kkcount + 2;
```

```
        rep_ct++;
        q_ct = 1;
        }
if (next_QTime[kkcount+1]!= (-1.5)){
        if (next_QTime[kkcount+1] != (-2.5)){
                qlen[kcount] = next_QTime[kkcount+1]
                - next_QTime[kkcount];
        if (qlen[kcount] > 0.0){
                if (rep_ct == 0){
                        fprintf(QTimes, "%f
                                QTimes\n",qlen[kcount]);
                        q_sum = q_sum + qlen[kcount];
                        Avg_q = q_sum/q_ct;
                        q_ct++;
                        }
                if (rep_ct == 1){
                        fprintf(QTimes1, "%f
                                QTimes1\n",qlen[kcount]
                                );
                        q1_sum = q1_sum +
                        qlen[kcount];
                        Avg_q1 = q1_sum/q_ct;
                        q_ct++;
                        }
                if (rep_ct == 2){
                        fprintf(QTimes2, "%f
                                QTimes2\n",qlen[kcount]
                                );
                        q2_sum = q2_sum +
                        qlen[kcount];
                        Avg_q2 = q2_sum/q_ct;
                        q_ct++;
                        }
                if (rep_ct == 3){
                        fprintf(QTimes3, "%f
                                QTimes3\n",qlen[kcount]
                                );
                        q3_sum = q3_sum +
                        qlen[kcount];
                        Avg_q3 = q3_sum/q_ct;
                        q_ct++;
                        }
        }
```

```
                    kcount++;
                    }
                    if (kkcount >=(out_ct+1)){
                    fclose(QTimes);
                    free(qlen);
                    return;
                    }
                    kkcount = kkcount + 2;
                    }
        }
        free(qlen);
}

float Go_Results(int t,float Q,float Qforget, int tcomp,float alpha,float
        beta,float QN, float gamma, int type, int u, int TprodRun)
{
                float Q1,Q2;

                double QF;
                double lf_qf, third_factor,xy,yx,yz,zy,zz,xyz,
                        abc,def,ghi,lf_QF,tau, psi;
                lf_QF = 0;xy = .09;yx = .05;zy = .1;yz = .2;zz = 20.0;
                tau = 20;
                psi = 10;

                if (tcomp >= tau)
                        Q2 = tcomp-tau;
                else
                        Q2 = 0.;
                if (tcomp >= psi)
                        Q1 = tcomp-psi;
                else
                        Q1 = 0.;

                if (type == 2){   // Type 2 means forgetting is
                        occurring.
                        if (Q > Q2){
                                QF = Q2;
                                first_factor = alpha
                                        *(20.0*(pow(QF,0.09))
                                        +pow(u,0.05));//comp_learn_nf(
                                        QF,u);
                                }
```

```
		else
			{
			QF = Q2;
			xyz = pow(QF,xy);
			abc = pow(u,yx);
			def = pow(Qforget, zy);
			ghi = pow(u, yz);
			lf_QF = ((zz * xyz) + abc - (def * ghi));
			first_factor = alpha * lf_QF;
			}
		if(Q > Q1){
			QF = Q1;
			sec_factor = beta *
				(20.0 * (pow(QF,0.09)) +
				pow(u,0.05));
			}
		else {
			QF = Q1;
			xyz = pow(QF,xy);
			abc = pow(u,yx);
			def = pow(Qforget, zy);
			ghi = pow(u, yz);
			lf_QF = ((zz * xyz) + abc - (def * ghi));
			sec_factor = beta * lf_QF;
			}

		QF = tcomp;
		xyz = pow(QF,xy);
		abc = pow(u,yx);
		def = pow(Qforget, zy);
		ghi = pow(u, yz);
		lf_QF = ((zz * xyz) + abc - (def * ghi));
		third_factor = gamma * lf_QF;
		resultant = first_factor + sec_factor +
			third_factor;
		return resultant;
	}

	if (type ==1)  // No forgetting
			{
			QF = Q2;
			first_factor = pow(u,0.05);
```

```
                first_factor =
                        first_factor+(20.0*(pow(QF,
                        0.09)));
                first_factor = first_factor*alpha;

                QF = Q1;
                sec_factor = beta * (20.0 * (pow(QF,
                        0.09)) + pow(u,0.05));

                QF = tcomp;
                third_factor = gamma * (20.0*(pow(QF,
                        0.09))+pow(u,0.05));
                resultant = first_factor + sec_factor +
                        third_factor;
                return resultant;
                }
if (type ==3){

// Type 3 means no forgetting and forgetting.
                if(Q2 > QN)
                        {
                        QF = Q2;
                        first_factor = alpha *
                                (20.0 * (pow(QF, 0.09)) +
                                pow(u, 0.05));
                        QF = Q1;
                        sec_factor = beta *
                                (20.0 * (pow(QF, 0.09)) +
                                pow(u, 0.05));

                        QF = tcomp;
                        third_factor = gamma *
                                (20.0 * (pow(QF, 0.09)) +
                                pow(u, 0.05));
                        resultant = first_factor +
                                sec_factor + third_factor;
                        return resultant;
                        }
                if (Q2 < QN)
                        {
                        QF = Q2;
                        xyz = pow(QF,xy);
                        abc = pow(u,yx);
```

```
                    def = pow(Qforget, zy);
                    ghi = pow(u, yz);
                    lf_QF = ((zz * xyz) + abc - (def
                          * ghi));
                    first_factor = alpha * lf_QF;
                    }
            if (Q1 < QN)
                    {
                    QF = Q1;
                    xyz = pow(QF,xy);
                    abc = pow(u,yx);
                    def = pow(Qforget, zy);
                    ghi = pow(u, yz);
                    lf_QF = ((zz * xyz) + abc - (def
                          * ghi));
                    sec_factor = beta * lf_QF;
                    }
            else{
                    QF = Q1;
                    sec_factor = beta * (20.0 *
                          (pow(QF, 0.09)) + pow(u,
                          0.05));
                    }
            QF = tcomp;
            third_factor = gamma * (20.0 *
                    (pow(QF, 0.09)) + pow(u, 0.05));
            resultant = first_factor + sec_factor +
                    third_factor;
            return resultant;
            }
    if (type ==5){
            QF = TprodRun;
            Qforget = t - TprodRun;
            xyz = pow(QF,xy);
            abc = pow(u,yx);
            def = pow(Qforget, zy);
            ghi = pow(u, yz);
            resultant = ((zz * xyz) + abc - (def *
                    ghi));
            if (resultant < 0.0)resultant = 0.0;
            return resultant;
    }
}
```

```
void main ()
{

        long entities;
        int Queue, i, j, serv_ct, arr_ct, reps, QTime_Count=0, toggle,
                out_ct=0, type, que_ct, t, u, u_ct=0, t_ct=-1, tic, Tcycle,
                Tprod, TcycleRun, TprodRun, Noprod, tcomp;
        float sim_time, serv_mean, upperb, lowerb, diff, warm_up,
                Avg_Que, row_sum, avg_queues[500], Q, alpha, QN,
                beta, gamma, max_time, max_u,
                array_delta,Qforget=0.0,rep;
        float QTMax, QTime;
        int row_count, row_min, l, line_no;
        FILE *fp;
        fp = fopen("outqueue", "w+");
        FILE *Que;
        Que = fopen("Queues","w+");
        FILE *Avgque;
        Avgque = fopen("Avg_Que","w+");
        FILE *que_array;
        que_array = fopen("QueArray","w+");
        FILE *op;
        op = fopen("Output","w+");
        FILE *queue_length;
        queue_length = fopen("q_length","w");
        FILE *nextQT;
        nextQT = fopen("next_QT", "w");
        cout<<"\n\t\tA Research Project Into Learning/Forgetting
                Curves";
        cout<<"\n\n\n\tThis simulation of a single server queue is used
                to generate\n";
        cout<<"\t\tqueue lengths and times between queues.\n\n";
        cout << "\tEnter the number of replications desired: ";
        cin >> reps;
        cout << "\tEnter the number of entities per replication:  ";
        cin >> entities;
        cout << "\tEnter the Warm Up Period Before Data Collection
                Begins:  ";
        cin >> warm_up;
        cout << "\tEnter the Upper bound on the arrival input: ";
        cin >> upperb;
```

```
        cout << "\tEnter the Lower bound on arrival input:  ";
        cin >> lowerb;
        diff = upperb - lowerb;
        // Server Rate
        cout << "\tEnter the server input:  ";
        cin >> serv_mean;
        serv_mean = serv_mean*.1;
        cout << "\n\tThe Sum of the Period Weights Must Equal 1.0.
                \n";
        cout << "\t  Enter the Weight Factor for t - 4;  ";
        cin >> alpha;
        alpha = alpha * .1;
        cout << "\t  Enter the Weight Factor for t - 2;  ";
        cin >> beta;
        beta = beta * .1;
        cout << "\t  Enter the Weight Factor for t; "   ;
        cin >> gamma;
        gamma = gamma * .1;
        cout << "\n\tEnter the Maximum Time:  ";
        cin >> max_time;
        cout << "\tEnter the Maximum Number of Units:  ";
        cin >> max_u;
        cout << "\tEnter the Incremental Delta:  ";
        cin >> array_delta;
        cout << "\n\tEnter the Production Cycle T:";
        cin >> Tcycle;
        cout << "\tEnter Production Run Time:";
        cin >> Tprod;

        randomize();    // Sets the Seed.

// Loop for the number of Replications

for (i=0;i<reps;i++)
{
        sim_time = 0;
        serv_ct = 0;
        arr_ct = 0;
        Queue = 0;
        toggle = 0;
        tic = 0;
        sim_time = Generate_Arrival(lowerb,upperb);
 //     cout <<"the value of first arrival is: "<<sim_time<<'\n';
```

```
arr_time[arr_ct] = sim_time;
arr_ct++;

// Generates Service Completion Time

Server_c[serv_ct] = Server_comp(serv_mean) + sim_time;

arr_time[arr_ct] = Generate_Arrival(lowerb,upperb) + sim_time;


// Loop for the number of Entities Generated Per Replication


for (j=1;j<(entities * 2);j++){

// Determine Next Event

    if ((sim_time >=warm_up) && (tic==0)) tic = 1;

        if (arr_time[arr_ct] < Server_c[serv_ct])
            {
            if ((tic == 1)&&(toggle == 0)){
            fprintf(fp,"%f  New Q
\n",arr_time[arr_ct]);
             next_QTime[QTime_Count] =
                  arr_time[arr_ct];
             out_ct++;
             toggle = 1;
             QTime_Count++;
            }
            Queue++;        // Add One to the Queue
            sim_time = arr_time[arr_ct];
            if (sim_time >= warm_up){
            if (Queue > 0)
             fprintf(queue_length,"%d        %f\n",
                  Queue,sim_time);
            }
            arr_ct++;

             // Generate an Arrival
```

```
                arr_time[arr_ct] = Generate_Arrival
                        (lowerb,upperb)+ arr_time[arr_ct
                        - 1];
        }

        else
                if(Queue > 0){
                        Queue--;
                        sim_time = Server_c[serv_ct];
                        if (sim_time >= warm_up){
                        if (Queue > 0)
                        fprintf(queue_length,"%d
                                %f\n", Queue,sim_time);
                        }
                        if  ((Queue == 0)&& (tic==1)
                                &&(toggle == 1)){
                                fprintf(fp,"%f  No Q \n",
                                Server_c[serv_ct]);
                                next_QTime[QTime_Cou
                                nt] = sim_time;
                                out_ct++;
                                QTime_Count++;
                                toggle = 0;
                        }
                        serv_ct++;
                        Server_c[serv_ct] =
                                Server_comp(serv_mean)
                                + sim_time;
                        }
                else    //if (Queue == 0)
                {
                                serv_ct++;
                        Server_c[serv_ct] =
                        Server_comp(serv_mean)+
                        arr_time[arr_ct];
                                sim_time =
                                arr_time[arr_ct];
                                if (sim_time >=
                                warm_up){
                                if (Queue > 0)
                                fprintf(queue_length,"%d
                                %f\n",Queue,sim_time);
                                }
```

```
                                        arr_ct++;
                                        arr_time[arr_ct] =
                                        sim_time +
                                        Generate_Arrival(lowerb,
                                        upperb);
                                }

                        }

        .                       fprintf(fp,"%f   End of Iteration\n", -
                                        1.5);
                                next_QTime[QTime_Count] = -1.5;
                                out_ct++;
                                QTime_Count++;

                }

                        fprintf(fp,"%f   End of Simulation\n", -2.5);
                        next_QTime[QTime_Count] = -2.5;
                        QTMax = QTime_Count;
                        out_ct++;

                fclose(queue_length);
                fclose(fp);

                    /* Create a list of Queue Start/Stop Times (Queue Lengths)
                            Plus Time Betweeen Queues.                    */

                        //Time_Bet_Queues(QTMax,out_ct);

                        //QLengths(QTMax,out_ct);

                    // Put the Queue Begin and End Times in a Square Array.

                    for (i=0;i<200;i++){  // Initializing the "queue".
                            for (j=0;j<reps;j++){
                                    queue[i][j] = 0.;
                            }
                    }
                    for (i=0;i<QTime_Count;i++){
                            fprintf(nextQT, "%f    \n", next_QTime[i]);
        }
```

```
        row_min = 100;
        row_count = 1;
        llcount = 0;
        line_no = 0;

        for (i = 0; i<reps; i++){
                row_count = 0;
                for (l=line_no; l < QTime_Count; l++){
                        if (next_QTime[l]!= (-1.5)){
                                queue[llcount][i] =
                                next_QTime[l];
                                fprintf(Que,"%f   \n",
                                queue[llcount][i]);
                                row_count++;
                                llcount++;
                                }
                        if ((next_QTime[l] == (-1.5))
                                &&(next_QTime[l+1]!= (-2.5))){
                                line_no = l+1;
                                llcount = 0;
                                break;
                                }

                        if ((next_QTime[l] == (-1.5)) &&
                                (next_QTime[l+1] ==(-2.5)))
                                 break;

                }
                        if (row_count < row_min)
                                        row_min = row_count;
        }
// Squaring the array with no zeros at the end of each column
// and making each column the same length and eliminating
//any queue start times without recorded end times.

        if ((row_min % 2)> 0)
                        row_min = row_min - 1;

        // Print Queue Array to file "QueArray".

        for (i=0;i<row_min;i++){
                        for (j=0;j<reps;j++){
```

```
                                        fprintf(que_array,"%f   ",
                                        queue[i][j]);
                                }
                                fprintf(que_array,"\n");
                }


// Construct a Single Element Array of Queue Time Averages.

        for (i=0; i<row_min;i++){
                row_sum = 0.;
                for(l=0; l<reps; l++){
                        row_sum = row_sum + queue[i][l];

                }
                rep = reps;
                avg_queues[i] = row_sum/rep;
        }
        for (j=0; j<row_min; j++){
                avg_queues[j]=avg_queues[j]-warm_up;
                if(avg_queues[j]<0.0)avg_queues[j]=0.0;
                fprintf(Avgque, "%f   \n", avg_queues[j]);
        }

/* This routine computes the resultant Performance Rates for given
                times and quantities.                                  */

        TcycleRun = Tcycle;
        TprodRun = Tprod;
        Noprod = Tcycle - Tprod;
        que_ct = 0;
        tcomp = - array_delta;
        max_time = avg_queues[row_min - 1] + (10.0);
        for (t = 0; t < max_time; t = t + array_delta){
                t_ct++;
                u_ct = 0;
                type=0;
                if (t>TcycleRun){
                                TcycleRun = TcycleRun + Tcycle;
                                TprodRun = TprodRun + Noprod +
                                        Tprod;
                                tcomp = t - Noprod + array_delta;
                }
```

```
else
                tcomp = tcomp + array_delta;
        if (t > TprodRun && t < TcycleRun) type = 5;
        for (u = 10; u < max_u; u = u + array_delta){
if ((avg_queues[que_ct+2]<t) &&((que_ct+2)<=row_min-1)){
        que_ct = que_ct + 2; cout << "In the loop";
}
                        if((t >= avg_queues[que_ct])&&(t <=
                                avg_queues[que_ct+1]))
                                Qforget = t-avg_queues[que_ct];
                        else
                                Qforget = t;
                        if (avg_queues[que_ct] < t){
                        if ((avg_queues[que_ct+1] < t) &&
                                (avg_queues[que_ct+2] > t)){
                                        if (type != 5)type = 3;
                                        Q = avg_queues[que_ct];
                                        QN = avg_queues[que_ct
                                                + 1];
                                        lf_QF_u[t_ct][u_ct] =
                                        Go_Results(t,Q, Qforget,
                                        tcomp,alpha, beta,QN,
                                        gamma, type,
                                        u,TprodRun);
                                        if (lf_QF_u[t_ct][u_ct]<=
                                                0.0)lf_QF_u[t_ct][
                                                u_ct] = 0.0;
                                        fprintf(op, "%d \t %d \t
                                                %f   \n",t,u,
                                                lf_QF_u[t_ct][u_c
                                                t]);
                                        u_ct++;
                                        if ((avg_queues
                                        [que_ct+1] < t) &&
                                        (avg_queues[que_ct+2]<
                                        (t +
                                        array_delta))&&(que_ct+
                                        2 < row_min-1))
                                        que_ct = que_ct + 2;
                                }
                                else
                                {
                                        if (type != 5)type = 2;
```

```
                        Q = avg_queues[que_ct];
                        QN = avg_queues[que_ct
                        + 1];
                        lf_QF_u[t_ct][u_ct] =
                        Go_Results(t,Q, Qforget,
                        tcomp,alpha,
                        beta,QN,gamma, type, u,
                        TprodRun);
                        if(lf_QF_u[t_ct][u_ct]<=
                        0.0)lf_QF_u[t_ct][u_ct] =
                        0.0;
                        fprintf(op, "%d \t %d \t
                        %f   \n", t,u,
                        lf_QF_u[t_ct][u_ct]);
                        u_ct++;
                    }
                }

                else {
                    if (type != 5)type = 1;
                    Q = avg_queues[que_ct];
                    QN = avg_queues[que_ct + 1];
                    lf_QF_u[t_ct][u_ct] =
                    Go_Results(t,Q, Qforget,
                    tcomp,alpha,   beta,QN, gamma,
                    type, u, TprodRun);
                    if(lf_QF_u[t_ct][u_ct] <=
                        0.0)lf_QF_u[t_ct][u_ct] =
                        0.0;
                    fprintf(op, "%d \t %d \t %f   \n",
                        t,u,lf_QF_u[t_ct][u_ct]);
                    u_ct++;
                }


            }
        }
}
```