# ABSTRACT


Title of Document: AN EVALUATION OF END OF MAINTENANCE DATES AND LIFETIME BUY ESTIMATIONS FOR ELECTRONIC SYSTEMS FACING OBSOLESCENCE

Anthony Joseph Konoza, Master of Science, 2012

Directed By: Professor Peter Sandborn, Department of Mechanical Engineering

The business of supporting legacy electronic systems is challenging due to mismatches between the system support life and the procurement lives of the systems' constituent components. Legacy electronic systems are threatened with Diminishing Manufacturing Sources and Material Shortages (DMSMS)-type obsolescence, and the extent of their system support lives based on existing replenishable and non-replenishable resources may be unknown. This thesis describes the development of the End of Repair/End of Maintenance (EOR/EOM) model, which is a stochastic discrete-event simulation that follows the life history of a population of parts and cards and operates from time-to-failure distributions that are either user-defined, or synthesized from observed failures to date. The model determines the support life (and support costs) of the system based on existing inventories of spare parts and cards, and optionally harvesting parts from existing cards to further extend the life of the system. The model includes: part inventory

segregation, modeling of part inventory degradation and periodic inventory inspections, and design refresh planning.

A case study using a real legacy system comprised of 117,000 instances of 70 unique cards and 4.5 million unique parts is presented. The case study was used to evaluate the system support life (and support costs) through a series of different scenarios: obsolete parts with no failure history and never failing, obsolete parts with no failure history but immediately incurring their first failures with and without the use of part harvesting. The case study also includes analyses for recording subsequent EOM and EOR dates, sensitivity analyses for selected design refreshes that maximize system sustainment, and design refresh planning to ensure system sustainment to an end of support date.

Lifetime buys refer to buying enough parts from the original manufacturer prior to the part's discontinuance in order to support all forecasted future part needs throughout the system's required support life. This thesis describes the development of the Lifetime Buy (LTB) model, a reverse-application of the EOR/EOM model, that follows the life history of an electronic system and determines the number of spares required to ensure system sustainment. The LTB model can generate optimum lifetime buy quantities of parts that minimizes the total life-cycle cost associated with the estimated lifetime buy quantity.

# AN EVALUATION OF END OF MAINTENANCE DATES AND LIFETIME BUY ESTIMATIONS FOR ELECTRONIC SYSTEMS FACING OBSOLESCENCE

By

Anthony Joseph Konoza

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2012

Advisory Committee:
Professor Peter Sandborn, Chair
Professor Abhijit Dasgupta
Professor Patrick McCluskey

# Dedication

This thesis is dedicated to my mother, my friends and family, and my officemates in graduate school.

# Acknowledgements

I would like to thank my advisor, Professor Peter Sandborn, for his continued support and guidance throughout my graduate career. I also wish to thank Harris Corporation for providing data and feedback into the practical applications of the methodology presented in this thesis. I would also like to thank the more than 100 companies and organizations that support research at the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland annually.

Table of Contents

# List of Tables

# List of Figures

# Nomenclature

$\sigma$, population standard deviation

$\mu$, population mean

$\mu_{DR}$, population mean for a selected design refresh

$\mu_{TC}$, population mean for the associated test case (no design refresh).

$\Delta SL$, support loss rate

$a$, lower bound of generated uniform distribution

$AFD_{im}$, adjusted forecasted demand of the $i$th part from the $m$th card

$b$, upper bound of generated uniform distribution

$C_{1,2}$. specified calendar dates for a given card where $C_2 > C_1$

$C_A$, administrative costs

$C_C$, card inventory costs

$C_D$, disposal costs

$C_{DR}$, cost per refresh

$C_H$, harvested part inventory costs

$C_{i,}$, recurring cost of holding a part in inventory to the $i$th maintenance event

$C_I$, inspection costs

$C_{IH}$, inventory holding costs

$C_{Inf}$, infrastructure costs

$C_M$, cost of maintenance activities

$C_{NRE}$, nonrecurring costs

$C_{OB}$ , overbuy penalty cost

$C_{PG}$ , packaging costs

$C_{PI}$ , part inventory costs

$C_{PN}$ , penalty costs

$C_{PQ}$ , part qualification costs

$C_{PR}$ , procurement cost

$C_{R}$ , replacement costs

$C_{sys}$ , system support cost

$C_{SQ}$ , supplier qualification costs

$C_{TLC}$ , total life-cycle cost

$C_{TS}$ , test/screening costs

$C_{UB}$ , underbuy penalty cost

$CF$ , card clumping factor ($n$-to-1, where $n$ is an integer of the number of cards being approximated as 1)

$CI_{j}$ , fielded number of $j$th cards

$CPI_{ij}$ , clumping approximation for a total number of part demands for the $i$th part on the $j$th card

$D_{A}$ , calendar year of the start of analysis in the model

$D_{FF}$ , calendar year of the first observed failure

$D_{i}$ ,$D_{i-1,}$ difference in years between the $i$th and previous maintenance event date

$Ds$ , calendar year the part was fielded

$DD_{ij}$, *i*th part forecasted degradation date from the *j*th inventory

$e$, standard error of the mean

$F_{p,}$ number of fielded parts

$FD_i$, *i*th part forecasted demand date

$FD_{im}$, *i*th part forecasted demand date from the *m*th card

$FDD_{ij}$, forecasted degradation date for the *i*th part from the *j*th inventory

$H_0$, null hypothesis

$H_1$, alternate hypothesis

$H_i$, preserved life fraction of *i*th part incurred from the physical action of harvesting (0-1)

$I_{i,}$, infrastructure costs associated with the *i*th maintenance event

$IN_{ij}$, date of the *i*th inspection event for the *j*th inventory

$IP_j$, period of time elapsed between each inspection event for the *j*th inventory

$ISD_j$, calendar date at which inspections will begin for the *j*th inventory

*k,* index used to identify a particular constraint

*K,* number of constraints

$L_{ij}$, remaining fraction of useful life for the *i*th harvested part from the *j*th card

$L_{t,}$, quantity of parts purchased at the lifetime buy

$LTB_{i,}$, lifetime buy quantity of parts stored in inventory at the *i*th maintenance event

$M_{i,}$ maintenance activity costs associated with the *i*th maintenance event

$\overline{M}_i$, *i*th-ordered mean EOM time

$M_{ij}$, *i*th-ordered EOM time in the *j*th life history

$n$, number of maintenance events

$N_j$ , number of inspections that occurred prior to this event for the *j*th inventory

$N_{ij}$ , number of occurrences as an *i*th-ordered EOM in the *j*th life history - either 1 (occurs) or 0 (does not occur)

$N_f$ , number of failures observed to date

$N_T$ , total number of fielded parts within the entire system

$NRE_{i,}$, *i*th non recurring cost

$O_P$ , operational hours per year

$p$ , sample size

$P$, procurement cost per part

$P_i$ , *i*th-ordered EOM probability

$PN_{i,}$, penalty costs associated with the *i*th maintenance event

$PQ_{ij}$ , part quantity of the *i*th part from the *j*th card

$Q_{i,}$ , quantity of parts stored in inventory at the *i*th maintenance event

$Q_{LTB}$ , lifetime buy quantity

$Q_{TQN}$ , total quantity needed

$Q_{TQN_i}$ , cumulative total quantity at the *i*th maintenance event

$r$, number of non recurring costs

$R$, after tax discount rate on money

$s$, number of life histories simulated

$t$ , current simulation time

$t_H$ , simulation time when harvesting action occurs

$t_i$ , simulation time when *i*th part was introduced into system

$U_{1,2}$ , cumulative unsupportability fraction for a given card evaluated at simulation time $C_{1,2}$ respectively

$y_b$, base year for money

# Chapter 1 : Introduction

The long-term sustainment of electronic systems is a challenging task for system supporters. Sustainment becomes even more of a challenge when the electronic system is part of a mission/safety critical system (i.e., a sub-system whose failure results in the failure of system operations, e.g., aerospace or military systems). The sustainment problem varies from system to system and encompasses a large number of factors including part reliability, electronic part obsolescence, required system availability, and supply chain and inventory management, all while trying to minimize system life-cycle costs. Legacy system[1] supporters have three fundamental concerns:

1. How long can my system be sustained based on the resources that I currently have (i.e., how much time do I have before I have to do something)?
2. What will be the cause of the eventual loss in systems operations (i.e., why is my system functionality going to be hindered)?
3. How much will supporting (e.g., operations, maintenance) my system cost?

If the supported system is comprised of available parts (i.e., parts that are still commercially procurable from their original manufacturers), then the supporter can readily purchase more parts and the system can continue to be supported and operational (allowing for logistics delays). However, many legacy systems are

[1] In this thesis, legacy systems are defined as fielded and operational systems for which no new system production is planned.

comprised of obsolete parts that are no longer available from their original manufacturers, and therefore, other strategies must be implemented after the spares for obsolete parts are depleted in order to retain system functionality.

Electronic part obsolescence[2] (referred to as Diminishing Manufacturing Sources and Material Shortages [DMSMS]-type obsolescence in this thesis) can occur at any moment during the system life cycle—many parts become obsolete even before the system is placed into service. Supporters of mission-critical systems facing obsolescence must find alternative methods to ensure system sustainment as failures of these systems could lead to catastrophic damages. This thesis proposes a model that addresses the fundamental concerns faced by supporters of legacy electronic systems (i.e., how long before, and what causes the loss of systems operations) regarding electronic part obsolescence, and aids system supporters in strategic management of their electronic systems.

### *1.1 Commercial Off-The-Shelf (COTS) Obsolescence*

In an effort to reduce system support and development costs, mission-critical systems designers shifted towards the use of commercial off-the-shelf (COTS) parts as a substitute for "government unique" parts. The introduction of COTS parts led to less expensive volume production, elimination of the confinement to single source purchasing, and increased application flexibility, but it had a negative side that also brought about its own set of problems [1].

[2] Electronic part obsolescence occurs when a part manufacturer discontinues a part, making it no longer procurable from the original source. Note, the part may remain procurable from aftermarket suppliers or may be superseded by a newer version of the part.

COTS parts created difficulties for many applications that include stringent system requirements (i.e., functionality or supportability) and specific environmental operating conditions. Additionally, the use of COTS parts can lead to a loss of supply chain control (i.e., it binds users to volatile market trends where technology continuously evolves) [2]. The key difference between mission-critical and commercial systems is that mission-critical systems often have requirements of 25-year or longer support lives (which are commonly extended), but the commercial parts that comprise these systems have limited procurement and support lives. Furthermore, commercial suppliers have no obligations for providing continued support or sales to mission-critical systems, leaving supporters of these systems at an ongoing risk of obsolescence. The truth of the matter is that the defense industry (often supporters of mission-critical systems) makes up a very small percentage of the total market share for commercial electronic parts, and therefore, has no control over the behavior of the commercial electronics market that they depend on.

In response to the evolution of electronic technologies, commercial suppliers must periodically introduce new or upgraded parts and discard or discontinue the support of older parts—it may be impractical for them to satisfy every customer. Eventually the supplier will discontinue the production of parts that some customers still need (i.e., the supplier's profit margin begins to decline), thereby, leaving system supporters at an impasse in dealing with DMSMS-type obsolescence [3]. Inventory or sudden obsolescence [4] refers to the opposite problem of DMSMS-type obsolescence. Inventory obsolescence occurs when design or system specifications change such that specific spare parts are no longer required or useful. This thesis considers the problem of system sustainment when faced with DMSMS-type obsolescence, and not inventory obsolescence.

DMSMS-type obsolescence is an unavoidable problem due to mismatches between system support life requirements and the procurement lives of the systems' constituent parts. The problem associated with obsolescence is that mission-critical systems have high qualification and certification requirements, meaning that even minor design changes to the system prove to be financially burdensome. The result of COTS obsolescence inevitably leads to higher system life-cycle costs, therefore becoming a major cost driver in systems that frequently experience long support lives (e.g., military and aerospace systems). The estimated costs for the U.S. Navy due to obsolescence are approximately $750 million annually [5].

The obsolescence problem is typically associated with systems considered "sustainment-dominated"; i.e., systems whose long-term sustainment (life-cycle) costs exceed their original procurement costs [6]. Examples of sustainment-dominated systems include avionics, naval systems, nuclear power plants, air traffic control systems, and medical equipment. Sustainment-dominated systems are low-volume and have long field lives (often 20 years or more). Sustainment-dominated systems frequently become legacy systems because they become too expensive to replace. Long-term support of these legacy systems eliminates potential redesign or replacement risks and is often less expensive. Redesign or replacement risks include requalification or recertification of the system, Form Fit and Function equivalencies, additional reliability assessments , and possibly consequential changes that might be needed. The focus for system supporters becomes minimizing system life-cycle cost while maximizing system support—this problem is typically resolved through a variety of reactive obsolescence mitigation approaches.

Reactive obsolescence mitigation approaches, although not a solution to the DMSMS-type obsolescence problem, provide the supporter with ways to manage the problem tactically. Reactive management approaches include: alternate or substitute parts, aftermarket sources, lifetime buys[3], thermal uprating of parts, and emulated parts [7]. The model described in this thesis focuses on strategies that use existing stocks (often the result of part lifetime buys—a developed model focused on estimating lifetime buy quantities is presented in Chapter 4) of parts and reclamation to extend system support life based on currently owned excess parts and fielded legacy systems. Both of these strategies mitigate the obsolescence problem through the use of existing resources (fielded parts and part spares) in hopes of extending the system support life. Having addressed the system sustainment challenges when faced with DMSMS-type obsolescence, we can then begin to develop the electronic system sustainment problem that this thesis addresses.

Electronic systems are commonly composed of systems of printed circuit assemblies, hereafter referred to as cards, which are circuit boards that contain electronic parts. As time elapses, obsolete parts on these cards fail and must be replaced using inventories of non-replenishable spare parts. As the non-replenishable inventories become depleted, system supporters ask: how long can the system(s) last based on the current number of spares and how can support costs of the system be quantified?

[3] Lifetime buy refers to buying enough parts from the original manufacturer prior to the part's discontinuance in order to support all forecasted future part needs throughout the system's required support life.

These questions become difficult to answer when one starts to consider system capabilities, uncertainties, and complexities. Therefore, the goal of this thesis is to develop a model that accurately describes the above characteristics of a legacy electronic system faced with DMSMS-type obsolescence (including unique parts, cards, inventories, reliabilities, etc.) and quantify the system support life and support costs as a function of the capabilities, uncertainties, and complexities of the system.

The following section discusses demand forecasting, which is an important factor in capturing the characteristics of legacy electronic systems operations and support.

### *1.2 Demand Forecasting*

Demand forecasting is a crucial issue in inventory management and plays a significant role in electronic systems sustainment modeling. The ability to forecast future part demands allows system supporters to predict when parts fail (or when spares become depleted) and implement risk management or mitigation plans (e.g., logistics management). For legacy systems, the demand forecasting challenge is developing a methodology that accurately forecasts part demands based on historical failure data.

Demand forecasting of parts to support a system is most commonly performed using renewal functions [8,9]. Renewal functions predict the number of renewal (part failure) events in a specific period of time and are a common method used to determine warranty reserve funds for products. However, renewal functions only calculate the expected number of events in a time period, not the respective dates that they would actually take place (the function only provides an expected number of events). This approach is not suited to characterize the support for a legacy electronic system, as there may be some periods where no part failures occur and other periods where an extensive number of part failures occur. Additionally, renewal functions and other basic sparing and warranty models are generally confined to calculating renewals for populations of parts represented by a single probability distribution. In order to effectively model an electronic system that is composed of populations of unique parts and cards, one would have to evaluate each unique population of parts individually (assuming these populations of parts do not draw from the same inventories) and then determine the system support life by finding the earliest time one of the evaluated population sets could not be supported.

Croston's method and variants thereof [10-13] are a common approach for intermittent demand forecasting involving exponential smoothing forecasts based on the size of a demand and time period between demands. Croston's method estimates the mean demand per period by applying exponential smoothing separately to the intervals between nonzero demands and their sizes. However, these intermittent demand forecasting methods only provide point forecasts and cannot produce forecast distributions and demand prediction intervals (deterministic forecasting versus stochastic forecasting). In deterministic models, variable states are determined by parameters in the model or by sets of previous states of these variables. Stochastic modeling is the representation of variable states through probability distributions rather than unique values, allowing for randomness to be present. Randomness is necessary for electronic system sustainment modeling because electronic parts do not always fail at the exact same time; one part may fail after 50 operational hours of use compared to an identical part that fails after 500 operational hours of use. The goal of this model is to incorporate the random nature (stochastic process) characteristic of part reliability with spare parts demand for mission-critical systems.

Stochastic processes are capable of gathering a multitude of probable and possible solutions based on associated input uncertainties. These processes allow for system complexities to be fully and accurately explored (i.e., representation of part reliabilities as probability distributions). Stochastic demand forecasting models [14-18] incorporate the inherent randomness associated with spare parts demands, meaning that demand for a part arises only when the part actually fails. The most common models for stochastic forecasting include Markov chain models [19-21], Petri nets (PNs) [22-24], and discrete event simulations [25, 26].

Markov chain models are defined by a random process that incorporates a state of "memorylessness", where the next event state depends only on the current state and not on the sequence of events that preceded it. This means that the probability distribution for the next event is only dependent on the current state and not by previous states. In electronic system sustainment modeling, unique parts should be characterized by unique probability distributions such that every part is modeled independently by its own part demand. The collections of part demands should be organized chronologically based on when they are forecasted to take place; the next event is not necessarily dependent on the current event (i.e., demands are not ordered consecutively, but chronologically). These properties are not suitable for Markov models as the next step may or may not be modeled by the current step (includes multiple quantities of a single part or multiple parts governed by different probability distributions).

PNs offer a formal and graphical technique for representing concurrent, discrete event dynamic systems. PNs are bipartite graphs (whose vertices can be divided into two disjoint sets) such that any connection always connects vertices from different subsets. PNs are useful in describing the process flow and behavior of a system; however, it becomes challenging to graphically represent the process flow for systems containing multiple stochastic parallel processes where the generation of the PNs reachability set (set containing all possible markings [scenarios] that can occur within the system state space) can be costly in terms of time and space [27]. Discrete event simulations account for these uncertainties and possible pathways in fast consecutive simulation executions based on provided system information.

The problem that this thesis hopes to address, described at the end of Section 1.1, requires an approach that involves stochastic demand forecasting unique for every instance of every part. This stochastic model should carry out the events based on the chronological ordering of all previous forecasted demands concerning each type of possible event. The following section addresses the solution for reaching the goal of the thesis through the introduction and use of discrete event simulation modeling.

### *1.3 Discrete Event Simulation Modeling*

A discrete event simulation represents a set of chronological events where each event occurs at an instant in time and marks a change of state in the system. The two primary discrete event simulation models include time-based and event-based simulations. Time-based models follow a chronological process flow of events as they occur at discrete points in simulation time (i.e., a timeline). At each discrete time, the process state is observed precisely; however, the progress between any two consecutive time steps is assumed to be negligible. Thus, time-based modeling techniques assume that important changes to the system (events) only occur at discrete times, and progression of the model is based on the chronological succession of events as they occur within the simulation time horizon. In event-based modeling, the occurrence of the events drives the modeled process (i.e., the model progresses by sequences of events rather than discrete time steps).

Discrete event simulation is often a preferred approach to modeling the maintenance of real systems in order to account for complexities and uncertainties (e.g., part reliabilities, quantities, and inventories) that must be included within the model. The system complexities (model inputs) are stochastically monitored to arrive at a unique solution (model output) using the simulation. In order to quantify the model outputs, it is necessary to consider a large number of simulation histories (histories of sustaining the system) in order to generate model output probability distributions that incorporate the stochastic natures of the inputs to the model. The following section briefly discusses the realm of inventory depletion models and how they pertain to system sustainment faced with DMSMS-type obsolescence.

### *1.4 Inventory Depletion*

There are many inventory depletion models in the literature that incorporate stochastic demand forecasts [28-30]; however, the majority of these are not concerned with the problem of obsolete parts that comprise the inventories. The focus of inventory modeling appearing in the literature is on the management aspect in response to different scenarios (e.g., order quantities, repairable or multiple items, suppliers) and is concerned with optimal inventory management for units that are still currently available from suppliers. The models are not focused on predicting how long the inventory is able to last, but rather the logistics and inventory management in order to accurately account for lead times, demands, etc. The implicit assumption included in most inventory management models is that the units in the inventory are always replenishable (i.e., available for ordering for the foreseeable future). Some attention has been given to inventory management modeling with the occurrence of sudden obsolescence [31-33], but not to the modeling of inventories for systems currently facing DMSMS-type obsolescence.

Discrete event simulations have also been used for maintenance and operations activities [34,35]. SIMAIR [34] is used to simulate daily operations of airlines, modeling the plane's operation as a sequence of events. The Ultra Reliable Aircraft Model (URAM) [35] is designed for investigating Maintenance Free Operating Periods (MFOPs) for maintenance activities in a system. URAM applies an MFOP window on either side of the forecasted point of failure. Rather than have a maintenance window, the model should accomplish its maintenance action in the same discrete time step where the failure was identified.

The following section defines a system's End of Repair (EOR) and End of Maintenance (EOM) and introduces the EOR/EOM model proposed in this thesis that is used to evaluate system sustainment based on existing resources when faced with DMSMS-type obsolescence.

### *1.5 Introduction to End of Repair and End of Maintenance*

The Federal Aviation Administration (FAA) defines *End of Maintenance* (EOM) as "the moment a site requisition cannot be replenished. This stage change begins with the depletion of limited depot and site spares quantities, followed by service degradation (i.e., loss of redundancy) and ultimately loss of system operations." [36]. The last portion of the FAA's definition (loss of system operations) is what the model proposed in this thesis defines as the EOM date for the system. In the model, the EOM date is defined as "the earliest date that all available inventories fail to support the demand for one or more specific parts resulting in the loss of system operation."

Additionally, the FAA [36] defines *End of Repair* (EOR) as "when hardware product support is no longer available by any means or is cost-prohibitive." In this thesis, the EOR date is defined as "the date that the last repair or manufacturing action associated with a part can be successfully performed."

The EOR/EOM model proposed in this thesis is a stochastic discrete event simulation that follows the life history of a population of parts and cards, and determines how long the system can be sustained based on existing inventories of spare parts and cards, and optionally harvesting of parts from existing cards to increase system support life. The model defines the system hierarchy in terms of parts and cards. A “part” refers to the lowest level possible for the system being analyzed, whereas a "card" is the highest level possible. Cards are composed of multiple parts and the same part may appear on different cards (referred to as type of part).

The EOM problem (and support costs) can be formulated as shown in equations (1.1) and (1.2):

$$f_1(\overline{p}) = \sum_{i=1}^{n} D_i - D_{i-1} \tag{1.1}$$

$$f_2(\overline{p}) = \sum_{i=1}^{n} \frac{(D_i - D_{i-1})Q_{i-1}C_{i-1}}{\left(1+\frac{R}{100}\right)^{D_i - y_b}} + \sum_{i=1}^{n} \frac{M_i}{\left(1+\frac{R}{100}\right)^{D_i - y_b}} \tag{1.2}$$

subject to[4]: $g_k(\overline{p}) \geq 0$ ; $k = 1,\ldots,K$

where,

| | |
|---|---|
| $D_i$; $D_{i-1}$ | Difference in years between the *i*th and previous maintenance event date |
| $Q_i$ | Quantity of parts stored in inventory at the *i*th maintenance event |
| $C_i$ | Recurring cost of holding a part in inventory to the *i*th maintenance event |
| $n$ | Number of maintenance events |
| $M_i$ | Maintenance activity costs associated with the *i*th maintenance event |
| $R$ | After tax discount rate on money |
| $y_b$ | Base year for money |
| $k$ | Index used to identify a particular constraint |
| $K$ | Number of constraints |

The objective function, $f_1(\overline{p})$, calculates the EOM date for the system being modeled. The EOM objective function is dependent on $\overline{p} = [p_1,\ldots,p_m]$, which is the set of system parameters that describe the system. The parameters used in the EOM objective function include part reliabilities and quantities, fielded card instances,

[4] The formulated problem assumes that EOM occurs at the first instance where a parametric constraint is violated—this is useful for finding the first EOM, but the model can also be extended (see Chapter 2) to track consecutive EOM and EOR dates within a simulated system life history.

inventories of spare parts and cards, and the operational profile of the system. Some of these parameters are uncertain; however, everything is known about the behavior and range of variation for each parameter. The system begins at a specific start date ($D_0$) and progresses upon arriving at $D_n$, where prior to the event, the considered constraint $g_k(\overline{p})$ equaled 0, and by the end of the time step, $g_k(\overline{p})$ will have been violated (drawing from an inventory that consists of no parts). The EOM date occurs at some maintenance event when a part demand cannot be met by any of the available inventories from which it can be drawn. The EOM objective function can also be applied to calculating the EOR dates for the system; this can occur during any $D_i$ when the last replacement for a part can be successfully performed ($g_k(\overline{p})$ becomes equal to 0).

The objective function, $f_2(\overline{p})$, calculates the support costs for the system being modeled, where the first expression accumulates inventory holding costs[5] and the second accumulates maintenance activity costs (these vary with each discrete event)—this function incorporates the same parameters as equation (1.1). Both objective functions are constrained in the same manner, whereby the system is assumed to be "operating successfully" as long as there are spare parts available for forecasted parts demands (spare parts or cards).

[5] Inventory costs are accumulated considering all accessible inventories and their associated holding costs (i.e., spare card inventories may cost more than inventories of spare parts), the expression has been generalized for the problem formulation.

### *1.6 Research Scope and Objectives*

The first objective of this thesis is to develop a model that determines how long legacy electronic systems drawing from existing non-replenishable inventories of parts and cards, can be sustained, to develop a methodology for calculating the effective EOR and EOM dates for systems composed of multiple cards where each card has multiple parts and parts may appear within more than one card, and to assess the support costs of the system (e.g., ownership of inventories). The second objective of this thesis is to develop a model that calculates the number of spares required to sustain an electronic system to a specific date, and generate the optimum lifetime buy quantity that minimizes the total life-cycle cost associated with the estimated lifetime buy quantity.

The EOR/EOM and LTB models developed in this thesis track every *obsolete* part on every card in the entire system independently. This means that each time-to-failure distribution of each part is sampled, and kept in sorted lists for determining successive chronological events towards model progression. Parts that are commercially available from their original vendor are deemed as "available" and are not included in EOR/EOM analysis. Every part on every card is characterized by its own time-to-failure distribution to account for the uniqueness of common parts across different cards—parts can either be assigned time-to-failure distributions or have their time-to-failure distributions generated based on part failure histories.

The following five research tasks are associated with fulfilling the objectives of this thesis:

**Task 1**: Develop a general methodology that follows the life history of a population of parts and cards and determine how long the system can be sustained based on existing inventories of spare parts and cards. The general methodology should track and calculate the effective EOR and EOM dates (and consecutive EOR and EOM dates) for systems composed of multiple cards where each card has multiple parts (parts may appear on more than one card).

**Task 2**: Develop a method for predicting the impact on EOR and EOM dates of using harvested parts from existing cards, perform modeling of part degradation and periodic inspection of the inventories, and determine a design refresh plan[6] that ensures sustainment of the system to a specific date.

**Task 3**: Implement detailed cost models capable of calculating the system support costs, allowing for the cost of ownership of inventories to be assessed.

**Task 4**: Apply developed methodologies (Tasks 1-3) to a specific case study.

**Task 5**: Develop a reverse application of the EOR/EOM model developed for lifetime buy planning to sustain fielded systems to a specific date. Implement detailed cost models capable of calculating the total life-cycle cost associated with the lifetime buy quantity. Develop a method for finding the optimum lifetime buy quantity that minimizes the total life-cycle cost associated with the lifetime buy.

Chapter 2 discusses the development of the EOR/EOM model. Chapter 2 includes the creation of discrete events within the model (e.g., part demands, part

[6] A design refresh means the replacement of one or more obsolete parts with non-obsolete parts in order to retain the functionality of the system. A design refresh refers to system changes that "must be done" in order for the system functionality to remain viable. A redesign refers to system changes that "are desired", which include both the new technologies to accommodate system functional growth and new technologies to replace and upgrade the existing functionality of the system [37].

degradation, and periodic inventory inspections), the generation of part failure distributions from collected historical failure data, evaluation of concurrent discrete events, and the discrete event modeling process. Additionally, Chapter 2 includes the calculations for EOR/EOM information, system support loss, part harvesting, and system support costs. Chapter 3 presents several simple example cases, and a case study involving an actual legacy electronic system. Lifetime buy quantity forecasting and costing is discussed in Chapter 4.

# Chapter 2 : Model Development

The End of Repair/End of Maintenance (EOR/EOM) model determines the length of time a system is able to support itself when faced with DMSMS-type obsolescence. The model describes the process of inventory depletion of obsolete parts through system operation and tracks the EOR and EOM dates, the critical parts associated with each EOR and EOM event, and the likelihood that these EOR and EOM events will occur. As previously mentioned, *End of Repair* (EOR) is defined as "the date that the last repair or manufacturing action associated with a part can be successfully performed." EOR dates are part-specific and may also be card-specific if a particular card can only draw from a subset of the available inventories. Similarly *End of Maintenance* (EOM) is defined as "the earliest date that all available inventories fail to support the demand for one or more specific parts resulting in the loss of system operation." EOM events are caused by a specific part on a specific card. For example, multiple EOM events would be recorded for a specific part that appears on different cards and draws from the same inventory (assuming there are no existing inventories to draw from).

The model is implemented as a discrete event simulation where system operation is represented as a chronological sequence of events driven by input parameters. In order to account for the inherent uncertainties, some input parameters are defined by probability distributions, and the simulation is run for many simulated life histories to generate probability distributions of the output information. The following sections describe the development and methodology of the EOR/EOM model as it relates to the sustainment problem associated with DMSMS-type obsolescence.

## *2.1 Creation of Discrete Events*

The events included in this analysis are either generated from the sampling of probability distributions of input parameters (e.g., part reliabilities, part degradation in inventory) or entered in a deterministic manner (e.g., periodic inventory inspections or design refreshes), and chronologically ordered (in relation to the simulation time) from earliest to latest. The initial generation and ordering of the discrete events is completed before the simulation clock is initiated.

Each obsolete part in the system is tracked independently throughout the system support life, and therefore, receives its own sampling from the time-to-failure distribution of the part (note, the same part on different cards may have different time-to-failure distributions). For example, if there a five instances of a single part on a card and there are five cards in the system, then there are 25 discrete events sampled from the time-to-failure distribution of the part to represent each of the instances of the part that appear within the system. The model does not generate demands for parts that are considered *available* (i.e., procurable from the original manufacturer) as we are not interested in modeling these parts—the model focuses only on obsolete parts.

The input model parameters (e.g., part reliabilities) can be represented by different distributions (e.g., Uniform, Exponential, Weibull, and Triangular). The sampling of the part demands is performed using Monte Carlo, a sampling technique used for obtaining random values from probability distributions in order to account for uncertainties or risk in quantitative analysis and decision-making processes[38].

The model described in this thesis has the ability to track information regarding individual parts from the moment they are introduced to the field, through failure and replacement, and possibly subsequent failures and replacements through the system support life until EOM (or end of support) occurs. A process flow of a part within the electronic system is depicted in Fig. 2.1.

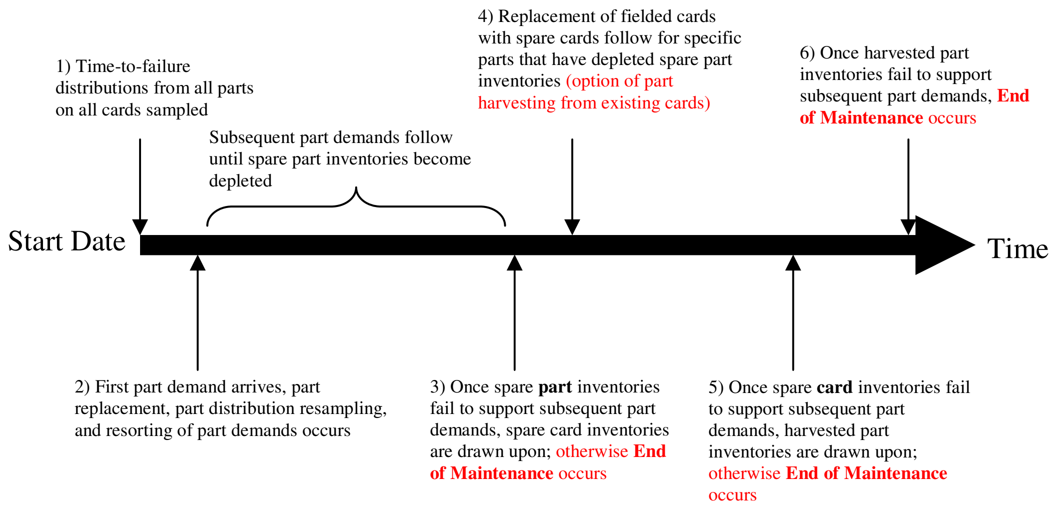



Fig. 2.1 EOR/EOM part failure process flow

The model starts by sampling the time-to-failure distributions of individual parts (referred to as forecasted demand dates for parts or part demand dates) that are located on cards within the system. After all of the part demand dates are sampled within the system, the demand dates are sorted from earliest to latest on a part-by-part basis.[7] The model then determines the earliest part demand date that occurs in the system, jumps ahead to its date, and performs a change to the system (this type of change is dependent on the type of event that occurs). After the change has been

[7] A part is defined as an item that is specific to a particular card that retains its own unique properties (e.g., time-to-failure distribution, quantity). Each instance of a part on a card is treated independently (represented by unique part demand dates sampled from its time-to-failure distribution).

applied at the earliest part demand date, the current part demand date is removed and a "new" part demand date is independently sampled from the time-to-failure distribution of the part and chronologically ordered into its list. After the part demand date at the first event has been removed, the next earliest part demand date is found (representing the second event to chronologically occur), the model jumps to its date, and the process continues. The model continues until the request at a part demand date cannot be fulfilled by spare inventories that previously sustained the demands for that part (i.e., when requests are made, and cannot be met due to a state of inventory stock-out).

The simulation begins at a specified calendar date (referred to as the start date of analysis or the analysis date) and the simulation time progresses until an EOM event occurs (where the request at a part demand date cannot be fulfilled)—this constitutes a single simulated life history of the entire system. In order to obtain an accurate representation of the system support life considering inherent system uncertainties, multiple system life histories are tracked (typically 1,000) in order to produce probability distributions of EOM dates (i.e., system support lives) and to identify the possible part and card combinations (and their associated likelihoods) that caused system support loss.

The next subsection details the methodology of generating time-to-failure distributions for parts from gathered failure histories.

### *2.1.1 Generating Time-to-Failure Distributions From Part Failure Histories*

Sometimes organizations that support legacy systems are uncertain or unaware of the failure characteristics associated with the parts in their systems, but they may have maintenance records containing part failure histories. The historical failure data for an individual part (observed failures to date and the recorded date of the first observed failure) and its total fielded part quantity can be used to generate the time-to-failure distribution for the part. In cases where only a few observed failures have occurred and there is no other existing information (from the part vendor or other sources), uniform time-to-failure distributions can be generated.[8] The generated uniform distribution with lower bound *a* and upper bound *b* for a particular part is given by,

$$a = (D_{FF} - D_S)O_P \tag{2.1}$$

$$b = \frac{(D_A - D_S)O_P - a}{\frac{N_f}{N_T}} + a \tag{2.2}$$

where,

$D_{FF}$ = calendar year of the first observed failure

$D_S$ = calendar year the part was fielded

$D_A$ = calendar year of the start of analysis in the model

$O_P$ = operational hours per year

$N_f$ = number of failures observed to date

$N_T$ = total number of fielded parts within the entire system.

The upper boundary of the distribution, *b*, is dependent on the ratio of failures to date (between the date the part was fielded and the start of the analysis) divided by the

[8] The methodology does not require the characterization of the failure histories for parts as uniform distributions where each value in the range is equally likely to occur. A uniform distribution is only an example treatment that can be used if no other information is known.

total number of fielded parts. When the ratio, $\frac{N_f}{N_T}$ equals 1 (all failures observed prior to the start of the analysis, see Fig. 2.2), the upper bound becomes the difference between the start of analysis and the date the parts were fielded. Likewise, as $\frac{N_f}{N_T}$ approaches 0 (no failures observed), the upper limit of the distribution approaches ∞. When $\frac{N_f}{N_T}$ approaches ∞ (a large number of failures relative to the population of fielded parts), the upper limit of the distribution approaches $a$. It is implicitly assumed that these part sites can exhibit more than one failure, thereby leading to ratios of $\frac{N_f}{N_T}$ greater than 1.

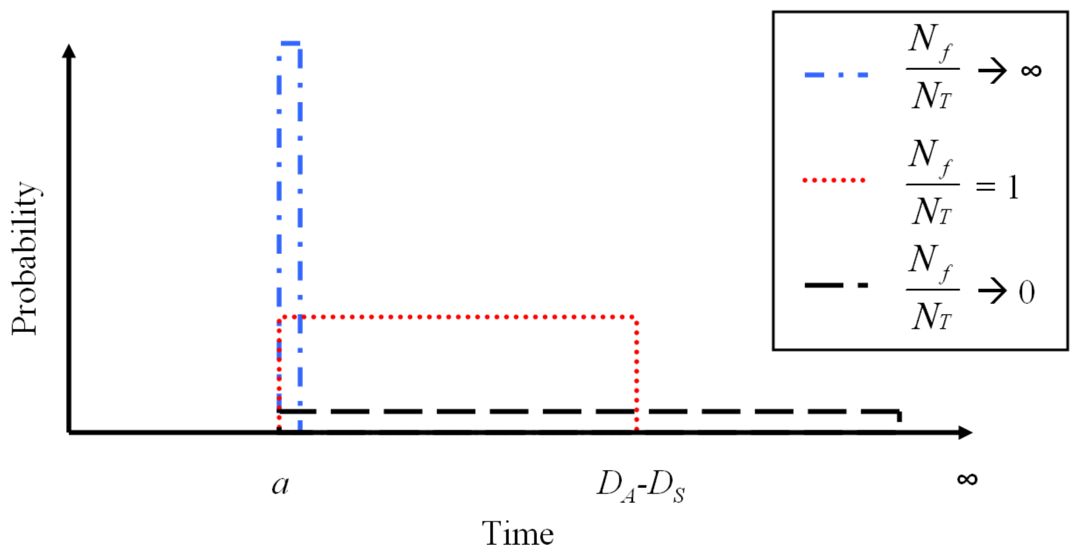



Fig. 2.2 Generated uniform distribution from part failure histories

As previously stated, this generated distribution is useful for parts with unknown failure characteristics. In this manner, one may approximate the time-to-failure distribution for an entire population of parts based on previously observed failure data.

The next question to address becomes, "What is the assumed frequency of these observed part failures?". In reliability engineering, a common failure behavior that parts exhibit occurs in three separate regions, the accumulation of these regions comprise the commonly named "bathtub curve" (see Fig. 2.3).

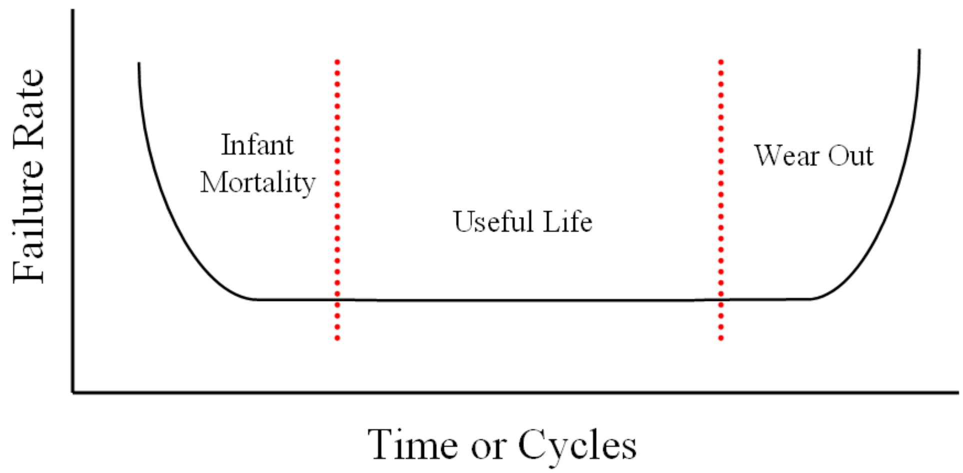


Fig. 2.3 Common electronic part failure behavior curve

*Stage 1: Infant Mortality*

"Infant mortality" is the period of time from when the part is introduced until its failure rate becomes relatively constant. During this period of elapsed time (i.e., the early life of the part), the failure rate is high but rapidly decreases as defective parts are identified and removed from service. In order to weed out the defective parts, part manufacturers may use a series of stress tests during production to identify defects caused by materials or machinery in an effort to weed out the root causes for the defective parts.

Another approach (commonly referred to as burn-in) is to use stress tests as an ongoing screening to weed out defects when the root causes may not be eliminated. Burn-in is a useful approach in lowering the total cumulative failure percentage of parts, but the major trade-off is the cost of performing the test. The question of "When to implement burn-in?" is dependent on the part being manufactured, the projected reliability improvement that will be made from performing burn-in, and the estimated cost and time associated with the burn-in. Although burn-in practices are not usually a practical economic method of reducing infant mortality failures, burn-in has proven effective for state-of-the-art semiconductors where root cause defects cannot be eliminated [39].

*Stage 2: Useful Life*

The "useful life" period of the part is the period of time where the failure rate of the part remains relatively constant. This is in the mid-life of the part, hopefully when it is received by customers, where the failure rate is generally low and approximately constant.

*Stage 3: Wear Out*

The third and final stage of the part behavior is characterized when the part comes to the end of its useful life period, the "wear out" stage. Towards the end of the useful life of the part, the failure rate begins to increase as different factors (i.e., mechanical stress, environmental conditions, etc.) take their toll on the part.

*Failure Behavior Prior to Simulation Analysis*

How is frequency of observed failures perceived in the EOR/EOM model when time-to-failure distributions are generated from part failure histories? There are two assumptions in the model regarding the observed failures. The first assumption is that the analysis date, $D_A$, is after the start of the "useful life" period of the bathtub curve. The second assumption is that all observed failures that are used to create the generated time-to-failure distributions are from the "useful life" period of the bathtub curve. The following section details how obsolete parts with significant failure histories are represented in the EOR/EOM model.

### *2.1.2 Parts Containing Significant Failure Histories with Right Censored Data*

The general approach to obsolete parts within the system involves generating the time-to-failure distribution of the part based on the existing (and limited) failure history. In some cases, there may be an extensive failure history that exists where there are a large number of observed failures (hundreds) recorded for a given part within the system. These parts may also have had a large number of fielded units (i.e., instances of the part) that had not failed (referred to as right censored data[9]). The right censoring also needs to be accounted for in estimating the time-to-failure distributions of the part. The time-to-failure distributions for parts with extensive failure histories were determined using Maximum Likelihood Estimation (MLE) to find the best fit to the failure data using 2-parameter Weibull distributions (i.e., the location parameter is equal to 0) while accounting for the surviving parts using life data analysis software (Weibull++®). The Weibull distribution can be used to model devices with decreasing, constant, or increasing failure rates—this versatility is one reason why it is widely used in reliability.

[9] The failure data is right censored because not all the fielded parts have failed to date. Right censoring occurs in reliability testing when some of the units in the population survive a test time period without failing.

MLE is a technique that is used to estimate the parameters of a statistical model. The derivations of the MLE for the Weibull distributions are provided for complete and censored data sets [40-42]. For Type I censored data, let $f(t)$ be the probability density function (PDF) and $F(t)$ the cumulative distribution function (CDF) for the chosen life distribution model. Note that these are functions of $t$ and the unknown parameters of the model. The likelihood function, $L$, for Type I (right censored) data[43] is given by,

$$L = C\left(\prod_{i=1}^{r} f(t_i)\right)(1 - F(T))^{n-r} \tag{2.3}$$

where,

$C =$ an arbitrary constant

$n$ = number of non-repairable units that undergo testing

$r$ = number of observed failures during testing

$T$= fixed time of the period of testing

The general mathematical technique for solving for MLEs involves setting partial derivatives of the log-likelihood function, $\ln(L)$, equal to zero and solving the resulting (usually non-linear) equations. However, the MLE technique is only useful considering certain conditions are met. MLE should not be used to estimate parameters for statistical models where there are a small number of observed failures (assumed less than 100 for this thesis). MLE's can be heavily biased, and the large sample optimality properties do not apply [43]. Another (technical) drawback is that MLE requires specialized software for solving complex non-linear equations.

The use of MLE in estimating part failure distributions is more appropriate concerning parts with significant failure data with right censoring than the generated uniform distribution. However, in cases with little historical data, it is more appropriate to generate the failure distribution from the available data than to estimate the parameters.

In the investigated case study of this thesis (see Chapter 3), there were two parts (3798-05 and 5004-02) with significant failure data with right censoring. MLE was used to best fit the failure data of these parts to 2-paramter Weibull distributions, while accounting for their right censored data (i.e., parts that had not failed). The comparisons between the MLE-fitted Weibull distributions and the generated uniform distributions for the two parts with significant failure histories can be seen in Figs. 2.4 and 2.5. The generated uniform distributions for both parts were sampled 1,000 times to develop the PDFs seen in Figs. 2.4 and 2.5. The MLE-fitted distributions represented the failure data more accurately than the generated uniform distributions.

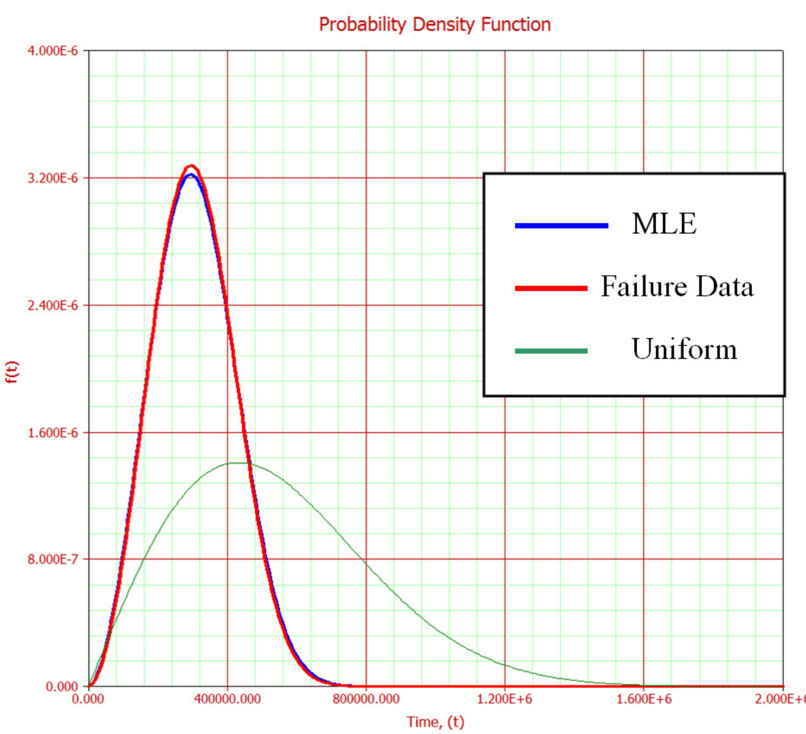



Fig. 2.4 Probability density functions comparing MLE-fit and generated uniform distributions to failure data for part number 3798-05

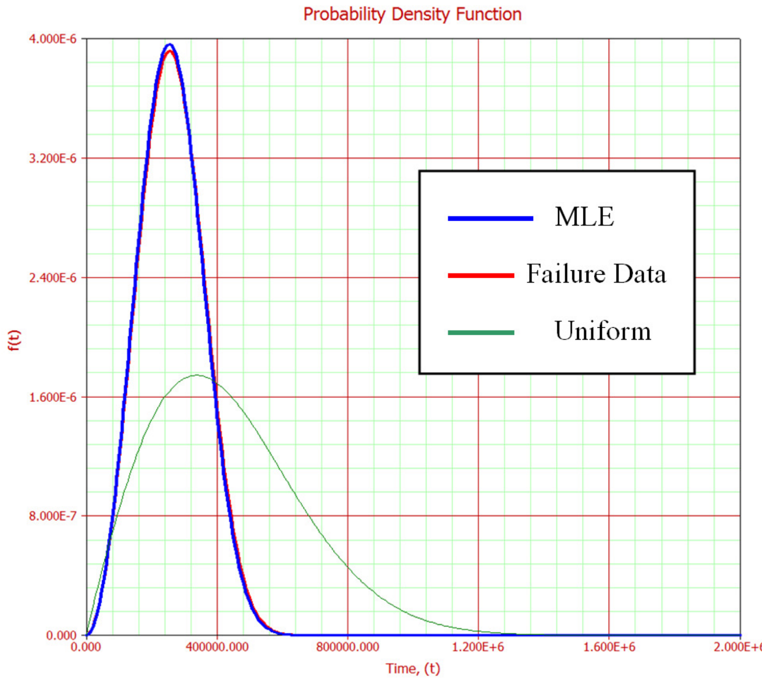



Fig. 2.5 Probability density functions comparing MLE-fit and generated uniform distributions to failure data for part number 5004-02

The resulting failure distributions for these parts (3798-05 and 5004-02) are shown in Figs. 2.6 and 2.7. The two lines on each graph represent the MLE-fitted failure distributions that were created with and without the consideration of the right-censored data for both parts.

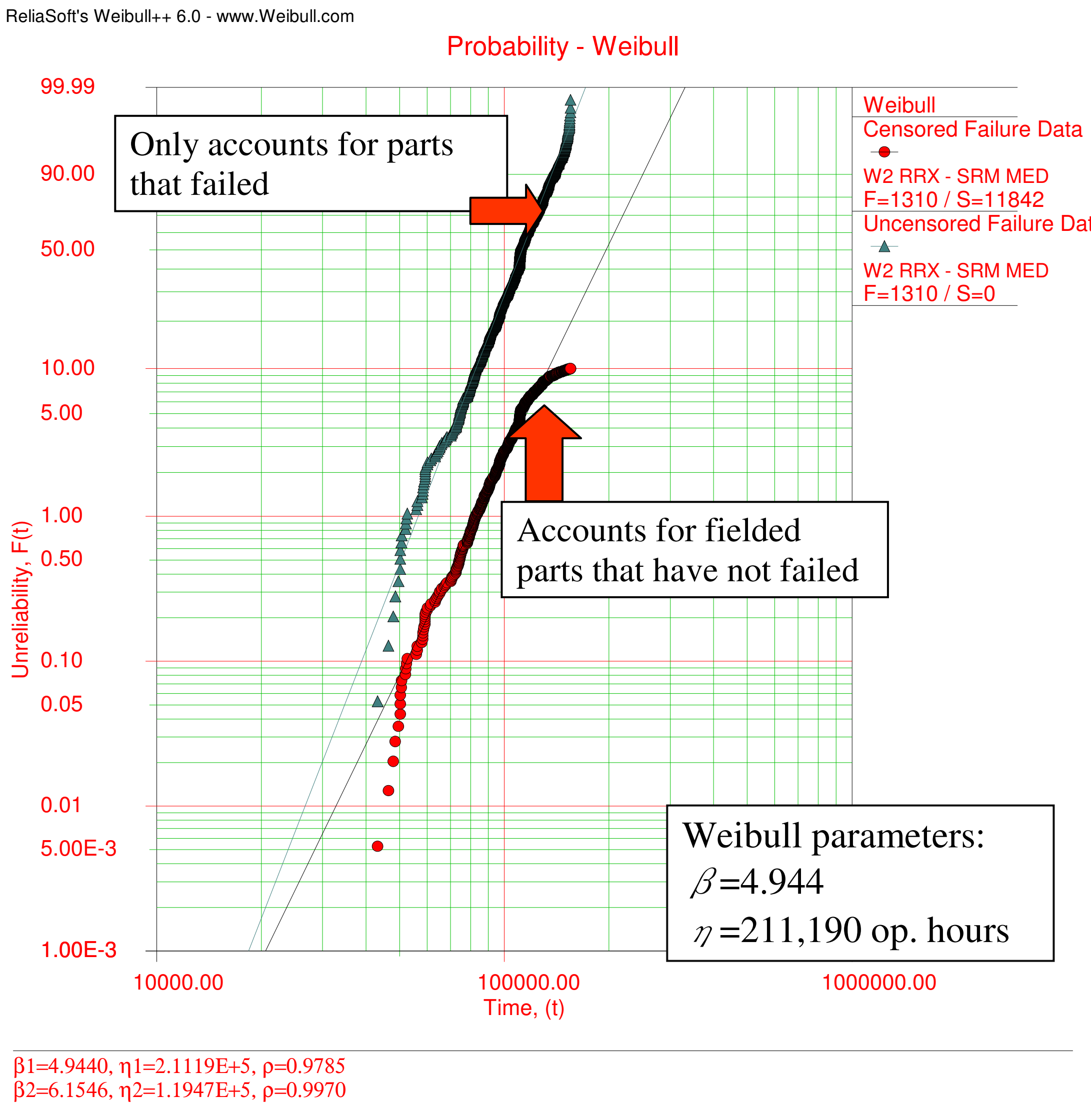



Fig. 2.6 Part number 3798-05 failure distribution. Both data sets are equal, one shows 10% unreliable, the other 100% unreliable (censored vs. uncensored).

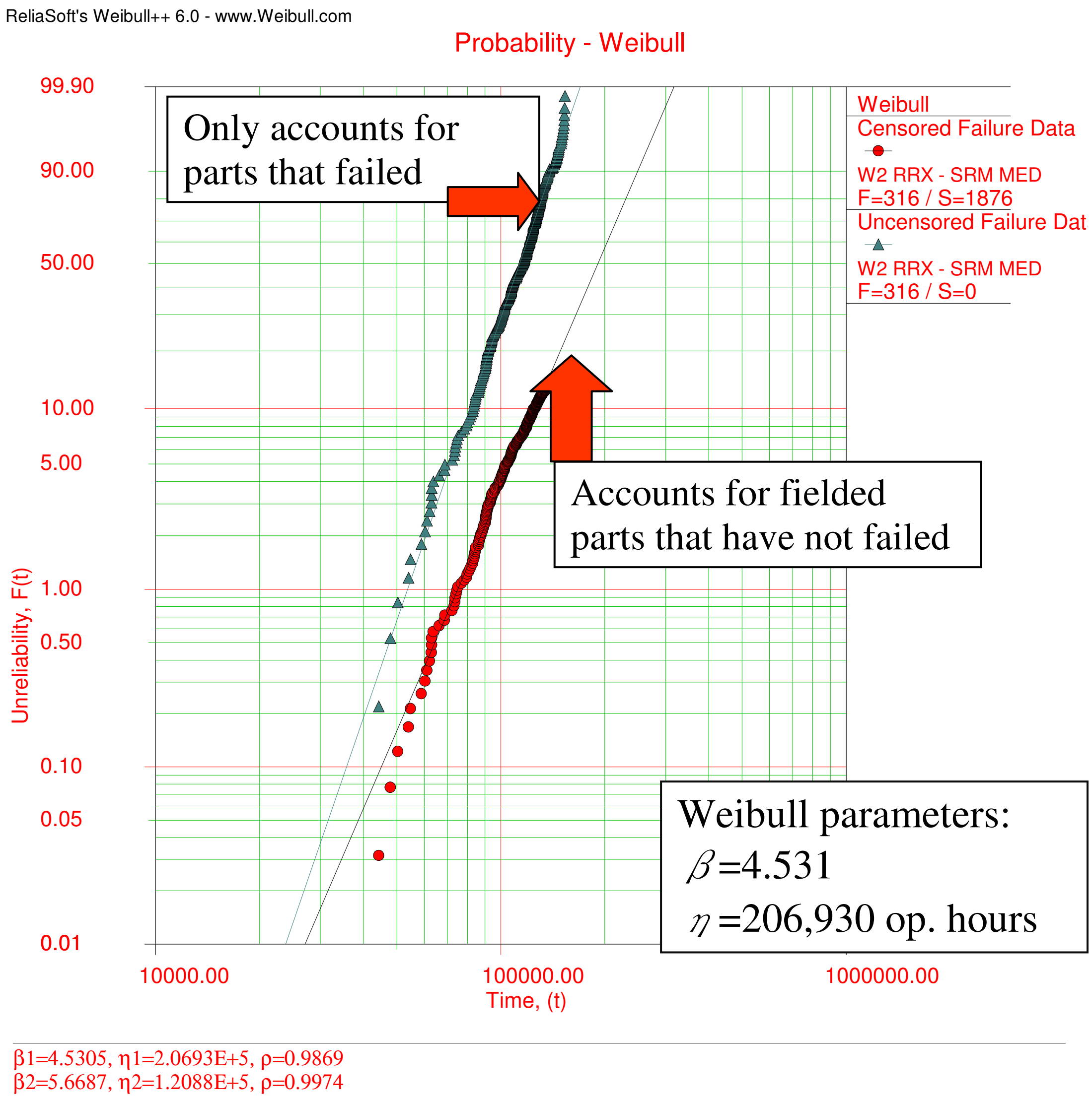



Fig. 2.7 Part number 5004-02 failure distribution. Both data sets are equal, one shows 10% unreliable, the other 100% unreliable (censored vs. uncensored).

Other obsolete parts included in the system from the case study had too few recorded failures to make MLE fitting practical, and their failure distributions were therefore, treated as uniform distributions created from historical failure data as described in equations (2.1) and (2.2).

### *2.1.3 Obsolete Parts With No Previous Failure History*

The model can also account for parts that have no prior observed failure history. Obsolete parts that have no observed failures are, in the best case, implicitly assumed to never fail during analysis. These parts are then not included within the analysis in terms of the creation of forecasted part demands. The question then is, “If the parts that have never failed before suddenly become subject to failures, how will this affect my system support life?” The model answers this question through the assumption of a 'worst-case' scenario, where parts with no previous observed failures incur immediate first failures just prior to the start of analysis. The uniform failure distributions for these parts are then generated based on the immediate single failure in conjunction with their additional historical data.

### *2.1.4 Concurrent Discrete Event Evaluation*

Discrete event simulations operate on the principle of a chronological ordering of a sequence of events. Therefore, the addition of different types of events can easily be implemented without disrupting the existing process flow of the simulation (an advantage of using discrete event simulation over other stochastic models). The addition of multiple event types is evaluated in the same manner as single event-type driven discrete event simulations. The only difference being that there must be chronological ordering not only within single events, but across all events (i.e., finding the earliest date out of all the events and event types within the simulation, see Fig. 2.8). This occurs in the EOR/EOM model when considering additional part degradation and inspection events (see Sections 2.1.5 and 2.1.5.1) that may or may not happen to the stored parts within inventories throughout the system support life.

The next paragraph describes how concurrent events are evaluated within the EOR/EOM model.

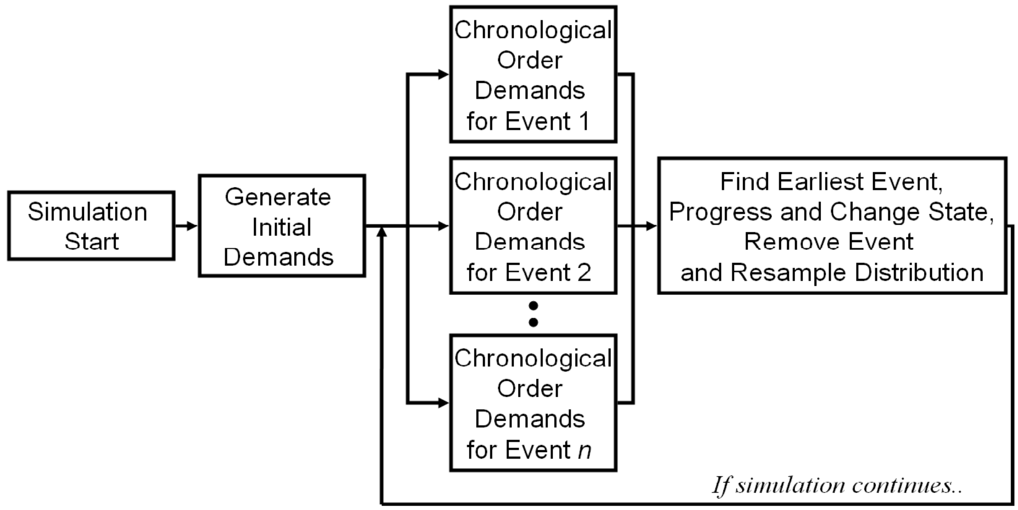



Fig. 2.8 Discrete-event simulation flow for multiple events

First, the model generates the initial number of demands associated with each type of event included in the analysis. Next, the model cycles through the possible types of events that can occur and finds the earliest date associated with each event type. Some events may not occur during every analysis that is executed (i.e., no part degradation) in which case those events are ignored during the evaluation. The model determines the earliest date among all event types and jumps ahead to that date, implements the appropriate changes to the system (dependent on the type of event), removes the date and event that just occurred, and resamples the distribution corresponding to the part (or inventory) that caused the current event. Afterwards, the concurrent event evaluation continues until EOM is reached.

*2.1.5 Modeling of Part Degradation*

Due to the nature of discrete-event simulation, the part degradation event can be emulated through assignments of probability distributions representing the likelihood of a part degrading while it is in inventory in a given time period. The forecasted degradation date for the *i*th part from the *j*th inventory, $FDD_{ij}$ is given by,

$$FDD_{ij} = DD_{ij} + t \tag{2.4}$$

where,

$DD_{ij}$ = *i*th part forecasted degradation date from the *j*th inventory

$t$ = current simulation time (starting at $t$ =0).

The model treats inventory degradation as a recurring event that identifies the degradation of a part from inventory once the forecasted degradation date of the part has been reached, assuming there are remaining spare parts left in the inventory. The degradation distribution of the part is then resampled and the next forecasted degradation date is calculated (where subsequent degraded parts are identified) and the process continues until either the inventory of spares runs out or the EOM date is reached.

This approach identifies the moment that a part has degraded, not discarded from inventory. The degraded parts are discarded either at the next inspection event associated with the inventory or after the attempted use of the degraded part towards replacement (probability of pulling a degraded part or 'good' part from inventory). For example, the probability at any given time that a degraded part is chosen from the inventory is represented by the fraction of degraded parts over the total quantity and is given by,

$$\frac{G_{ij}}{NP_{ij}} \tag{2.5}$$

where,

$G_{ij}$ = number of $i$th parts considered degraded from the $j$th inventory

$NP_{ij}$ = number of $i$th parts remaining from the $j$th inventory.

If $\frac{G_{ij}}{NP_{ij}}$ equals 0, then none of the $i$th parts from the $j$th inventory are considered to be degraded. Conversely, when $\frac{G_{ij}}{NP_{ij}}$ equals 1, then all of the $i$th parts from the $j$th inventory are considered to be degraded. Randomly pulling a part from inventory can be represented by randomly choosing a number between 0 and 1, called $R_N$. If there are degraded parts located in the inventory and a replacement part needs to be pulled, then a good part is pulled from inventory, $R_N > \frac{G_{ij}}{NP_{ij}}$

or a degraded part is pulled from inventory, $R_N < \frac{G_{ij}}{NP_{ij}}$.

The respective quantities ($G_{ij}, NP_{ij}$) are updated after the part is drawn (whether good or degraded) and the simulation continues. The degradation process flow can be seen in Fig. 2.9.

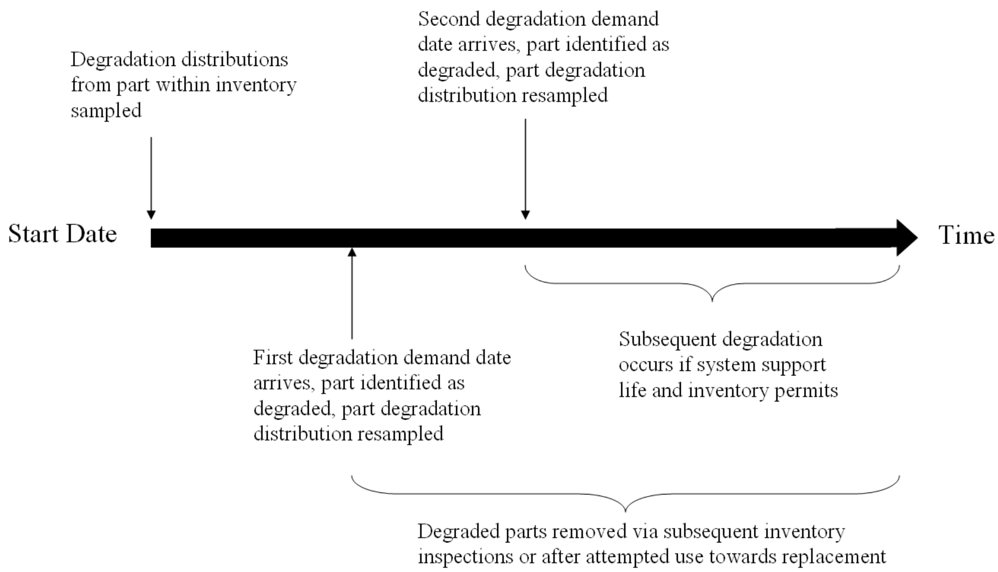



Fig. 2.9 Part degradation process flow

The process flow in Fig. 2.9 demonstrates the degradation for a single part within a single inventory. In the event that additional parts are assigned degradation distributions, additional process flows are added in parallel and are associated with each part involved (each process flow is associated with a single part located in a specific inventory). The EOR/EOM model allows for part degradation probability distributions to be included for each part appearing in specific part inventories within the system.

### *2.1.5.1 Modeling Periodic Inspections of the Inventories*

Periodic inspections are required in order to identify the condition and functionality of stored parts within inventories. The storage of electronic parts is a delicate process that may involve storage facilities that are environmentally controlled. In terms of inspection activities, a number of stored parts from inventory are removed for testing. These parts removed for testing may or may not be replaced after testing.

The EOR/EOM model represents inventory inspection as a periodic recurring event that may result in the removal (and permanent disposal) of a number of parts from inventory as a result of testing. Additionally, the inventory inspections locate and remove degraded parts within the inventory undergoing inspection. The date of the *i*th inspection event for the *j*th inventory, $IN_{ij}$, is given by,

$$IN_{ij} = ISD_j + N_j \cdot IP_j \tag{2.6}$$

where,

$ISD_j$ = the calendar date at which inspections will begin for the *j*th inventory

$N_j$ = the number of inspections that occurred prior to this event for the *j*th inventory

$IP_j$ = the period of time elapsed between each inspection event for the *j*th inventory.

The periodic inventory inspection begins with the first inspection, $ISD_j$, when $N$ equals 0 and consecutively occurs at $IP_j$ periods until either the inventory is exhausted of parts (assuming the inspection withdraws parts from inventory) or EOM is reached for the system.

### *2.1.6 Card Clumping Approximation*

At some level of system complexity the simulation evaluation and execution process becomes arduous. The simulation time is directly affected by the input system complexity. The system complexity is dependent on the total number of forecasted demands for obsolete parts in the system. The model is based on a chronological ordering of events, and increasing the number of events increases the simulation time. In order to decrease the number of events, a procedure that effectively 'clumps' together instances of cards within the system to limit the total number of demands has been used.

This card clumping approximation allows for an *n:*1 ratio to be executed for all cards in the system and trims all part demand lists (collections of forecasted demand dates) by a factor of *n*. For example, if there are 50 instances of a part on a card and five cards in the system, then there are 250 part demand dates. If a 5:1 card clump ratio is chosen, then this reduces the part demands to 50 part demand dates, a fifth of its original amount. The trade-off is that every part is not effectively modeled independently—as a result of the grouping of demands, the model may lose some of its accuracy (including calculations of EOR and EOM dates, their causations, and the likelihoods). The clumping approximation for a total number of part demands for the *i*th part on the *j*th card (once enabled), $CPI_{ij}$ is given by,

$$CPI_{ij} = PQ_{ij} \cdot \frac{CI_j}{CF} \tag{2.7}$$

where,

$PQ_{ij}$ = part quantity of the *i*th part from the *j*th card

$CI_j$ = fielded number of *j*th cards

$CF$ = the card clumping factor (*n*:1, where *n* is an integer of the number of cards being approximated as 1).

As the number of part demands is effectively reduced by a factor of *n* and whenever a forecasted demand occurs, the resulting quantity taken from inventory for a part is equal to the card clumping factor, $CF$, rather than one (in order to account for the aggregated cards). Fig. 2.10 demonstrates the card clumping approximation considering the case 1 from the case study presented in Chapter 3. The graph shows the error that some of the other (simpler) model solutions have compared to the solution provided by the discrete-event simulation model. Simpler models are unable to incorporate the complexity offered by the discrete-event simulation, and therefore, must "clump" part demands together in order to simulate larger or more complex systems. This inability to effectively model system complexities (i.e., modeling each instance of each part independently) leads to inaccuracy in the solution.

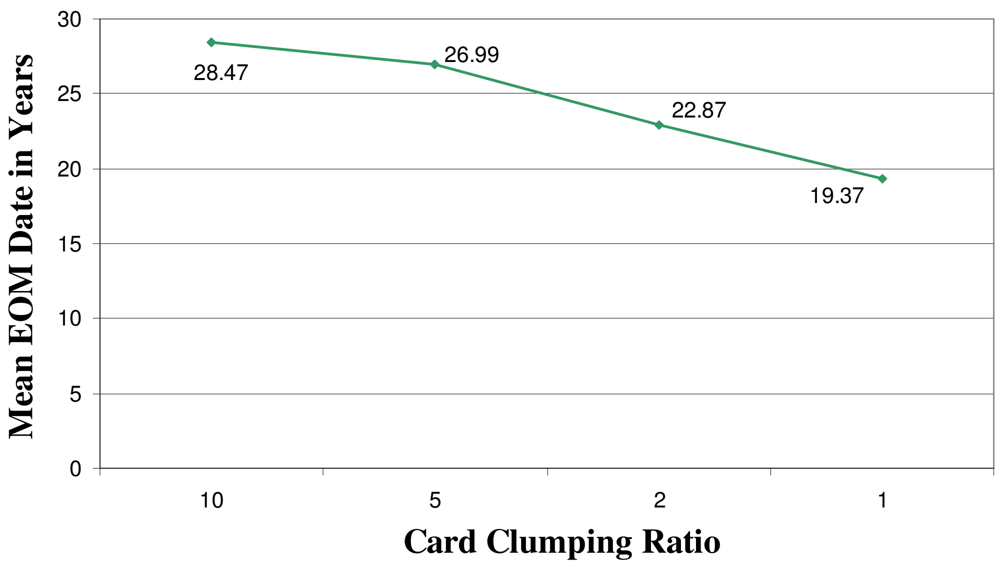



Fig. 2.10 Convergence of mean EOM date for case 1 with card clumping

Each data point represents the average EOM date after 1,000 system life histories. At a card clumping ratio of 10:1, the first EOM date occurs 28.47 calendar years after the analysis date. As the card clumping ratio decreases, the mean EOM date monotonically decreases and converges to 19.37 calendar years. Additionally, the part/card combinations that cause the first EOM (and their likelihoods) vary with the card clumping ratio. At a card clumping ratio of 10:1, there are three recorded part/card combinations that cause the first EOM. At a card clumping ratio of 1:1 (where all parts modeled independently), there are five recorded part/card combinations that cause the first EOM. One of the part/card combinations that causes the first EOM at the 10:1 card clumping ratio does not even cause the first EOM at the 1:1 card clumping ratio.

### *2.2 Discrete Event Modeling Process*

The discrete event simulation model process is shown in Fig. 2.11. The model starts with information regarding the electronic system to be evaluated. These inputs are in the form of Bills of Materials (BOMs) that contain various properties of unique card types that comprise the system. Each BOM contains unique parts (and part quantities) that appear on the card, the obsolescence status of the part, the inventories the card is allowed to access spare parts from, and the failure of the part and/or reliability information. The second piece of information that must be input into the model is the inventory information associated with the system. This includes segregated inventories and the parts and quantities of parts the respective inventories contain. The third required input is the fielded card information for the electronic system. The other pieces of information that must be included are simulation inputs and cost inputs.

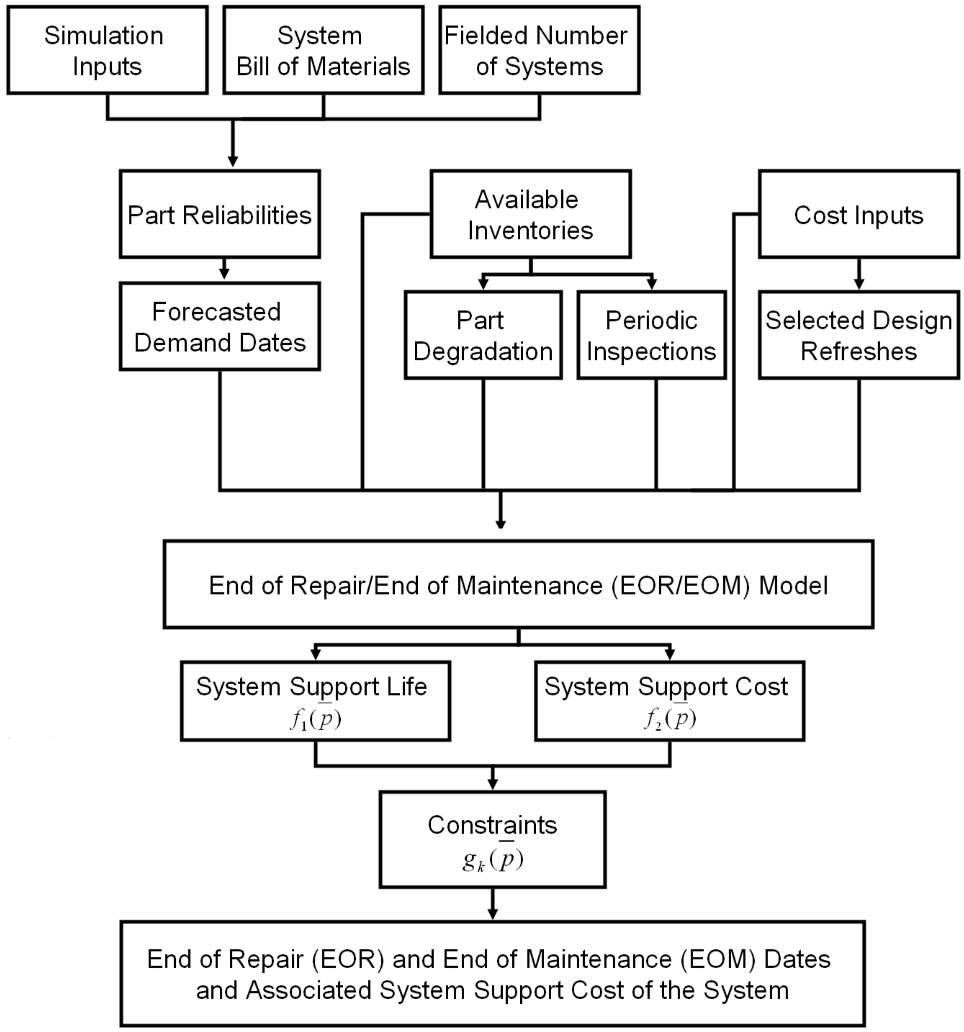



Fig. 2.11 Inputs and outputs of the EOR/EOM model

After all input information has been, the simulation runs a set of system life histories to capture system uncertainties and the resulting outputs can then be expressed in terms of probability distributions. The output information contains the parts and cards and ordering of the EOR and EOM events within the life histories that can be used later for statistical analysis.

*2.3 System Operational Profile*

The operational profile of the system affects the entire simulation and how it is analyzed. The system operational profile is expressed in terms of *operational hours per year* where 8,760 operational hours per year means that the system is operational 24 hours a day, 7 days a week, 365 days a year. The operational profile affects the frequency at which discrete events occur in simulation time as events occur based on operational hours, calendar hours, or in cycles per operational year.

*2.4 Evaluation of Subsequent EOR and EOM Dates*

Oftentimes for mission-critical systems, the primary concern of the system supporter is tracking the system until the first EOM event occurs. However, system supporters are also interested in how the system will function or behave after the first EOM event has occurred (i.e., supporters want to know the rate at which system instances become unsupported). The addition to the functionality of the model can easily be implemented by altering the constraints used in equations (1.1) and (1.2). Therefore, the model can be extended to identify possible consecutive subsequent EOMs that may occur during system operation.

It is venturing past the first EOM event where we introduce a new characteristic of the system known as *unsupportability*. The first moment of unsupportability in the system is when the first EOM event takes place whereby a request to replace a failed part cannot be fulfilled. When this type of event takes place, the card containing the failed part is deemed unsupportable and is removed from service. Unsupportability is a property for each type of card in the system that quantitatively measures how many instances of a given card type are fully functional (or remain supportable) over time.

The model evaluates consecutive EOR and EOM events based on two separate points of termination:

*1) Run simulation to the first fully unsupportable card*

*2) Run simulation to a specific End of Support (EOS) date*

The first evaluation tracks consecutive EOR and EOM events until the first instant whereby the total population (all instances) of a given card type has been removed (and deemed unsupportable) from the system. The second evaluation tracks consecutive EOR and EOM events until a specified calendar date. If the event arises where a card becomes fully unsupportable (total population of cards taken out of service), that simply means there are no forecasted part demands left for that type of card, and it is removed from the analysis until the EOS date is reached.

### *2.4.1 System Support Loss and Support Loss Rate*

In the previous section we defined the term *unsupportability* and explained how the model evaluates consecutive EOR and EOM events within a simulated life history of the system. Therefore, the system support loss (cumulative unsupportability for cards in the system) can be quantified (as well as the rate of support loss) and observed as a function of time. System support loss is calculated by card type and is accumulated based on subsequent failed requests to meet part demands on cards in the system. The unsupportability (for a given type of card at any given time) is measured as the ratio of unsupportable cards over the number of cards introduced into the field at the beginning of the analysis given by,

$$U = \frac{C_U}{C_I} \tag{2.8}$$

where,

$C_U$ = number of cards deemed unsupportable

$C_I$ = number of cards introduced into the field at the beginning of the analysis.

The support loss rate (for a given type of card) can also be linearly estimated between any two fractions of unsupportability, $\Delta SL$ and is given by,

$$\Delta SL = \frac{U_2 - U_1}{C_2 - C_1} \tag{2.9}$$

where,

$C_{1,2}$ = Specific calendar dates for a given card where $C_2 > C_1$

$U_{1,2}$ = The cumulative unsupportability fraction evaluated at simulation time $C_{1,2}$ respectively.

Measures of unsupportability and system support loss are useful metrics for system supporters because they provide a representation of how the system will behave as a result of consecutive EOM events. Additionally, system supporters can also extrapolate the unsupportability of cards based on previously observed system support loss rates.

## *2.5 Determining EOR and EOM Information*

EOR and EOM events are chronologically recorded within every simulated life history of the system. The information associated with each of these events is also recorded for analysis after the simulation has ended. The EOR and EOM events are referred to as "ordered" events based on their chronological occurrences. For example, the first-ordered EOM event is synonymous with the first EOM event (in a given simulated life history) and so on. The calculated EOR/EOM information that is analyzed across the simulated life histories is based on their chronological "order" of occurrences.

The $i$th-ordered mean EOM time (organized by order of occurrence within a single life history) for a given part-card combination is given by,

$$\overline{M}_i = \sum_{j=1}^{s} \frac{M_{ij}}{N_{ij}} + D_A \tag{2.10}$$

where,

$\overline{M}_i$ = $i$th-ordered mean EOM time

$M_{ij}$ = $i$th-ordered EOM time in the $j$th life history

$N_{ij}$ = number of occurrences as an $i$th-ordered EOM in the $j$th life history - either 1 (occurs) or 0 (does not occur)

$s$ = number of life histories simulated.

The corresponding probability for the given part-card combination causing the $i$th-ordered EOM is given by,

$$P_i = \frac{\sum_{j=1}^{s} N_{ij}}{s} \tag{2.11}$$

where,

$P_i$ = $i$th-ordered EOM probability.

The mean EOR times for given part-card combinations and their associated probabilities are analyzed in the same manner (chronological ordering based on the last available repair action).

### *2.5.1 Card-Specific EOM Information*

The EOM event information can also be organized at the card-level (by card) rather than the system-level (order of occurrence). The associated means and probabilities can be generated to provide probability distributions of EOM dates on a card-basis rather than an ordering basis. The card-level EOM information tracked was organized for first-ordered events associated with each card. Therefore, the mean EOM time and corresponding probability for a given part-card combination concerning its first EOM event is given by equations (2.10) and (2.11) with $i$=1, respectively.

The system-level analysis partitions events by order of their occurrences, while the card-level analysis partitions first-ordered events by particular cards.

## *2.6 Inventory of Spare Cards and Throwaway*

The model draws from inventories of spare parts as forecasted part demands are requested, but what happens when these inventories of spare parts are depleted? Typically when a failed part cannot be replaced from its inventories of spare parts, a spare card is used to replace the existing card that the failed part is located on (along with all the non-failed parts on the existing card). To further extend system support life capabilities, the model includes inventories of spare cards that can be accessed once the part inventories are depleted. In the event that a part demand cannot be satisfied for a particular card that has available spare cards to draw from (see Fig. 2.12) the existing card is thrown away and replaced with one of its available spare cards. The actions of throwaway and replacement of an existing card means the existing card must be discarded and replaced— accounted for by the removal and resampling of its part demands from their corresponding failure distributions—spare cards are assumed to be "new".

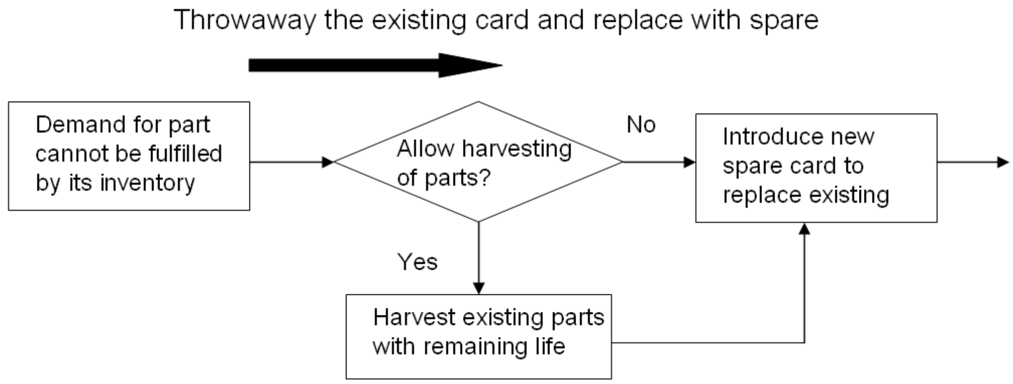



Fig. 2.12 Throwaway and part harvesting process

*2.6.1 Part Harvesting*

Another viable option is the harvesting or salvaging of parts off of the discarded cards (i.e., the obsolescence mitigation strategy commonly known as reclamation). The action of part harvesting removes parts off the discarded card that have not failed and places them in a separate inventory of harvested parts. When inventories of spare parts and spare cards are depleted, this third inventory (of harvested parts) is then accessed and drawn from until there are no more spares available— a process that extends the EOM date (i.e., system support life) of the system. Generally, the physical activity of harvesting or salvaging parts will damage (reduce life) from the part. The remaining fraction of useful life for the *i*th harvested part from the *j*th card, $L_{ij}$, is given by,

$$L_{ij} = H_i \frac{FD_i - t_H}{FD_i - t_i} \tag{2.12}$$

where,

$H_i$ = life fraction of *i*th part preserved from the physical action of harvesting (0-1)

$FD_i$ = *i*th part forecasted demand date

$t_H$ = simulation time when the harvesting activity occurs

$t_i$ = simulation time when the *i*th part was introduced into the system

The numerator in the fraction of equation (2.12) is the remaining part life represented as the difference between the forecasted demand date of the part and the simulation time when the harvesting activity occured. The denominator represents the forecasted part life when the part was first introduced.

The remaining part life must be preserved as a fraction rather than a time-to-failure because the harvested part may be used to repair a different type of card (not the same type of card the part was harvested from) where the part may have a different time-to-failure distribution. The remaining fraction of useful life is then used to adjust future forecasted part demands during part replacements when all other existing inventories are depleted.

The adjusted forecasted demand of the *i*th part from the *m*th card (*m* may equal *j*), $AFD_{im}$, is then given by,

$$AFD_{im} = L_{ij} FD_{im} \tag{2.13}$$

where,

$FD_{im}$ = *i*th part forecasted demand date from the *m*th card

The following section discusses the modeling of unusable card spares (i.e., degradation of spare cards).

## *2.7 Unusable Card Spares*

The next topic of discussion concerns the degradation of the inventories of spare cards. The model accounts for this effect by assigning a fraction of degraded spare cards per year. As the event occurs on a yearly basis, the degradation occurs on the first discrete event of each progressed year during the simulation. Each progressed year in the simulation can be found by evaluating the current simulation time and dividing by the operational profile of the system. For example, when the system is fully operational (8,760 operational hours per year), the first progressed year would occur at the next event that occurs either on or after 8,760 operational hours (one year of elapsed time for system operation). The number of spare cards in stock are then removed by a fraction of the existing stock (0-1) to account for a fraction of spare cards that have degraded and become inaccessible for possible maintenance activities. This degradation occurs periodically on the first event of every progressed year during simulation until spare card inventories have been depleted or the simulation has been terminated.

## *2.8 Design Refresh To Increase System Sustainment*

Technology or design refreshes are used in the replacement of one or more obsolete parts with non-obsolete parts in order to keep the system sustainable. In the EOR/EOM model, a completed design refresh for a particular type of card means that all obsolete parts from the entire population of cards (of the refreshed card type) have been replaced (and potentially harvested) with non-obsolete parts (i.e., all the part demands associated with a card type are removed).

The model treats design refreshes in two separate analyses:

*1) Selective design refreshes of cards and their refresh completion dates prior to analysis to EOM*

*2) Construction of a design refresh plan to ensure system sustainment to a specific date*

The first type of analysis, assuming the simulation progresses to the selected refresh completion date, implements design refreshes for selected cards until EOM is reached. The second analysis constructs a design refresh plan to ensure that the system is able to be sustained until a specific date. The assumption associated with design refresh planning (the second analysis) is that the planned design refreshes are completed on the dates when they are needed (i.e., the date the first EOM would have occurred for a given card). The analysis used in design refresh planning tracks and records the planned design refreshes and their planned completion dates. The results of design refresh planning are a list of the planned refreshes and their probability distributions of planned completion dates.

### 2.9 Implementation of System Support Costs

Not only are we interested in the event-driven methodology for calculating EOR and EOM events for electronic systems, but we also wish to assess the system support costs associated with system sustainment. The system support cost ($C_{sys}$) at any given time during the life history of the system is the total cost of the maintenance activity costs ($C_M$), inventory holding costs ($C_{IH}$), and infrastructure costs ($C_{Inf}$).

The system support cost can be calculated at any time (for a given base year of money $y_b$ and discount rate $R$) using,

$$C_{sys}(y_b, R) = C_M(y_b, R) + C_{IH}(y_b, R) + C_{Inf}(y_b, R) \qquad (2.14)$$

Prior to accumulation of the system support cost, each sub-cost is converted to its net present value (*NPV*) based on the current simulation time, base year of money, and discount rate. The *NPV* of a cost ($C_x$) at time $t$ (in years) is given by,

$$C_x(NPV) = \frac{C_x}{(1+R)^{t-y_b}} \qquad (2.15)$$

The system support cost is then accumulated after the NPV of each sub-cost is calculated.

The maintenance activity costs ($C_M$) include the costs associated with administrative actions ($C_A$), replacement ($C_R$), disposal ($C_D$), inspection ($C_I$), and cost per design refresh ($C_{DR}$) as shown,

$$C_M = C_A + C_R + C_D + C_I + C_{DR} \qquad (2.16)$$

Maintenance activity costs are accumulated based on the type of event that occurs. Administrative costs are accrued from any type of event that occurs in the simulated life history (e.g., part failures, degradation, inspection, design refresh). Replacement costs are accrued from corrective maintenance activities (replacing failed parts). Disposal costs are accrued whenever a part is disposed (e.g, part failures, card throwaway). Inspection costs are added for parts that are inspected within inventory. The non-recurring cost associated with a selected design refresh occurs on the date the design refresh is completed and implemented within the system. It is assumed that all design refreshes for all cards in the system cost the same amount.

The inventory holding costs include the cost of storing spare parts and cards over periods of time, these holding costs can be separated into part inventory costs ($C_{PI}$), harvested part inventory costs ($C_H$), and spare card inventory costs ($C_C$),

$$C_{IH} = C_{PI} + C_H + C_C \tag{2.17}$$

The inventory holding costs are accumulated as a result of time periods <u>between</u> discrete events and account for the time a certain quantity of items is held in inventory. The infrastructure cost is a periodic recurring cost that occurs every simulated calendar year.

Additionally, the system support cost can be accumulated and displayed as a cumulative total support cost over the simulated life history of the system. The resulting output cost information are probability distributions of the total system support costs and the computation of the average sub-costs (maintenance, inventory holding, and infrastructure) comprising the system support cost.

Chapter 3 describes simple test cases used to demonstrate the capabilities of the EOR/EOM model followed by a case study including an actual legacy electronic system.

# Chapter 3 : EOM Case Study

In order to demonstrate the model and exercise its capabilities, several simple example cases (drawing from same and separate inventories) were developed, followed by a case study using an actual legacy electronic system. The system used in the case study is comprised of unique cards, each card containing unique parts and historical part failure histories. The objective of the case study is twofold: 1) to demonstrate the capabilities of the EOR/EOM model and 2) to observe the legacy system sustainment and support cost ramifications through a composition of different scenarios (i.e., "what if" situations including part harvesting or immediate first failures for no-failure obsolete parts). The two simple example cases demonstrating the capability of the EOR/EOM are first presented in Sections 3.1 and 3.2.

## *3.1 Simple Example Case Drawing from the Same Inventory*

The following example case is comprised of two parts that appear on two different cards. There is only one instance of each part located on each card and one instance of each card within the system (i.e., there are four unique parts in the system). Both cards draw their part spares from the same inventories. The reliabilities for each unique part that comprise the system are shown in Table 3.1 (fixed values are assumed for the simple example cases). The number of spares associated with both types of parts are shown in Table 3.2. The electronic system is assumed to always be operational (8,760 hours per year), and it is assumed there is no degradation or inspection of the inventory.

Table 3.1 Part Time-to-Failures for Simple Example Cases

| | Sample Card 1 | Sample Card 2 |
|---|---|---|
| Part 1 | 350 operational hours | 100 operational hours |
| Part 2 | 100 operational hours | 275 operational hours |

Table 3.2 Number of Part Spares for Sample Example Case 1

| | Inventory 1 |
|---|---|
| Part 1 | 10 |
| Part 2 | 9 |

The simple test case is divided into three separate analyses (see Sections 3.1.1 through 3.1.3). The simulation of each analysis is terminated with the occurrence of the first EOM event for the system. The first analysis examines the simple case using the inventory of part spares from Table 3.2. The second analysis introduces an inventory of spare cards that are used when part spares have been depleted. The third analysis introduces the implementation of part harvesting with the previous analysis (described in Section 3.1.2).

### *3.1.1 Simple Example Case Results Without Spare Cards (Same Inventory)*

The part spares depletion for this example case is shown in Fig. 3.1. The results from the case show that the system is capable of being supported for 800 operational hours. The first EOM event occurs for Part 2 from Sample Card 1 (based on the observed forecasted demands), and first EOR event occurs for Part 2 from Inventory 1 at 700 operational hours.

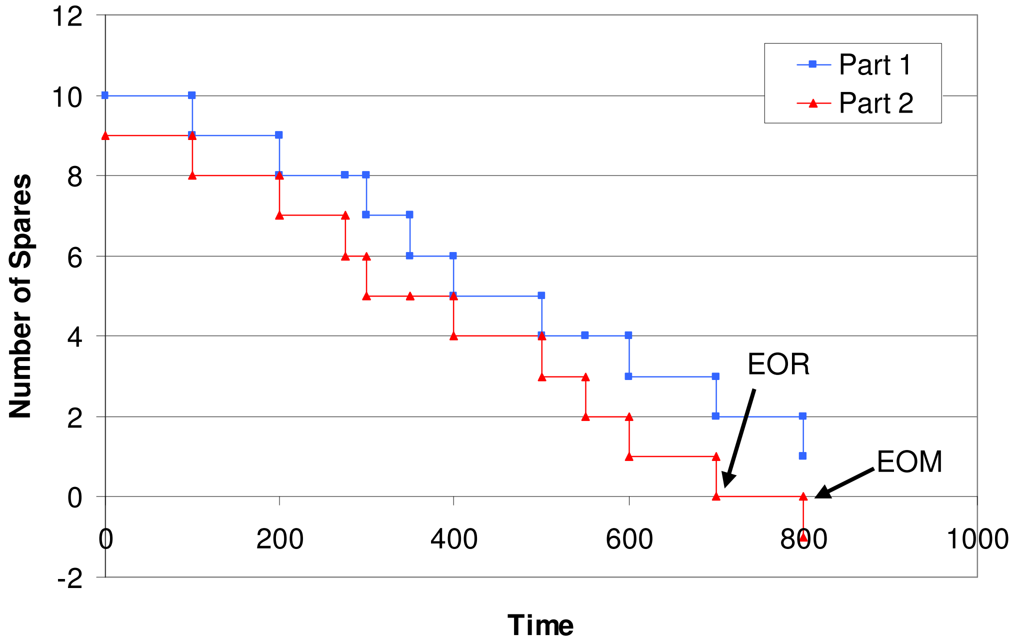


Fig. 3.1 Part spares depletion for example in Section 3.1.1

*3.1.2 Simple Example Case Results Including Spare Cards (Same Inventory)*

The second analysis introduces an inventory of spare cards for each card within the system. There are six spare cards for replacing Sample Card 1 and three spare cards for replacing Sample Card 2; these spare cards replace the existing card when a spare part cannot be located to replace its failed counterpart. The part spares depletion for this example case is shown in Fig. 3.2. The results from the sample case show that the system is capable of being supported for 1125 operational hours. The first EOM event occurs for Part 1 from Sample Card 2 (based on the observed forecasted demands). The two EOR events that occur are by Part 2 from Inventory 1 and by Part 1 from a spare card inventory at 700 and 1025 operational hours, respectively.

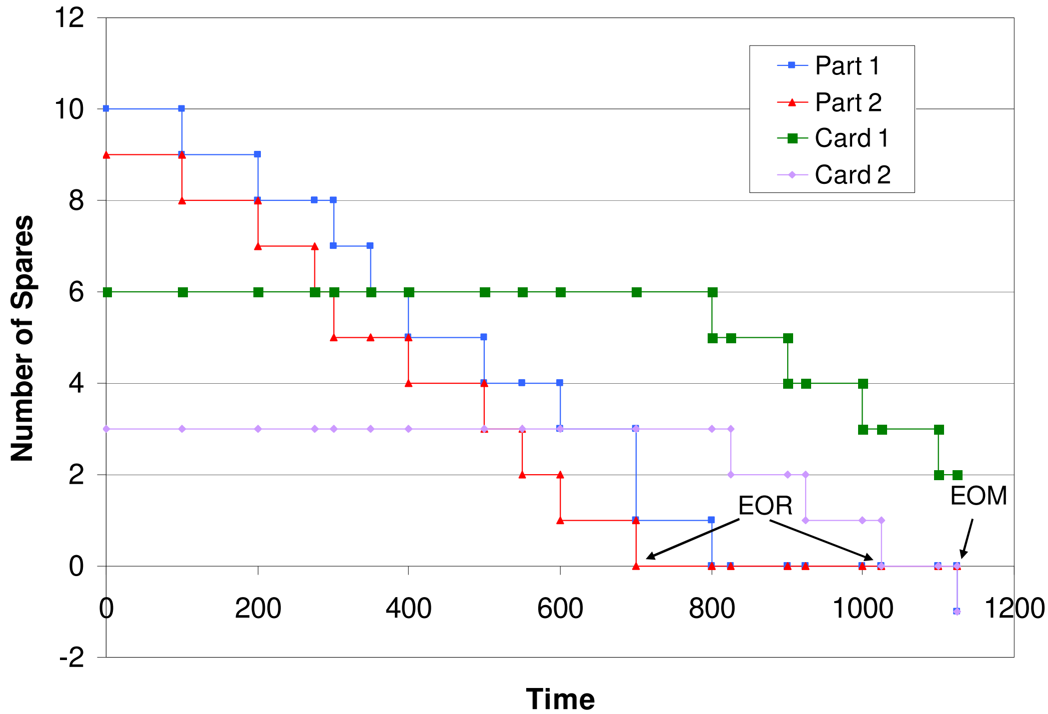


Fig. 3.2 Part spares depletion for example in Section 3.1.2

*3.1.3 Simple Example Case Results Including Spare Cards and Part Harvesting (Same Inventory)*

The third analysis introduces the action of harvesting parts during card replacements. The remaining life of the harvested part is preserved, and the harvested part is used towards replacements after the inventory of spare cards have been depleted. The part spares depletion for this example case is shown in Fig. 3.3. The results from the sample case show that the system is capable of being supported for 1463 operational hours. The first EOM event occurs for Part 2 from Sample Card 1. The EOR event that occurs is by Part 2 from its harvested inventory at 1400 operational hours. The analysis is the same as Section 3.1.2; however, parts are harvested from cards that are replaced instead of thrown away.

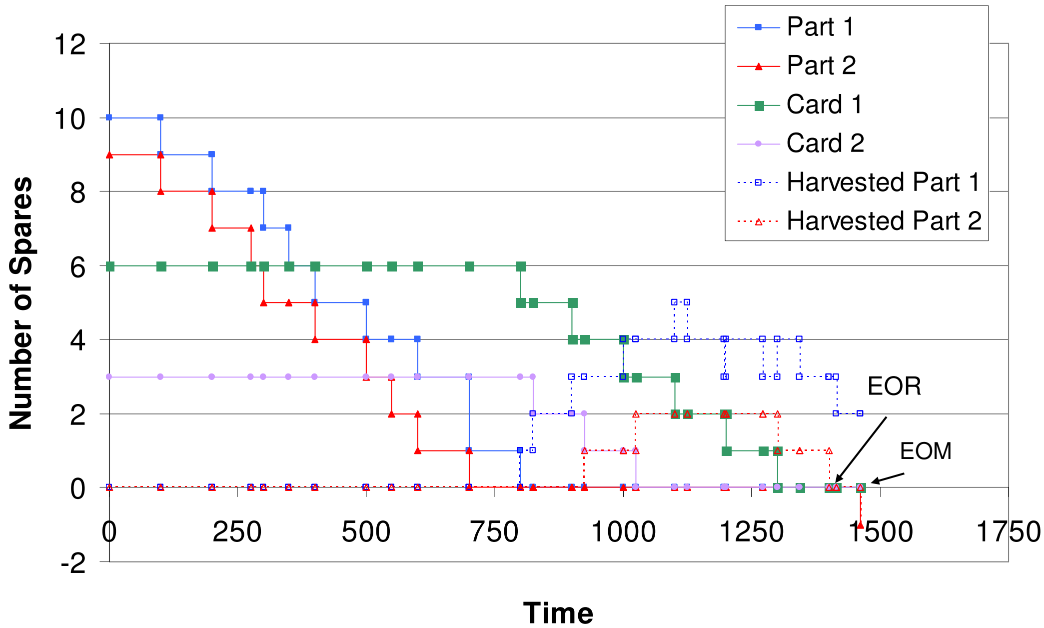



Fig. 3.3 Part spares depletion for example in Section 3.1.3

### *3.2 Simple Example Case Drawing from Separate Inventories*

The following example case is comprised of two parts that appear on two different cards. There is only one instance of each part located on each card and one instance of each card within the system (i.e., there are four unique parts in the system). Sample Card 1 draws its part spares from an inventory labeled "Inventory 1" and Sample Card 2 draws its part spares from an inventory labeled "Inventory 2". The reliabilities for each unique part are shown in Table 3.1. The number of spares associated with both types of parts are shown in Table 3.3. The electronic system is assumed to always be operational (8760 hours per year) and it is assumed there is no degradation or inspection of the inventory.

Table 3.3 Number of Part Spares for Simple Example Case 2

| | Inventory 1 | Inventory 2 |
|---|---|---|
| Part 1 | 6 | 4 |
| Part 2 | 4 | 5 |

The simple test case is divided into three separate analyses (see Sections 3.2.1 through 3.2.3). The simulation of each analysis is terminated with the occurrence of the first EOM event for the system. The first analysis examines the simple case using the inventory of part spares from Table 3.3. The second analysis introduces an inventory of spare cards that are used when part spares have been depleted. The third analysis introduces the implementation of part harvesting with the previous analysis (described in Section 3.2.2).

### *3.2.1 Simple Test Case Results Without Spare Cards (Separate Inventories)*

The part spares depletion for this example case is shown in Fig. 3.4. The results from the sample case show that the system is capable of being supported for 500 operational hours. The first EOM event occurs for Part 2 from Sample Card 1 and the two EOR events (Part 2 from Inventory 1 and Part 1 from Inventory 2) that occur at 400 operational hours. The part spares depletion for this example case is shown in Fig. 3.4. It is also observed that Part 1 from Sample Card 2 has an EOM event at the same time as Part 2 from Sample Card 1.

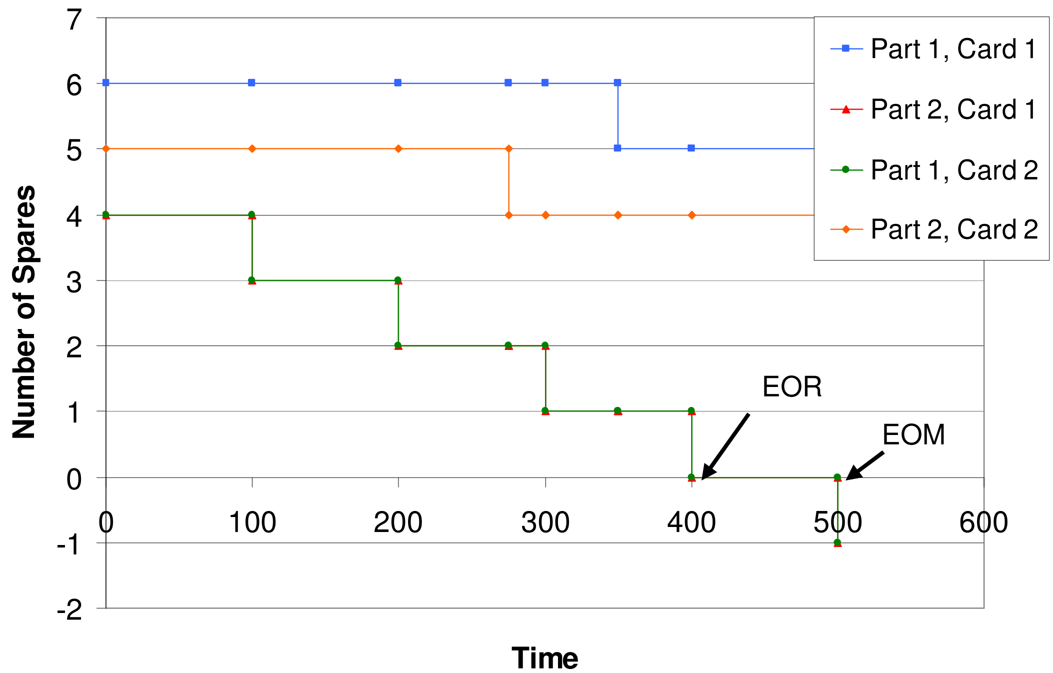



Fig. 3.4 Part spares depletion for example in Section 3.2.1

*3.2.2 Simple Test Case Results Including Spare Cards (Separate Inventories)*

The second analysis introduces an inventory of spare cards for each card within the system. There are six spare cards for replacing Sample Card 1 and three spare cards for replacing Sample Card 2; these spare cards replace the existing card when a spare part cannot be located to replace its failed counterpart. The part spares depletion for this example case is shown in Fig. 3.5. The results from the sample case show that the system is capable of being supported for 800 operational hours. The first EOM event occurs for Part 1 from Sample Card 2. The two EOR events that occur are by Part 2 from Inventory 1 and by Part 1 from a spare card inventory at 400 and 800 operational hours, respectively.

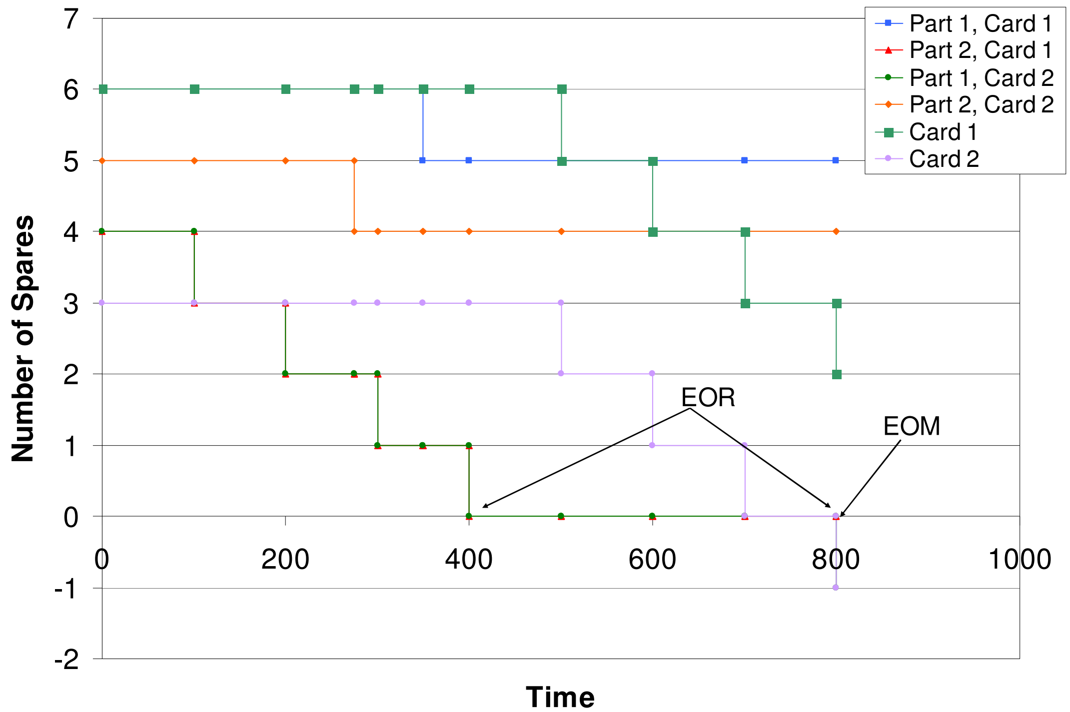


Fig. 3.5 Part spares depletion for example in Section 3.2.2

*3.2.3 Simple Test Case Results Including Spare Cards and Part Harvesting (Separate Inventories)*

The third analysis introduces the action of harvesting parts during card replacements. The remaining life of the harvested part is preserved and the harvested part is used towards replacement after the inventory of spare cards have been depleted. The part spares depletion for this example case is shown in Fig. 3.6. The results from the sample case show that the system is capable of being supported for 1,214 operational hours. The first EOM event occurs for Part 1 from Sample Card 2. The two EOR events that occur are by Part 1 from its harvested inventory and Part 2 from its harvested inventory at 1,142 and 1,182 operational hours, respectively. The analysis is the same as Section 3.2.2; however, parts are harvested from cards that are replaced instead of thrown away.

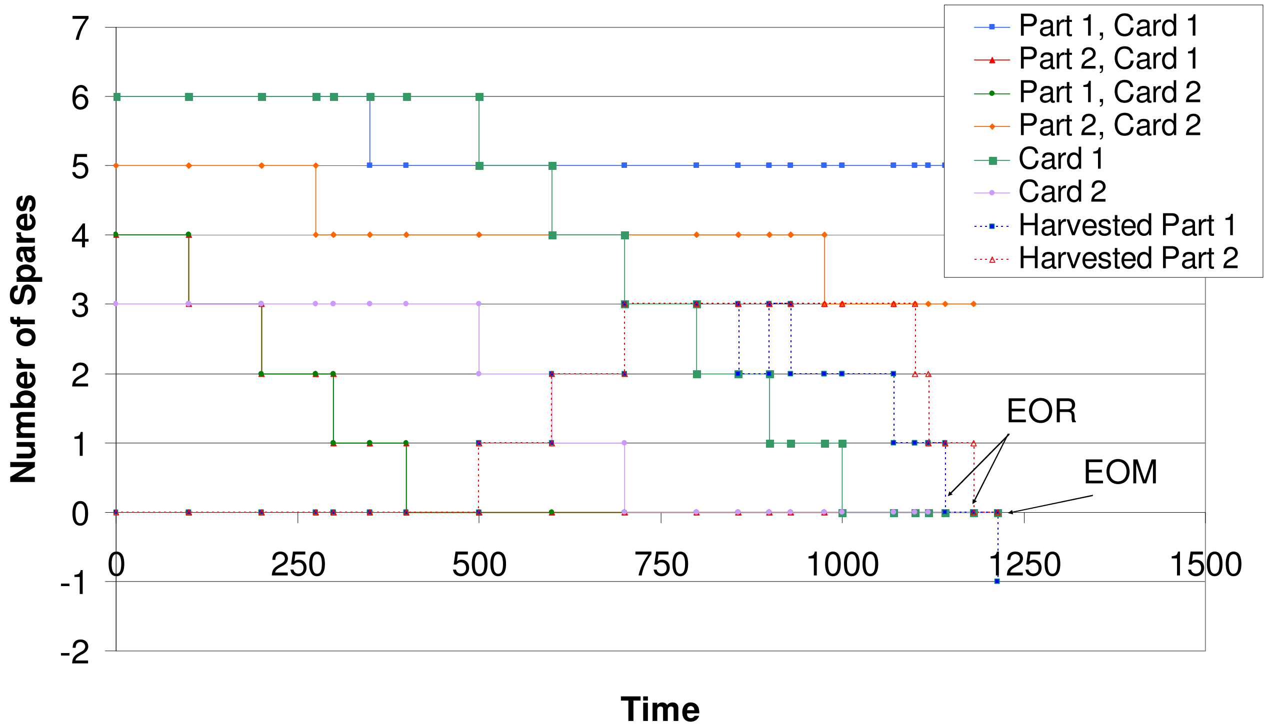



Fig. 3.6 Part spares depletion for example in Section 3.2.3

## *3.3 Case Study Description*

The previous section presented a very simple test case to demonstrate the basic operation of the model. This section presents a case study for a real system.

The legacy system under investigation contains 117,000 instances of 70 different cards totaling 4.5 million unique obsolete parts. Each card has a unique number of fielded units and a number of available spare cards to draw from. The provided legacy system was introduced in 1993 and the simulated analysis begins on January 1, 2011. The legacy system is tracked for 1,000 simulated system life histories for each test-case scenario in order to construct probability distributions of the EOM dates and observed support costs.

The case study assumes that all obsolete parts[10] from each card are modeled independently from their part failure distributions and are based on the observed historical failure data of the part. Furthermore, the entire system is assumed to have been fielded on the same date, and no additional instances of the system are manufactured or fielded at later dates. This may not always be the case as certain instances of the system may be requested or fielded earlier than others or system requirements or specifications may change resulting in requested alterations to the system. It is also possible that some instances of the system could be retired at specific dates before the EOM is reached. The system is assumed to remain fully operational throughout each year (8,760 hours per year) and is included in the analysis.

This case study assumes that there are no periodic inventory inspections, no degradation, and no infrastructure costs. The cost inputs for the case study (discussed in Sections 3.4.1 through 3.4.3) can be seen in Table 3.4.

Table 3.4 Cost Analysis Inputs for Case Study

| | |
|---|---|
| Administrative cost of a draw, $C_A$ (\$ per draw) | 1.5 |
| Replacement cost of a draw, $C_R$ (\$ per draw) | 5 |
| Part inventory cost, $C_{PI}$ (\$ per part per year) | 5 |
| Harvest inventory cost, $C_H$ (\$ per harvested part per year) | 25 |
| Card inventory cost, $C_C$ (\$ per card per year) | 20 |
| Unusable part disposal cost, $C_D$ (\$ per part) | 0.5 |
| Cost per refresh, $C_{DR}$ (\$) | 1,500,000 |
| Discount rate, $R$ | 3% |
| Base year for money, $y_b$ | 2011 |

[10] This statement excludes obsolete parts with no observed failure history and obsolete parts that have significant (hundreds or more) observed failures with right censoring (i.e., some part failures have not occurred and are unknown).

It is assumed that these "per-action" support costs and the discount rate and base year are treated as constants (no uncertainty) and they do not change values with respect to time or any other parameter.

*3.3.1 Solution Convergence*

How many life histories need to be considered in order for the solution for the population to be accurately represented? Typically, the EOR/EOM model tracks EOR and EOM dates for electronic systems for 1,000 life histories. Is 1000 life histories enough? Alternatively, due to the complexity of the case study system (4.5 million unique parts modeled), the analysis takes a considerable amount of time to perform, we do not wish to run more life histories than necessary. Therefore, we wish to find the number of simulations where the analysis converges to an EOM result that is accurate enough.

Figure 3.7 shows the average predicted EOM date as a function of the number of life histories included. The case study system appears to converge to an approximate steady-state solution around 250 simulations as seen in Fig. 3.7 (and retain all of its possible EOR and EOM part-card combinations).

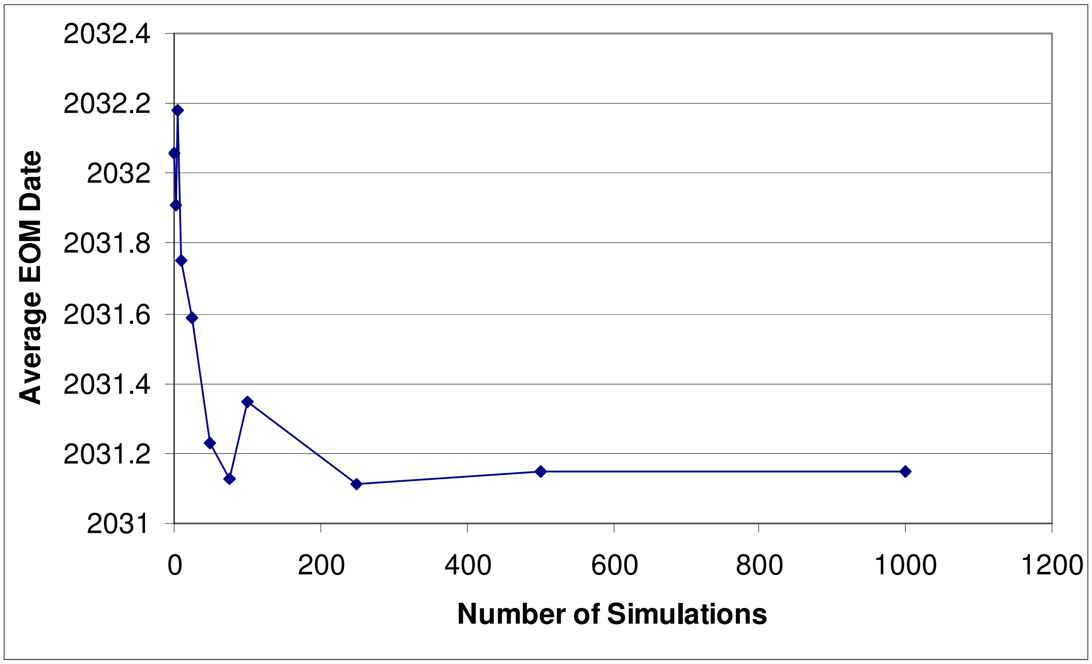



Fig. 3.7 Convergence of average EOM date for case 1

Additionally, there are well-define stopping criteria for Monte Carlo analyses based on the allowed standard error for the mean. The standard error of the mean is given by,

$$p = \left(\frac{z_{\alpha/2}\sigma}{e}\right)^2 \qquad (3.1)$$

where,

$\sigma$ = population standard deviation

$\mu$ = population mean

$p$ = sample size

$z_{\alpha/2}$ = z-statistic for two-tailed level of confidence

$e$ = standard error of the mean.

If we allow for the stopping criterion for analysis to be when the standard error of the mean is less than 1% (e equals 0.01μ), a 95% level of confidence (α=0.05), and using our test case 1 results (see Section 3.4.1), the required sample size is only 181 life histories. Therefore, we can conclude that tracking the case study system for 500 life histories is acceptable for the allowed level of confidence and standard error of the mean for EOR and EOM analysis.

### *3.4 EOR and EOM Test Cases*

The legacy system was examined using five different management assumptions representing "worst-case" and "best-case" scenarios while incorporating the use of part harvesting towards system sustainment. The 'best-case' scenario assumes that parts with no previous observed failure history within the system never fail, and are not considered during EOR/EOM analysis. This assumption may be valid depending on the nature of the system and when the legacy system was introduced (i.e., elapsed time without observed failures). The "worst-case" scenario assumes that parts with no previous failure history experience their first failure immediately at the beginning of the analysis, and then their failure distributions are synthesized according to the single observed failure. Each test case was tracked for 500 system life histories (based on the converged solution presented in Section 3.3.1) to construct probability distributions of EOM dates. The analysis ignored parts that were deemed non-obsolete, and inventories of spare cards were included in all test cases and used before inventories of accumulated harvested parts were considered.

1. Best case- no harvesting
2. Worst case- no harvesting
3. Worst case- with harvesting

Run to the first EOM in the system

4. Worst case- no harvesting
5. Worst case- with harvesting

Run until every card has reached its first EOM (2050 max)

Fig. 3.8 Legacy system test cases

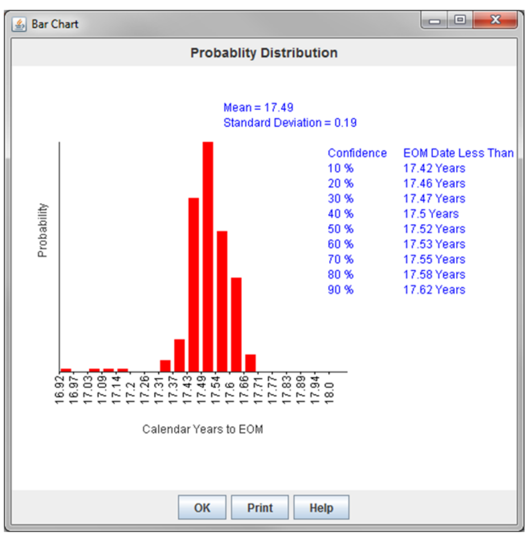
The first three test cases were analyzed to sustain the system until the first EOM date for the entire system (first instance that a part demand could not be fulfilled from available inventories) was reached. Test cases 4 and 5 sustain the system until one of two conditions was either met: 1) Run the simulation until every card type within the system has observed its first EOM date or 2) Run the simulation until the year 2050 has been reached. In both test cases 4 and 5, the first condition was never met so the simulation ran to 2050 and recorded the EOM events until that time. Test cases 4 and 5 were also ordered to organize EOM events and calculate associated means and probabilities on a card-level rather than system-level. This means that probability distributions of EOM dates were analyzed by individual cards rather than as a representation of the entire legacy system (by order of occurrence).

<u>*3.4.1 No Failure of Non-Failed Parts and No Harvesting (Test Case 1) Results*</u>

The results for the first test case can be seen in Figs. 3.9 and 3.10. The mean time to the first EOM date for the system was approximately 17.5 years (2028.5). The left side of Fig. 3.9 shows a distribution of the first EOM dates for the legacy system. On the basis of running 500 system life histories, the following statement conclusions can be drawn:

- 50% probability that at least one instance of the system will be unsupportable by 2028
- 95.4% probability of all instances of the system being supportable to 2028
- 100% probability that at least one instance of the system will be unsupportable by 2029

The right side of Fig. 3.9 shows the most probable causes of EOR/EOM events. The part that is most likely to result in the first-ordered EOM is part 6763-24 from Card 63 (81.6%) with a mean EOM time of 17.5 calendar years. This probability demonstrates that 408 out of 500 life histories, the 6763-24 parts from Card 63 caused the first EOM in the system. The system support costs are shown in Fig. 3.10.




| Part ID | Card ID | Mean EOM Time | Probability |
|---|---|---|---|
| 6763-24 | Card 63 | 2028.528 | 81.6% |
| 5004-02 | Card 61 | 2028.496 | 11.6% |
| 4000-44 | Card 30 | 2027.82 | 3.6% |
| 6006-51 | Card 41 | 2028.12 | 3.2% |

| Part ID | Repair Action | Mean EOR Time | Probability |
|---|---|---|---|
| 4000-44 | RPB Inventory | 2025.311 | 66.4% |
| 6763-24 | RPB Inventory | 2020.11 | 16.0% |
| 6006-51 | RPB Inventory | 2025.719 | 8.4% |
| 1788-63 | RPB Inventory | 2024.618 | 4.8% |
| 4000-44 | Card Stock | 2027.799 | 2.8% |
| 6006-51 | Card Stock | 2027.744 | 1.2% |

Fig. 3.9 System-level EOM distribution (left), EOM results (top right), and EOR results (bottom right) for case 1

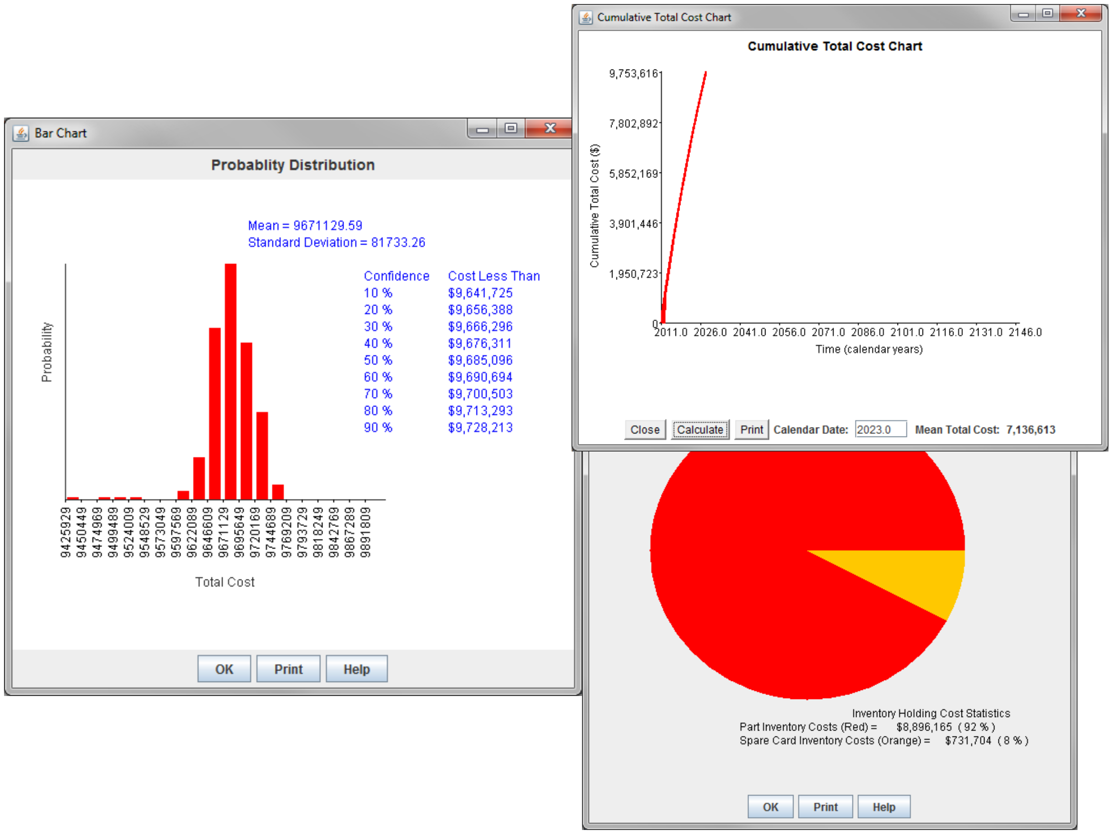
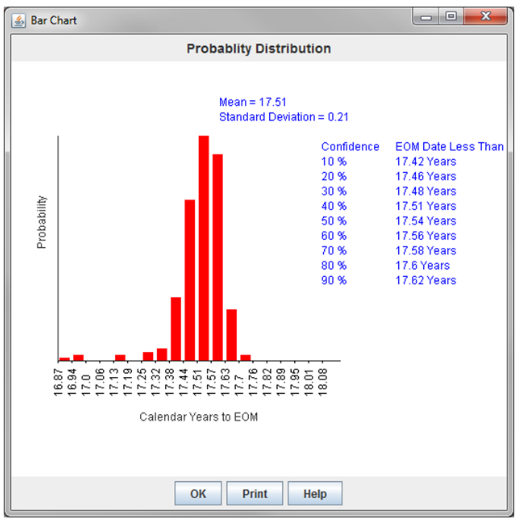



Fig. 3.10 System support cost distribution (left), cumulative total cost (top right), and inventory holding costs (bottom right) for case 1

The system support costs totaled, on average, $9.6 million for supporting the 117,000 instances of 70 different cards for 17.5 years. The system support cost could cost as little as $9.4 million, or as much as $9.8 million. The cumulative total costs (top right) for each life history are shown in Fig. 3.10, where the highest total cost observed was approximately $9.8 million. The inventory of spare parts accounts for 92 percent of the inventory holding costs, while the inventories of spare cards only account for the remaining 8 percent.

*3.4.2 Immediate First Failure of Non-Failed Parts and No Harvesting (Test Case 2) Results*

The results for the second test case can be seen in Figs. 3.11 and 3.12. The mean time to the first EOM date for the system was approximately 17.5 years (2028.5). The left figure shows a distribution of the first EOM dates for the legacy system. The right side of Fig. 3.11 shows the tabulated results of the six most probable causes of EOR/EOM events in the system.




| Part ID | Card ID | Mean EOM Time | Probability |
|---|---|---|---|
| 6763-24 | Card 63 | 2028.539 | 82.8% |
| 5004-02 | Card 61 | 2028.522 | 12.4% |
| 4000-44 | Card 30 | 2027.61 | 2.4% |
| 6006-51 | Card 41 | 2028.171 | 2.0% |
| 4000-44 | Card 57 | 2028.408 | 0.4% |

| Part ID | Repair Action | Mean EOR Time | Probability |
|---|---|---|---|
| 4000-44 | RPB Inventory | 2025.298 | 63.2% |
| 6763-24 | RPB Inventory | 2020.06 | 16.0% |
| 6006-51 | RPB Inventory | 2025.79 | 11.2% |
| 1788-63 | RPB Inventory | 2025.018 | 4.4% |
| 4000-44 | Card Stock | 2027.957 | 3.2% |
| 6006-51 | Card Stock | 2028.117 | 2.0% |

Fig. 3.11 System-level EOM distribution (left), EOM results (top right), and EOR results (bottom right) for case 2

The first EOM date does not decrease due to the "worst-case" assumption for obsolete parts with no failure histories. The same part-card combinations cause the EOM date to be reached.

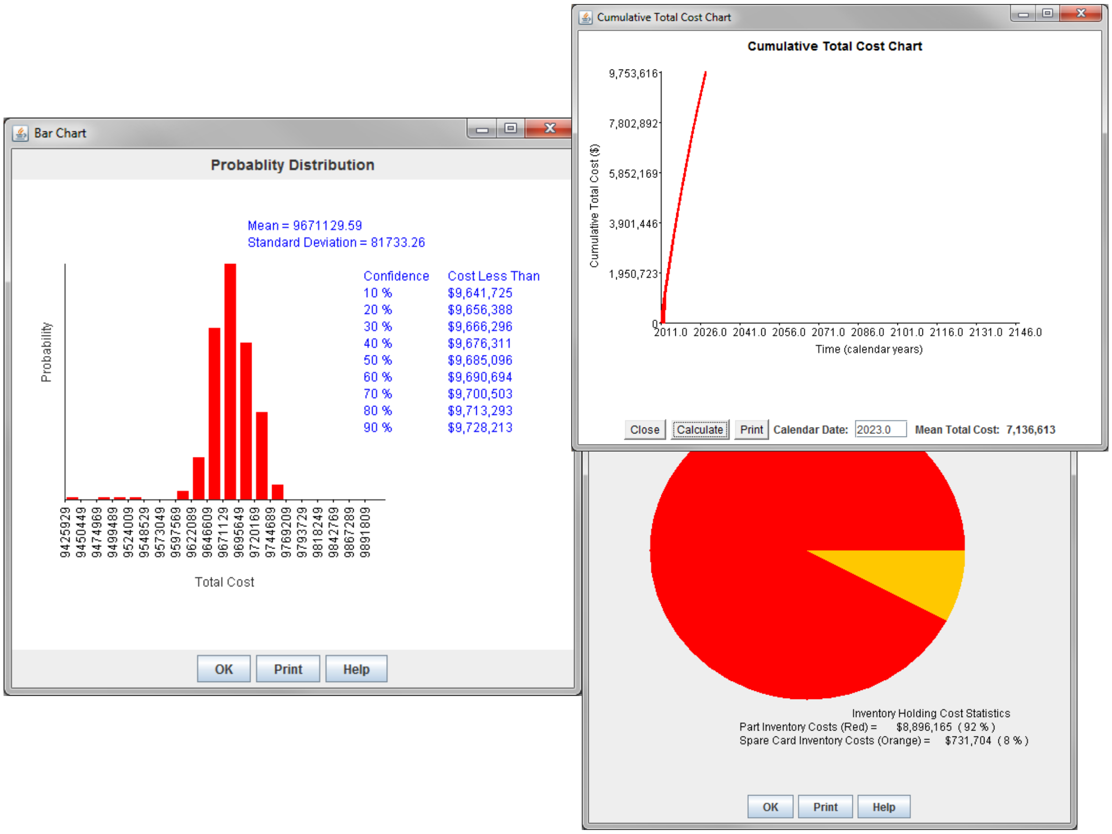
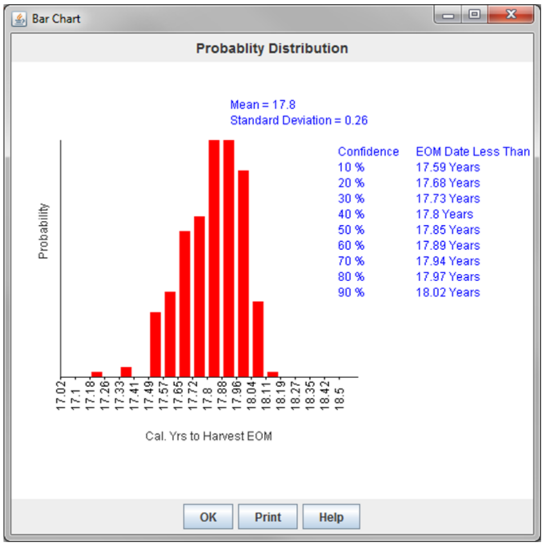



Fig. 3.12 System support cost distribution (left), cumulative total cost (top right), and inventory holding costs (bottom right) for case 2

The system support costs totaled, on average, $9.6 million for supporting the 117,000 instances of 70 different cards for 17.5 years. The system support cost could cost as little as $9.4 million, or as much as $9.8 million.

### *3.4.3 No Failure of Non-Failed Parts and Harvesting (Test Case 3) Results*

The results for the third test case can be seen in Figs. 3.13 and 3.14. The mean time to the first EOM date for the system was approximately 17.8 years (2028.7)—resulting in a quarter-year gain, on average, in system sustainment due to the action of harvesting of parts.




| Part ID | Card ID | Mean EOM Time | Probability |
|---|---|---|---|
| 5004-02 | Card 61 | 2028.795 | 78.8% |
| 6763-24 | Card 63 | 2029.013 | 16.8% |
| 6006-51 | Card 41 | 2028.115 | 4.4% |

| Part ID | Repair Action | Mean EOR Time | Probability |
|---|---|---|---|
| 4000-44 | RPB Inventory | 2025.33 | 65.6% |
| 6006-51 | RPB Inventory | 2026.246 | 18.4% |
| 6763-24 | RPB Inventory | 2025.072 | 7.6% |
| 1788-63 | RPB Inventory | 2025.062 | 4.8% |
| 4000-44 | Card Stock | 2027.654 | 1.6% |
| 6006-51 | Card Stock | 2028.115 | 1.2% |

Fig. 3.13 System-level EOM distribution (left), EOM results (top right), and EOR results (bottom right) for case 3

The same parts cause the first EOM event, even when part harvesting is implemented. One of the part-card combinations that caused the first EOM event in the previous cases is now delayed.

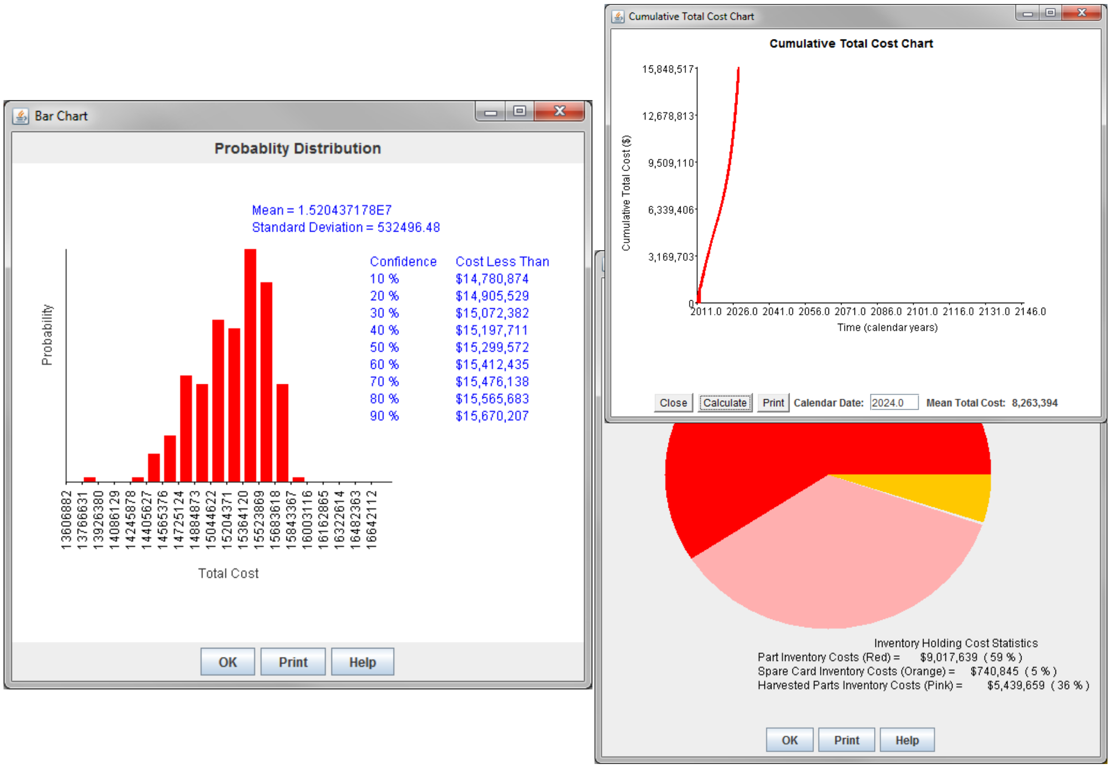



Fig. 3.14 System support cost distribution (left), cumulative total cost (top right), and inventory holding costs (bottom right) for case 3

The system support costs totaled, on average, $15.2 million for supporting the 117,000 instances of 70 different cards for 17.8 years. The system support cost could cost as little as $13.6 million, or as much as $15.8 million. The inventory of spare parts accounts for 59 percent, the inventories of spare cards account for 5 percent, and the inventory of harvested parts accounts for the remaining 36 percent of the inventory holding costs.

### *3.4.4 Immediate First Failure of Non-Failed Parts and No Harvesting (Test Case 4) Results*

The fourth test case initiates the change in analyses. The analysis ran until the year 2050, tracking all EOM events observed. The EOR/EOM model also can track specific cards through the system support life showing how the fielded number of cards is removed over time due to EOM events. Each of the tracked cards shown in Fig. 3.15 become fully unsupportable (all fielded cards are removed due to the failure of meeting part demands) by specific calendar dates. The card-level EOM results for the fourth test case can be seen in Fig. 3.16. The support loss rate of different cards can also be obtained based on the information displayed in Fig. 3.16. The support loss rate for Card 4 from its first EOM date to becoming fully unsupportable is 6.3 calendar years, an average of 15.7% unsupportable per calendar year.

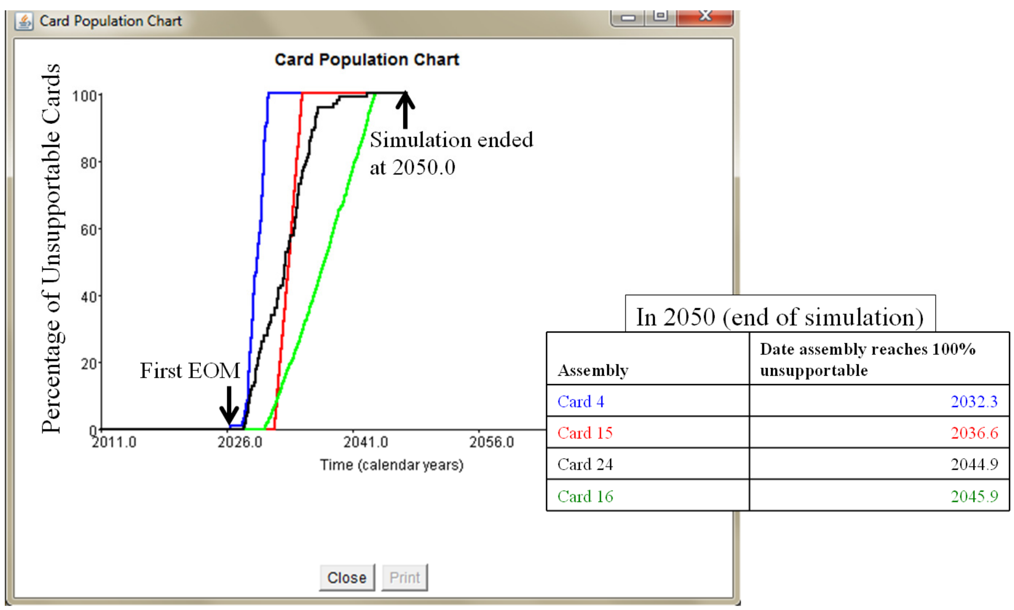
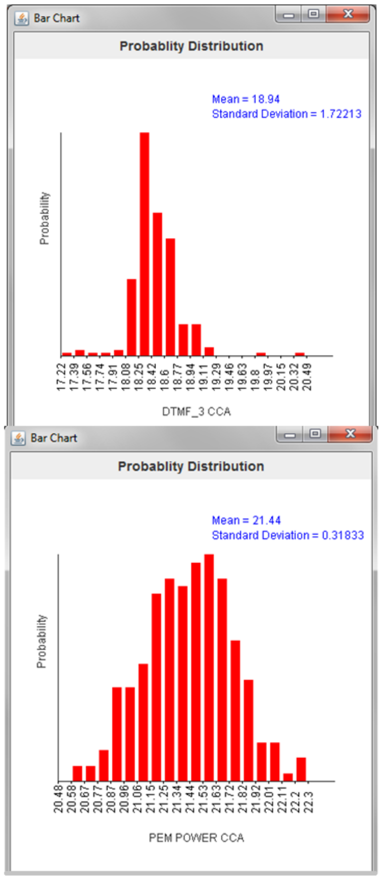


Fig. 3.15 Card-level support tracking and loss

The left figures in Fig. 3.16 show the card-level EOM probability distributions for specific cards in the legacy system. The table on the right side of Fig. 3.15 shows a list of the cards within the legacy system that observed at least one EOM event up until the calendar date (2050) when the simulation was terminated for a number of simulated life histories. It was shown that 41 of the 70 cards in the legacy system (22 shown in Fig. 3.16) exhibited first EOM dates prior to 2050, and probability distributions for each card that experienced EOM can be provided.



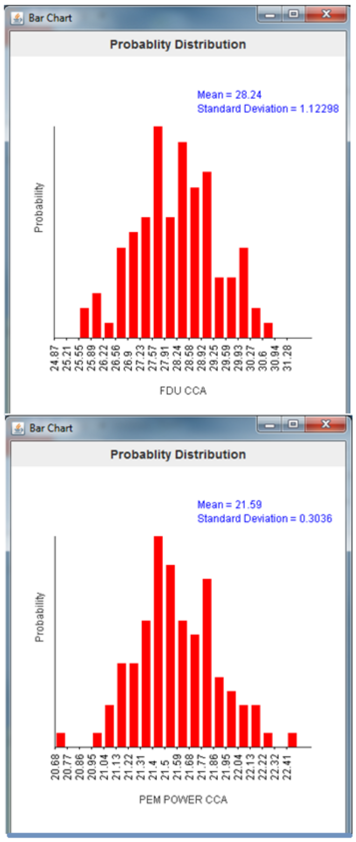

| Part ID | Card ID | Mean EOM Time | Probability |
|---|---|---|---|
| 3002-89 | Card 1 | 2038.93 | 100.0% |
| 2000-04 | Card 2 | 2033.677 | 100.0% |
| 8971-33 | Card 3 | 2040.943 | 100.0% |
| 3834-35 | Card 4 | 2029.941 | 100.0% |
| 1313-88 | Card 5 | 2033.624 | 100.0% |
| 0048-02 | Card 6 | 2034.917 | 100.0% |
| 5025-07 | Card 7 | 2037.781 | 48.8% |
| 7979-66 | Card 8 | 2035.702 | 100.0% |
| 3985-00 | Card 9 | 2039.394 | 100.0% |
| 8282-04 | Card 10 | 2035.365 | 100.0% |
| 5362-13 | Card 11 | 2029.686 | 40.8% |
| 9398-55 | Card 12 | 2048.817 | 86.4% |
| 1723-00 | Card 13 | 2034.625 | 100.0% |
| 6347-27 | Card 14 | 2048.333 | 78.4% |
| 6763-24 | Card 15 | 2033.515 | 100.0% |
| 5004-02 | Card 16 | 2032.438 | 100.0% |
| 5322-78 | Card 17 | 2033.583 | 100.0% |
| 5890-74 | Card 18 | 2030.658 | 100.0% |
| 6006-51 | Card 22 | 2034.376 | 100.0% |
| 3118-66 | Card 23 | 2035.823 | 100.0% |
| 4000-44 | Card 24 | 2028.473 | 100.0% |
| 4000-60 | Card 25 | 2037.753 | 100.0% |

Fig. 3.16 Card-level EOM distributions (left) and EOM results for case 4 (right)

### *3.4.5 Immediate First Failure of Non-Failed Parts and Harvesting (Test Case 5) Results*

The results for the last test case can be seen in Fig. 3.17, the only difference between cases 4 and 5 being the inclusion of part harvesting. The case 5 results differ from case 4 in that 22 of the 70 cards in the legacy system exhibited first EOM dates prior to 2050. The implementation of part harvesting delayed the EOM dates for 19 cards past the year 2050.

The common result is that part harvesting allows for card-level EOM dates to be delayed for significant periods of time. This result may not always be the case and depends on many different factors including the parts' failure distributions and whether critical parts that cause card-level EOM events appear on multiple cards within the legacy system. In addition, the action of harvesting parts may not significantly delay card-level EOM dates when faced with high-failure parts (due to excessive number of demands at a given time and lack of supply).




| Part ID | Card ID | Mean EOM Time | Probability |
|---|---|---|---|
| 3687-32 | Card 1 | 2040.258 | 100.0% |
| 2000-04 | Card 2 | 2034.572 | 100.0% |
| 5362-13 | Card 3 | 2042.956 | 100.0% |
| 6006-51 | Card 4 | 2041.642 | 95.0% |
| 1628-68 | Card 5 | 2035.631 | 100.0% |
| 1438-85 | Card 6 | 2039.244 | 100.0% |
| 4022-14 | Card 7 | 2042.208 | 95.0% |
| 5503-89 | Card 8 | 2038.164 | 100.0% |
| 3985-00 | Card 9 | 2042.335 | 95.0% |
| 9074-13 | Card 10 | 2039.843 | 100.0% |
| 6262-95 | Card 11 | 2043.962 | 100.0% |
| 9398-55 | Card 12 | 2049.859 | 43.7% |
| 1723-00 | Card 13 | 2031.952 | 18.0% |
| 6347-27 | Card 14 | 2048.018 | 100.0% |
| 3798-05 | Card 15 | 2034.301 | 100.0% |
| 5004-02 | Card 16 | 2032.5587 | 100.0% |
| 1818-11 | Card 17 | 2034.437 | 100.0% |
| 2008-84 | Card 18 | 2038.71 | 83.0% |
| 6006-51 | Card 22 | 2036.528 | 100.0% |
| 8897-53 | Card 23 | 2034.627 | 40.3% |
| 4000-44 | Card 24 | 2028.78 | 98.7% |
| 4000-60 | Card 25 | 2047.68 | 52.6% |

Fig. 3.17 Card-level EOM distributions (left) and EOM results for case 5 (right)

### *3.5 Selective Design Refreshes To Maximize System Sustainment*

As previously mentioned in Chapter 1, a design refresh refers to the replacement of one or more obsolete parts with non-obsolete parts, in order to keep the system sustainable. In this section, a sensitivity analysis is conducted for the previous test cases 1-3 (see Sections 3.4.1 through 3.4.3) to determine the additional system support life gained from design refreshing individual cards in the system. The completion dates for the individual card refreshes for each case were determined from their respective EOM date probability distributions (see Section 3.4.1 through 3.4.3). The completion dates for the selective card refreshes are implemented to occur prior to the earliest EOM date observed for the respective test case (i.e., the earliest EOM date observed for case 1 was 2027.9, and the selective design refreshes for case 1 are assumed to be complete in 2027). This assumption implements design refreshes at the latest possible time—resulting in cheaper (cost of money) design refreshes, and the possibility of refreshing additional obsolete parts that may become obsolete over the support life of the system (see Section 5.2.4). The completion dates for the individual card refreshes for cases 1-3 are shown in Table 3.5.

The individual cards chosen for design refreshes for each test case were determined first by the identified cards that potentially caused the first EOM date for the system (for test cases 1 and 2). Additional cards were also chosen and tested individually when they were identified to cause the first EOM date for the system as a result of implemented individual card refreshes. Selective design refreshes for case 3 included testing all cards within the legacy system, as harvested parts can potentially be used to support other cards in the system. The results for each test case and their selected cards are seen in Figs. 3.18 through 3.20. The results shown include individual refreshes for which there was a statistical difference among the means between the individual card refresh first EOM dates and the first EOM date from each test case from Sections 3.4.1 through 3.4.3.

The two-sided hypothesis testing is given by,

$$H_0: \mu_{DR} = \mu_{TC} \tag{3.2}$$

$$H_1: \mu_{DR} \neq \mu_{TC} \tag{3.3}$$

where,

$H_0$ = null hypothesis

$H_1$ = alternate hypothesis

$\mu_{DR}$ = population mean for a selected design refresh

$\mu_{TC}$ = population mean for the associated test case (no design refresh).

The EOM date probability distributions were assumed to approximate normal distributions based on the Central Limit Theorem, which states as the sample size becomes larger, the population frequency distribution approximates a normal distribution. A confidence level of 95% ($\alpha$=0.05) was used for each investigated case, and the selected refreshes that were shown to be statistically different (able to reject $H_0$) are shown in Figs. 3.18 through 3.20.

Table 3.5 Completion Dates for Implemented Individual Card Refreshes

| Case # | Card Refresh Completion Date |
|---|---|
| 1 | 2027 |
| 2 | 2027 |
| 3 | 2028 |

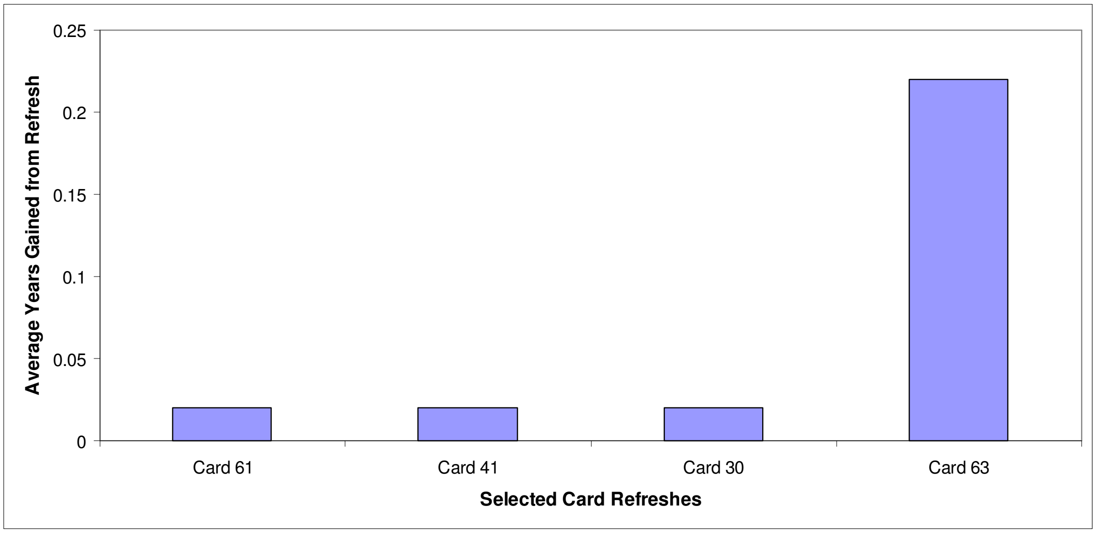


Fig. 3.18 System support life gained from individual card refresh, case 1

The results from case 1 show that there were four individual cards that had a statistically different mean system support life via design refresh compared to case 1. The best candidate for design refresh was Card 63 granting, on average, a system extended life of 0.22 years.

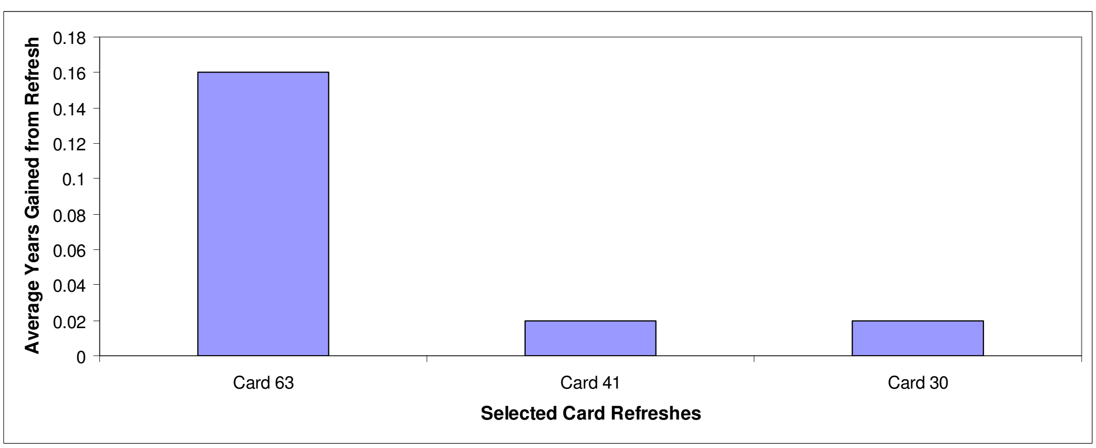


Fig. 3.19 System support life gained from individual card refresh, case 2

The results from case 2 show that there were three individual cards that had a statistically different mean system support life via design refresh compared to case 2.

The best candidate for design refresh was Card 63 granting, on average, a system extended life of 0.16 years.

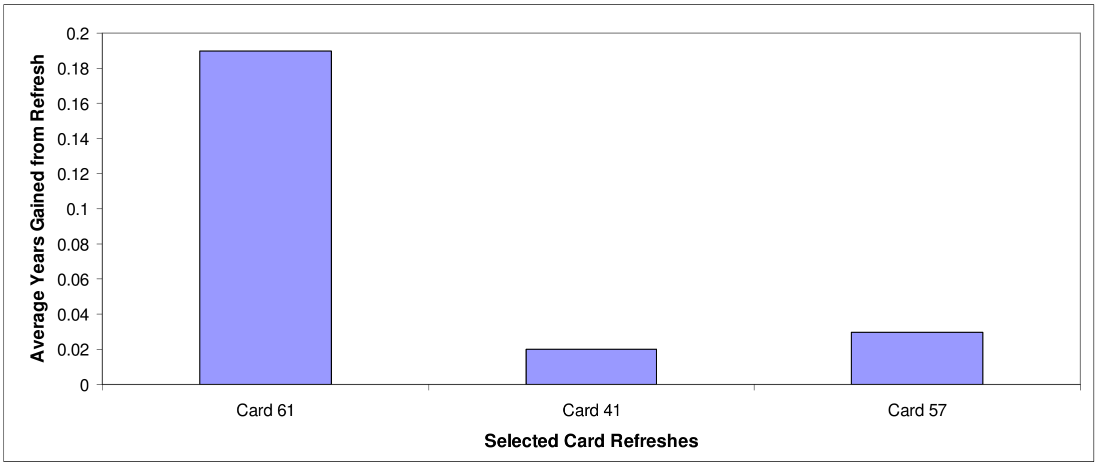



Fig. 3.20 System support life gained from individual card refresh, case 3

The results from case 3 show that there were three individual cards that had a statistically different mean system support life via design refresh compared to case 3. The best candidate for design refresh was Card 61 granting, on average, a system extended life of 0.19 years.

*<u>3.6 Design Refresh Planning For System Sustainment to End of Support</u>*

Another capability that may be concluded from the EOR/EOM model is the generation of a design refresh plan of selected cards to ensure system sustainment to a specific date. In this context, the EOR/EOM model assumes that a design refresh is completed for an individual card on the date that it is *necessary* (i.e., the first EOM date for that particular card) and records the completed date of the refresh and the identity of refreshed card. The following analyses assume a maximum of one design refresh per type of card—all the obsolete parts on the populations of the refreshed card are removed and excluded from the analysis for all times after the refresh is completed. The EOR/EOM model was used to determine design refresh plans to ensure system sustainment until the year 2050 for the first three test cases. The refreshes during design refresh planning are assumed to be completed "just-in-time" on the earliest EOM date associated with each type of card. This assumption implements design refreshes at the latest possible time—resulting in cheaper (cost of money) design refreshes, and allows for design refresh planning (design refresh as needed) rather than selectively entering design refreshes at specified dates (see Section 3.5)

The results from case 1 show that there were, on average, 10 individual cards generated in the design refresh plan to ensure system sustainment to the year 2050. The design refresh plan and a probability distribution of completion dates for individual refreshed cards are shown in Fig. 3.21.

| Card ID | Mean Refresh Time | Probability |
|---|---|---|
| Card 41 | 2033.572 | 100.0 |
| Card 63 | 2033.494 | 100.0 |
| Card 61 | 2032.452 | 100.0 |
| Card 39 | 2047.228 | 63.0 |
| Card 31 | 2044.837 | 75.4 |
| Card 32 | 2044.634 | 71.0 |
| Card 57 | 2036.629 | 100.0 |
| Card 29 | 2042.602 | 90.4 |
| Card 30 | 2034.271 | 99.8 |
| Card 40 | 2043.655 | 97.8 |

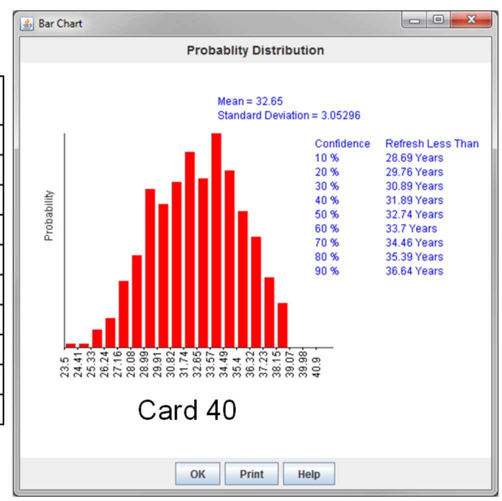



Fig. 3.21 Design refresh plan (left) and completed refresh date distribution (right), case 1

The results from case 2 show that there were, on average, 39 individual cards generated in the design refresh plan to ensure system sustainment to the year 2050. The design refresh plan is seen in Fig. 3.22. The introduction of the immediate failure of obsolete parts with no failure histories led to the increase in the failure of the additional cards prior to 2050, requiring them to be design refreshed. The number of EOMs (generated design refreshes)

| Card ID | Mean Refresh Time | Probability | Card ID | Mean Refresh Time | Probability |
|---|---|---|---|---|---|
| Card 4 | 2038.912 | 100.0 | Card 64 | 2048.25 | 81.4 |
| Card 59 | 2033.677 | 100.0 | Card 63 | 2033.489 | 100.0 |
| Card 14 | 2040.466 | 100.0 | Card 61 | 2032.457 | 100.0 |
| Card 18 | 2033.599 | 100.0 | Card 39 | 2030.75 | 100.0 |
| Card 50 | 2040.971 | 100.0 | Card 31 | 2030.381 | 100.0 |
| Card 12 | 2032.647 | 100.0 | Card 32 | 2030.402 | 100.0 |
| Card 41 | 2029.366 | 100.0 | Card 54 | 2033.627 | 100.0 |
| Card 22 | 2033.541 | 100.0 | Card 15 | 2037.023 | 100.0 |
| Card 21 | 2037.006 | 100.0 | Card 57 | 2030.038 | 100.0 |
| Card 42 | 2029.573 | 100.0 | Card 29 | 2030.146 | 100.0 |
| Card 48 | 2043.173 | 100.0 | Card 30 | 2029.469 | 100.0 |
| Card 3 | 2035.692 | 100.0 | Card 40 | 2030.255 | 100.0 |
| Card 25 | 2032.863 | 100.0 | Card 20 | 2048.438 | 72.6 |
| Card 24 | 2035.141 | 100.0 | Card 1 | 2042.682 | 100.0 |
| Card 49 | 2035.341 | 100.0 | Card 19 | 2033.759 | 100.0 |
| Card 13 | 2029.627 | 100.0 | Card 17 | 2034.596 | 100.0 |
| Card 55 | 2037.562 | 100.0 | Card 16 | 2034.445 | 100.0 |
| Card 62 | 2034.602 | 100.0 | Card 21 | 2039.545 | 100.0 |

Fig. 3.22 Design refresh plan, case 2

The results from case 3 show that there were, on average, 22 individual cards generated in the design refresh plan to ensure system sustainment to the year 2050. The design refresh plan and a probability distribution of completion dates for individual refreshed cards are shown in Fig. 3.23. The implementation of part harvesting reduced the required design refresh plan for case 2 by 17 cards.

| Card ID | Mean Refresh Time | Probability |
|---|---|---|
| Card 4 | 2043.182 | 100.0 |
| Card 59 | 2034.657 | 100.0 |
| Card 50 | 2041.634 | 100.0 |
| Card 12 | 2038.21 | 100.0 |
| Card 41 | 2033.915 | 100.0 |
| Card 23 | 2040.03 | 100.0 |
| Card 56 | 2041.305 | 100.0 |
| Card 6 | 2035.57 | 100.0 |
| Card 48 | 2041.04 | 100.0 |
| Card 3 | 2040.69 | 100.0 |
| Card 25 | 2045.361 | 100.0 |
| Card 49 | 2038.741 | 100.0 |
| Card 13 | 2037.087 | 100.0 |
| Card 55 | 2038.228 | 100.0 |
| Card 62 | 2034.939 | 100.0 |
| Card 63 | 2034.672 | 100.0 |
| Card 61 | 2032.624 | 100.0 |
| Card 54 | 2034.419 | 100.0 |
| Card 57 | 2036.236 | 100.0 |
| Card 7 | 2034.584 | 100.0 |
| Card 1 | 2044.713 | 100.0 |
| Card 16 | 2035.107 | 100.0 |

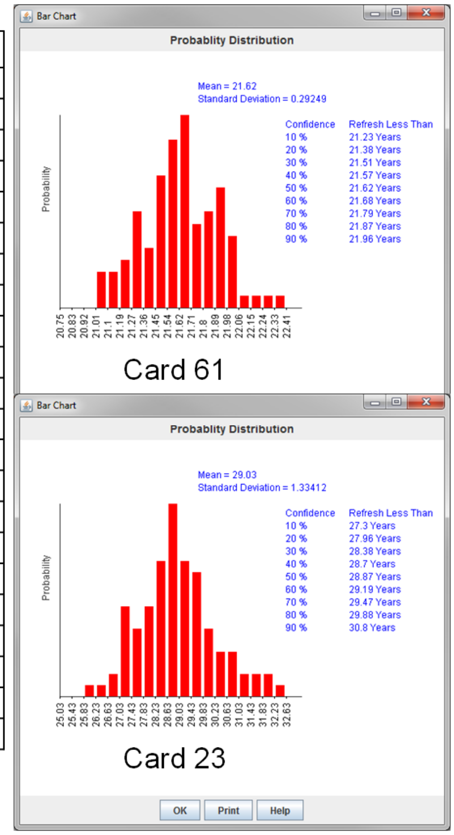



Fig. 3.23 Design refresh plan (left) and completed refresh date distributions (right), case 3

## *3.7 Summary of Case Study Results*

Case study scenarios were presented to demonstrate the methodology and capabilities of the EOR/EOM model. The test case scenarios included results for an actual legacy electronic system using the harvesting of parts, immediate first failure assumption for no-failure obsolete parts, and system sustainment to a specified End of Support date to track subsequent EOM events. An assessment of system support costs for each of the five presented test cases was also performed.

The model predicted that the electronic system used in the case study would be able to last, on average:

- 17.5 calendar years (best-case)
- 17.5 calendar years (worst-case)
- 17.8 calendar years (worst-case including part harvesting)

The immediate first failure assumption for no-failure obsolete parts did not reduce the system support life capabilities, and the implementation of part harvesting <u>extended</u> the system support life by approximately 0.33 years. Therefore, the activity of part harvesting for the case study system resulted in an extension of the overall support life of the system by approximately 2 percent.

The EOR/EOM model was then used to track subsequent EOM events in order to sustain the electronic system to the year of 2050. The model predicted that the system used for the case study would incur first EOM events, on average (appeared at least 50% of the time), for:

- 41 individual cards within the system (worst-case)
- 22 individual cards within the system (worst-case including part harvesting)

In this test case, the implementation of harvesting led to an avoidance of 19 additional cards incurring EOM events by the year 2050—showing that there is part similarities among the cards within the electronic system, and that part harvesting is a viable tactic for delaying additional EOM events for systems whose cards have part similarities.

The EOR/EOM model was also used to observe the effects of individual selected card refreshes on system sustainment. The model predicted that the system used for the case study could extend its average first EOM date by:

- 0.22 calendar years through refreshing Card 63 (best-case)
- 0.16 calendar years through refreshing Card 63 (worst-case)
- 0.19 calendar years through refreshing Card 61 (worst-case including part harvesting)

The extension of the system support life is most extended through the individual card refresh of the Card 63 when part harvesting is not considered. However, the system support life is most extended through the individual card refresh of Card 61 when part harvesting is used.

The EOR/EOM model was also used to produce design refresh plans, in order to ensure system sustainment to a specified end of support date. The model predicted that design refresh plans to ensure system sustainment until the year 2050 for the case study system would include, on average (appeared at least 50% of the time):

- 10 individual card refreshes (best-case)
- 39 individual card refreshes (worst-case)
- 22 individual card refreshes (worst-case including part harvesting)

The implementation of harvesting led to an avoidance of 19 additional cards incurring EOM events by the year 2050—showing that there is part similarities among the cards within the electronic system, and that part harvesting is a viable tactic for delaying additional EOM events for systems whose cards have part similarities. Design refresh planning (worst-case) delayed 2 EOM events (compared to worst-case of tracking subsequent EOMs to 2050), due to eliminating the additional part demands via design refreshing.

In this chapter, the EOR/EOM model was used to observe the legacy system sustainment and support cost ramifications through a composition of different scenarios (immediate first failures and part harvesting). The results of the case study showed that the support life of the system was, on average, 20 years. The assumption of immediate failures lowered the average system support life by 2 years, while the implementation of part harvesting extended the system support life by approximately 2 years.

The EOR/EOM model was also used in conjunction with the design refresh concept to conduct a sensitivity analysis on the system to determine the individual selected card refresh that would result in maximum system sustainment, and design refresh planning to ensure system sustainment to a specific date. The implementation of design refresh planning delayed additional EOM events that were observed in similar test cases where no design refreshes were used.

# Chapter 4 : Evaluation of Lifetime Buy Estimations to Minimize Life-Cycle Cost

Lifetime buy is an obsolescence mitigation strategy that refers to buying enough parts from the original manufacturer prior to the discontinuance of the part in order to support all forecasted future part needs throughout the system support life. This can be challenging for system supporters as they must be able to predict how many part spares will be needed to support their system for the remainder of its support life (referred to as the *total quantity needed* in this thesis) at the moment they make the lifetime buy purchase. The *lifetime buy quantity* or *initial buy quantity* is the quantity of spares purchased at the time of the lifetime buy. The *total quantity needed* is the quantity of spares required to support the future part needs through the system support life. Spares purchased at the lifetime buy are placed in inventory for storage until they are requested; however, the total quantity needed may be affected by spares that may be removed from inventory for reasons other than replacing failed parts within the field (i.e., part degradation, scheduled manufacturing demands, periodic inspections, and testing as discussed in Section 4.1).

The main questions that system supporters think of when considering lifetime buys are, "What is the correct lifetime buy quantity that will meet my systems' needs?"[11] and "What is the total life-cycle cost associated with the lifetime buy quantity that I purchase?". The procurement of spare parts is only the first step in evaluating the total life-cycle cost associated with the purchased spares; these (purchased) parts must also be stored and held in inventory and used.

---

[11] Note, the "correct lifetime buy quantity" is not generally the same as the total part demand (even with the extra parts needed to accommodate testing, degradation, etc.). The correct lifetime buy quantity is the quantity that minimizes the life-cycle cost of the system.

After the lifetime buy quantity purchase is received, the parts are stored in inventory until they are needed. Consequently, there may also be penalties for buying more (overbuy) or less (underbuy) spares at the lifetime buy than what is required to support the system (see Fig. 4.1), resulting in additional costs to system supporters. If the supporter should overbuy, the additional parts may simply be disposed. However, if the supporter does not buy enough parts <u>at</u> the lifetime buy (underbuy), the system supporter will need to purchase the parts elsewhere at a later date (i.e., buying from aftermarket sources) for a higher price.

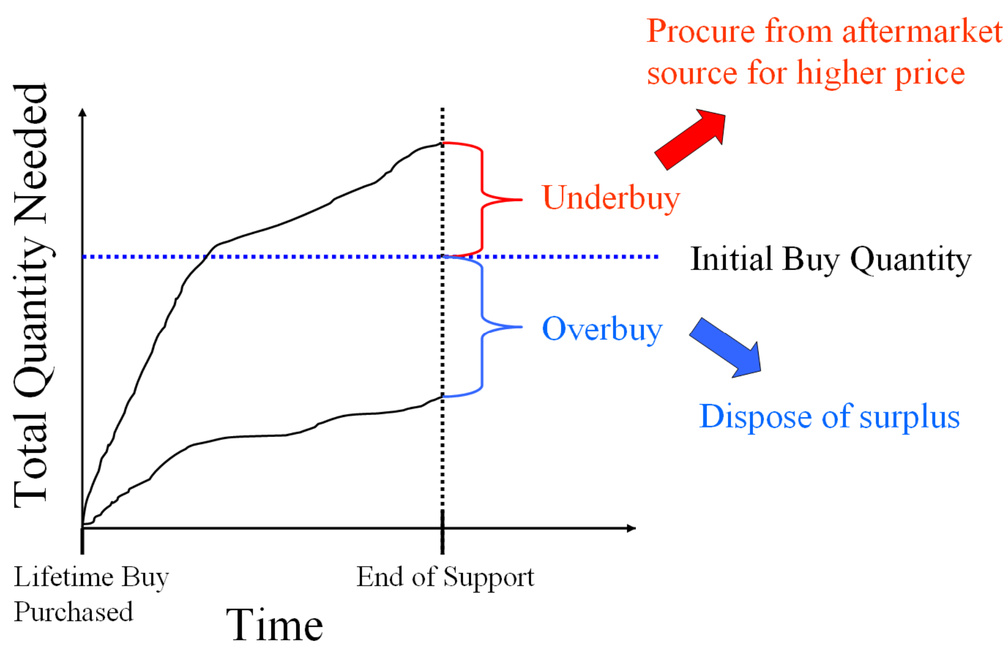



Fig. 4.1 Penalty costs for underbuy and overbuy

The asymmetry of the penalties define a “newsvendor” optimization problem. The "newsvendor" problem [44] is a one-time business decision that is applicable in many different business contexts and has been around for over 100 years [45]. The problem concerns a newsvendor who must order newspapers for the day. If the vendor <u>orders too many</u> newspapers, some of the papers will have to be thrown away or may even be sold as scrap paper. If the vendor does not order enough newspapers, some of the customers will be disappointed and sales and profit will be lost. The problem is to find the optimal number of newspapers to buy that will maximize the expected (average) profit given that the demand distribution and cost parameters are known. In the classically defined newsvendor problem, the penalties are symmetric because papers that are purchased by the newsvendor but not sold cost the newsvendor a different amount than demand for papers that the newsvendor could not fulfill.

The application of classical newsvendor solutions to lifetime buys of electronic parts has been discussed in [50]. There has also been previous work done on the lifetime buy problem which includes addressing the problem from the buyer's perspective [46] and the seller's perspective [47]. Feng, et. al [48] extended the final order model [47] and applied it to electronic part obsolescence; however, these models operate under a set of assumptions. The planning horizon for the final order model [47] is divided into intervals of equal length where demand and supply are allotted at the end of each interval. Additionally, penalty costs are allocated at the end of the intervals and inventory holding costs are allocated at the beginning of the intervals. The lifetime buy model proposed in this thesis is developed using a discrete event simulation model where parts demand are independently requested and individual costs are allocated. Additional efforts have been made to investigate a similar problem (referred to as the 'last buy problem') to provide continuous-time solutions for various cases involving no replenishment, batch replenishment, and incremental replenishment of spare parts [49]. The following section discusses the development of a discrete-event simulation lifetime buy model for finding the optimum lifetime buy quantity that minimizes the total life-cycle cost associated with the lifetime buy quantity given that the demand distribution and cost parameters are known.

### *4.1 Development of a Discrete Event Simulation Lifetime Buy (LTB) Model*

The development of the Lifetime Buy (LTB) model stems from the reverse application of the EOR/EOM model and is also implemented as a stochastic discrete-event simulation. The LTB model tracks a fielded population of a single part in order to support its forecasted demands to a specified *End of Support Date* defined as the date when systems' operations are either discontinued or no longer required. In the EOM problem, the model was developed to determine the support life of the system based on non-replenishable inventories of spare parts and cards. Instead of starting with inventories full of parts and counting down to zero, the LTB model starts with an inventory containing zero parts and counts up based on the forecasted demands (see Fig. 4.2) obtained from sampling the failure distribution of the part.

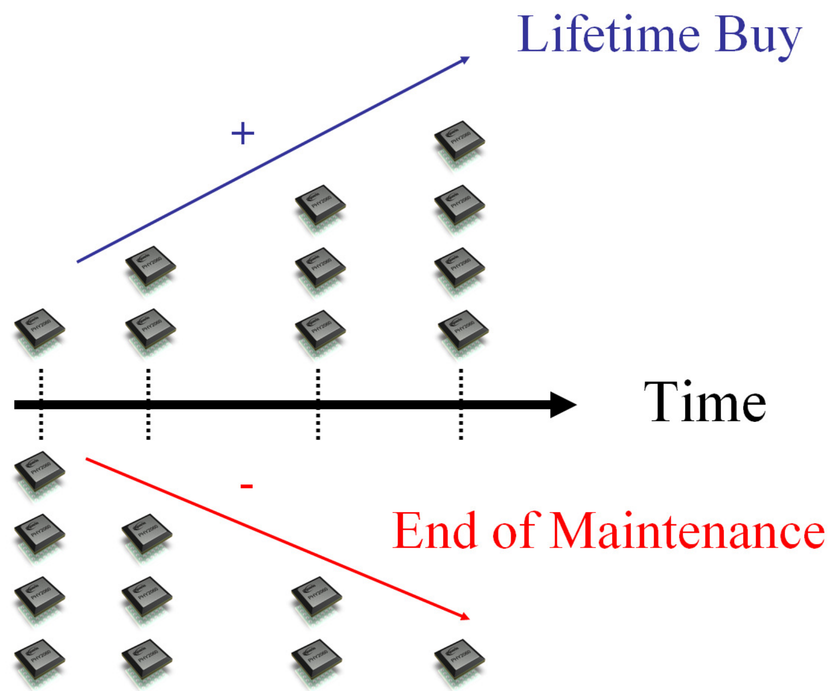


Fig. 4.2 Count up (lifetime buy) versus count down (end of maintenance) of spares

The simulation adds a spare part to the inventory when forecasted demand dates are reached and terminates when all fielded systems have been sustained to the specified End of Support date representing the total quantity needed for the system. As previously mentioned, there are additional actions where spares may be requested outside of replacing failed parts within the field. The LTB model contains the same events that prompt demands for parts as the EOR/EOM model (see Chapter 2) including:

- Spares due to part failures
- Part degradation in inventory (i.e., shelf life)
- Periodic inspection and testing
- Manufacturing demands (from a provided schedule)

*4.1.1 Scheduled Manufacturing Demands*

The continuation of manufacturing new and additional systems can affect the total quantity needed if the discontinued part in question is included on the newly added systems. These additional parts must be available on the date the scheduled manufactured systems are fielded. Additionally, these fielded parts must be represented with subsequent forecasted part demands and sustained to the End of Support date for determining the total quantity needed.

*4.1.2 Retirement Schedules*

The continuation of manufacturing new and additional systems can affect the total quantity needed if the discontinued part in question is included on the newly added systems. These additional parts must be available on the date the scheduled manufactured systems are fielded. Additionally, these fielded parts must be represented with subsequent forecasted part demands and sustained to the End of Support date for determining the total quantity needed.

*4.1.3 Lifetime Buy Problem Formulation*

The total quantity needed (and total life-cycle costs associated with the lifetime buy quantity) for the lifetime buy problem can be determined using the formulae shown in equations (4.1) and (4.2):

$$f_3(\overline{p}) = \sum_{i=1}^{n} Q_i \tag{4.1}$$

$$
\begin{aligned}
f_4(\overline{p}) = P \cdot L_t + \sum_{i=1}^{r} \frac{NRE_i}{\left(1+\frac{R}{100}\right)^{y_b}} + \sum_{i=1}^{n} \frac{(D_i - D_{i-1})LTB_{i-1}C_{i-1}}{\left(1+\frac{R}{100}\right)^{D_i - y_b}} \\
+ \sum_{i=1}^{n} \frac{M_i}{\left(1+\frac{R}{100}\right)^{D_i - y_b}} + \sum_{i=1}^{n} \frac{(D_i - D_{i-1})I_i}{\left(1+\frac{R}{100}\right)^{D_i - y_b}} + \sum_{i=1}^{n} PN_i(Q_i, \overline{p})
\end{aligned}
\tag{4.2}
$$

subject to: $g_k(\overline{p}) < F_p;\; k = 1,\ldots,K$

where,

| | |
|---|---|
| $P$ | Procurement cost per part |
| $L_t$ | Quantity of parts purchased at the lifetime buy |
| $NRE_i$ | Cost of the *i*th non-recurring cost |
| $r$ | Number of non-recurring costs |
| $D_i$; $D_{i-1}$ | Difference in years between *i*th and previous maintenance event date |
| $Q_i$ | Quantity of parts added to inventory at the *i*th maintenance event |
| $LTB_i$ | Lifetime buy quantity of parts stored in inventory at the *i*th maintenance event |
| $C_i$ | Recurring cost of holding a part in inventory to the *i*th maintenance event |
| $n$ | Number of maintenance events needed to support all fielded parts to end of support date |
| $M_i$ | Maintenance activity costs associated with the *i*th maintenance event |
| $I_i$ | Infrastructure costs associated with the *i*th maintenance event |

| | |
|---|---|
| $PN_i$ | Penalty costs associated with the *i*th maintenance event |
| $R$ | After tax discount rate on money |
| $y_b$ | Base year for money |
| $k$ | Index used to identify a particular constraint |
| $K$ | Number of constraints |
| $F_p$ | Number of fielded parts |

The objective function, $f_3(\overline{p})$, calculates the total quantity needed to sustain the fielded systems to the End of Support date. The objective function is dependent on $\overline{p} = [p_1, \ldots, p_m]$, which is the set of system parameters that describe the system. The parameters used in the total quantity objective function include part reliabilities and quantities, system support life duration and operational profile, and additional events that request demands for parts (e.g., manufacturing, inspections). Some of these parameters are uncertain; however, everything is known about the behavior and range of variation for each parameter. The system begins at a specific start date ($D_0$) and progresses upon arriving at $D_n$, where prior to the event, the considered constraint $g_k(\overline{p})$ equaled $F_p$ minus one, and by the end of the time step, $g_k(\overline{p})$ will have been violated (equaling $F_p$ at some $D_n$).

In equation (4.2) the objective function, $f_4(\overline{p})$, calculates the total life-cycle costs associated with the lifetime buy quantity purchased at $D_0$. The expressions in the equation represent the procurement cost, non-recurring costs, inventory holding costs, maintenance costs, infrastructure costs, and penalty costs, respectively. This function, $f_4(\overline{p})$, incorporates the same parameters as equation 4.1. Both objective functions are constrained in the same manner, whereby the simulation is terminated after all fielded parts of the system have been sustained beyond the End of Support date.

### *4.2 Inputs and Outputs of the LTB Model*

The LTB model tracks a single population of fielded parts and accumulates the number of spares needed to meet the systems' demands until the End of Support date is reached representing the total quantity needed. The inputs of the model include the systems' characteristics (part reliability, fielded quantity, estimated initial buy or lifetime buy), the simulation inputs (analysis date, end of support date, and operational profile), and the cost inputs discussed in detail in Section 4.3. The outputs from the LTB model (see Fig. 4.3) include the total quantity needed and the total life-cycle cost associated with the lifetime buy quantity. The model accounts for other activities (i.e., periodic inspection and testing) that may or may not demand additional spares to be accumulated towards the total quantity needed. However, this quantity is not exactly the same each time the simulation is conducted for a given set of parameters—the output is represented as a probability distribution of *total needed quantities* (and probability distributions of the total life-cycle costs associated with the *lifetime buy quantities*) to account for inherent system uncertainties (i.e., part reliabilities).

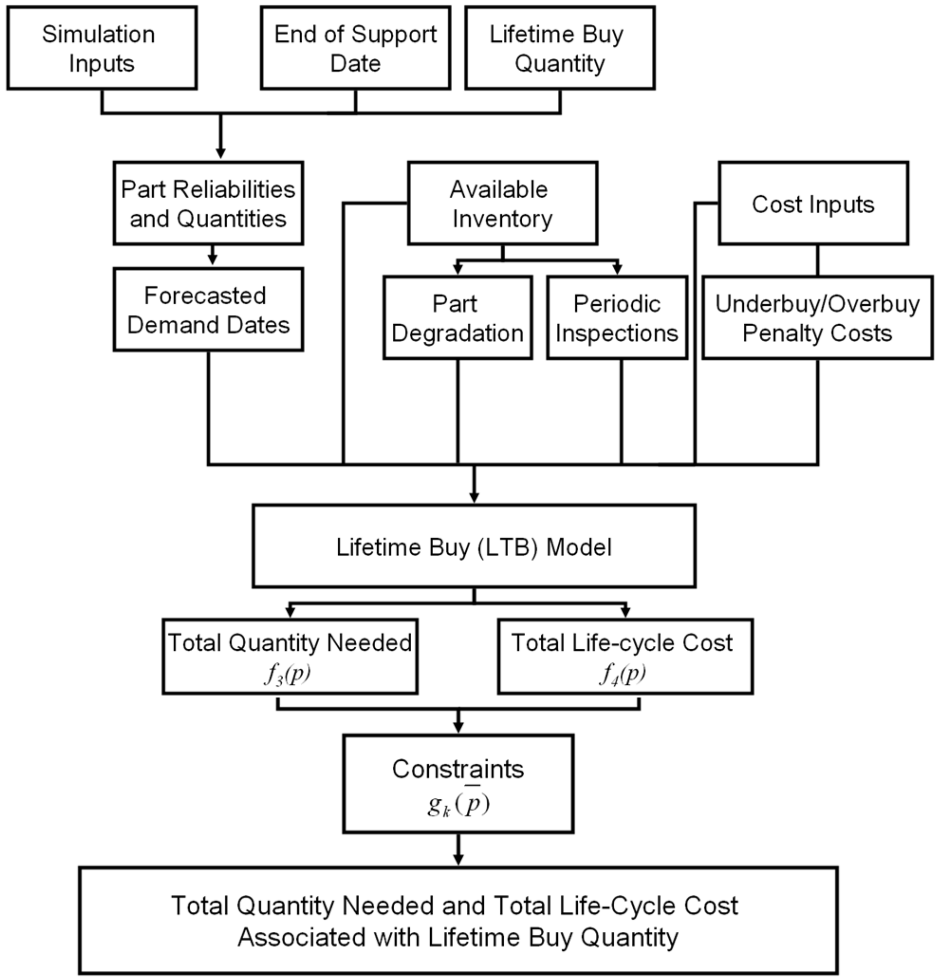



Fig. 4.3 Inputs and outputs of the lifetime buy (LTB) model

## *4.3 Implementation of Total Life-Cycle Costs Associated with the Lifetime Buy Quantity*

The total life-cycle cost objective function, $f_4(\overline{p})$, assesses life-cycle cost associated with the spare parts purchased at the lifetime buy. The total life-cycle cost ($C_{TLC}$) at any given time during the life history of the system is the sum of the procurement cost ($C_{PR}$), nonrecurring costs ($C_{NRE}$), cost of maintenance activities ($C_M$), inventory holding costs ($C_{IH}$), infrastructure costs ($C_{Inf}$), and penalty costs ($C_{PN}$). The total life-cycle cost associated with the lifetime buy quantity can be calculated at any simulation time using (for a given base year of money $y_b$ and discount rate *R*),

$$C_{TLC}(y_b, R) = C_{PR}(y_b, R) + C_{NRE}(y_b, R) + C_M(y_b, R) + C_{IH}(y_b, R) + C_{Inf}(y_b, R) + C_{PN}(y_b, R) \tag{4.3}$$

Prior to accumulation, each cost is converted to its net present value (*NPV*) based on the current simulation time, the base year, and discount rate. The *NPV* of a cost ($C_x$) at time *t* is given by,

$$C_x(NPV) = \frac{C_x}{(1+R)^{t-y_b}} \tag{4.4}$$

The total life-cycle costs associated with the lifetime buy quantity are then accumulated after the net present value of each sub-cost is calculated.

The procurement cost includes the costs associated with the purchasing of the parts. This cost is treated as a non-recurring cost upfront at the analysis start date (date the lifetime buy purchase is made). The sum of the procurement cost, $C_{PR}$, is calculated by multiplying the procurement cost per part, (*P*), and the lifetime buy quantity, ($L_t$) as shown:

$$C_{PR} = PL_t \tag{4.5}$$

The non-recurring costs, $C_{NRE}$, are costs that are charged at the same time that the lifetime buy is purchased. These non-recurring costs are sub-divided into test/screening ($C_{TS}$), packaging ($C_{PG}$), part qualification ($C_{PQ}$), and supplier qualification ($C_{SQ}$) as shown:

$$C_{NRE} = C_{TS} + C_{PG} + C_{PQ} + C_{SQ} \tag{4.6}$$

The maintenance activity costs, $C_M$, include the costs associated with administrative actions ($C_A$), replacement ($C_R$), disposal ($C_D$), and inspection ($C_I$) as shown:

$$C_M = C_A + C_R + C_D + C_I \tag{4.7}$$

Maintenance costs are accumulated as a result of a discrete event occurring. Administrative costs are accrued from any type of event that occurs in the simulated life history (part replacement, degradation, inspection). Replacement costs are accrued from corrective maintenance activities (replacing failed parts). Disposal costs are accrued per part and occur with replacement and removal of parts. Inspection costs are added per part and are dependent on the number of parts that are inspected within a specific inventory (additionally, the inspected parts may also be disposed).

The inventory holding costs include the cost of storing spare parts over time, which is described in equation (4.2). The inventory holding costs are accumulated as a result of time periods between discrete events and account for the time a certain quantity of items is held in inventory. The infrastructure cost, also described in equation (4.2), is a recurring cost that represents the basic organizational and physical structures needed for systems' operations.

The penalty costs, $C_{PN}$, are recurring costs that incorporate the lifetime buy quantity initially purchased and the cumulative total quantity of parts throughout the system field life. As previously mentioned, penalty costs are sub-divided into two types (both are $ per part): an underbuy penalty ($C_{UB}$) and an overbuy penalty ($C_{OB}$).

The overbuy penalty is charged at the end of the simulation if the lifetime buy quantity is greater than the total quantity needed. The penalty cost is as shown:

$$C_{PN} = C_{OB}(Q_{LTB} - Q_{TQN}) \tag{4.8}$$

where,

$Q_{LTB}$ = lifetime buy quantity

$Q_{TQN}$ = total quantity needed.

In the case of underbuy, the penalty cost is then treated as a recurring cost that is charged at the *i*th maintenance event and is shown as,

$$C_{PN_i} = (P + C_{UB})(Q_{TQN_i} - Q_{LTB}) \tag{4.9}$$

where,

$Q_{TQN_i}$ = cumulative total quantity at the *i*th maintenance event.

The underbuy penalty is charged in addition to the original procurement part price and accumulates for each additional event where enough parts were not purchased at the lifetime buy until all fielded parts are sustained through the $Q_{TQN}$.

In this manner, the LTB model can use the total life-cycle cost associated with the estimated lifetime buy quantity (assuming constant values for underbuy and overbuy penalties) to find the optimum $Q_{LTB}$ that results in the minimal total life-cycle cost associated with the lifetime buy quantity. This can be performed by choosing an estimated lifetime buy quantity, running the simulation to observe the total life-cycle cost, and increasing or decreasing the estimated lifetime buy quantity based on the assumed penalties for overbuying and underbuying spare parts.

*4.4 Lifetime Buy Case Study*

The following case study demonstrates the capability of the LTB model and how it can be used to generate the optimum average lifetime buy quantity that results in a minimal total life-cycle cost associated with the estimated lifetime buy quantity. The system is composed of 1,000 fielded parts where each part is characterized by a 2-parameter Weibull failure distribution (β equals 2 and η equals 35,000 operational hours). The system receives their lifetime buy purchase on January 1, 2011, and the system must be supported until January 1, 2019. The system is assumed to be fully operational (8,760 hours per year). There is part degradation in the inventory—assumed to degrade one part every 4,000 operational hours. Periodic inventory inspections occur every six months and pull five parts from the inventory that are not replaced. The cost inputs for the case study can be seen in Table 4.1. The part purchase price is for the date the lifetime buy purchase is made and assumed to be received (January 1, 2011).

Table 4.1 Cost Analysis Inputs for Lifetime Buy Case Study

| | |
|---|---:|
| Test/screen NRE cost, $C_{TS}$ ($) | 7,000 |
| Packaging NRE cost, $C_{PG}$ ($) | 15,000 |
| Part purchase price, $P$ ($ per part) | 25 |
| Underbuy penalty, $C_{UB}$ ($ per part) | 100 |
| Overbuy penalty, $C_{OB}$ ($ per part) | 2 |
| Administrative cost of a draw, $C_A$ ($) | 2.5 |
| Replacement cost of a draw, $C_R$ ($) | 13 |
| Part inventory cost, $C_i$ ($ per part per year) | 1.5 |
| Part inspection cost, $C_I$ ($ per part) | 8 |
| Unusable part disposal cost, $C_D$ ($ per part) | 0.5 |
| Discount rate, $R$ | 5% |
| Base year for money, $y_b$ | 2011 |

It is assumed that these "per-action" costs (including discount rate and base year) are treated as constants and that they do not change values with respect to time or any other parameter.

### *4.4.1 Lifetime Buy Quantity of 2,000 Parts*

The estimated lifetime buy quantity chosen at the analysis start date was 2,000 parts. The total quantity needed and total life-cycle cost results can be seen in Table 4.2. The values in Table 4.2 represent average values of each cost over the total number of simulated life histories (typically 1,000 are conducted). The probability distributions of the total quantity needed and total cost for the lifetime buy quantity of 2,000 parts are shown in Fig. 4.4. The total quantity needed is independent of the costs. The generation of the output probability distributions allows for statistical interpretation of the collected results from each simulated life history of the system. The total quantity required could range from as little as 2,912 parts to as many as 3,066 parts and, on average, requires 2,993 parts to support the 1,000 fielded parts within the system through January 2019.

Table 4.2 Cost Analysis Outputs for Lifetime Buy Case Study (2,000 Parts)

| | |
|---|---|
| Total quantity needed, $Q_{TQN}$ | 2,993 |
| NRE cost, $C_{NRE}$ (\$) | 22,000 |
| Procurement cost, $C_{PR}$ (\$) | 50,000 |
| Inventory holding cost, $C_{IH}$ (\$) | 12,105 |
| Administrative cost, $\Sigma C_A$ (\$) | 3,887 |
| Replacement cost, $\Sigma C_R$ (\$) | 19,862 |
| Disposal cost, $\Sigma C_D$ (\$) | 803 |
| Inspection cost, $\Sigma C_I$ (\$) | 512 |
| Penalty cost, $C_{PN}$ (\$) | 83,795 |
| Total cost, $C_{TLC}$ (\$) | 192,963 |

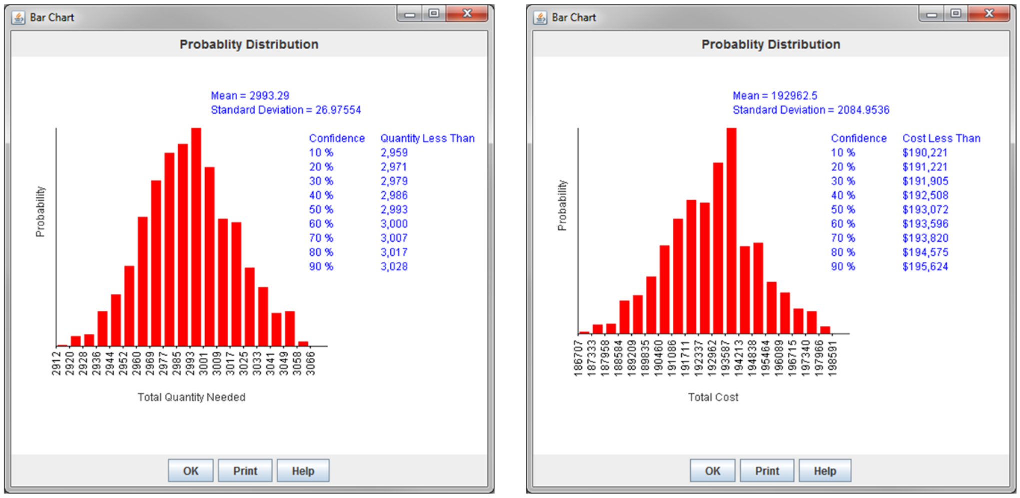



Fig. 4.4 Total quantity needed and total cost for lifetime buy of 2,000 parts

The levels of confidence associated with the output data can also be extrapolated. Based on the results, one could say they are 50% confident that the total quantity needed is less than 2,993 parts and will cost less than $193,072.

### *4.4.2 Lifetime Buy Quantity of 3,000 Parts*

The estimated lifetime buy quantity chosen at the analysis start date was changed to 3,000 parts (to better reflect the average total quantity needed of 2,993 parts). The total quantity needed and total life-cycle cost results can be seen in Table 4.3. The values in Table 4.3 represent average values of each cost over the total number of simulated life histories (typically 1,000 are conducted). The probability distributions of the total quantity needed and total cost for the lifetime buy quantity of 3,000 parts are shown in Fig. 4.5. The total quantity needed is independent of the costs. The generation of the output probability distributions allows for statistical interpretation of the collected results from each simulation life history. The total quantity required does not change between the two test cases—the only parameter that has been changed is the lifetime buy quantity. All of the costs (except for the penalty) increase due to the increase in the lifetime buy quantity (from 2,000 to 3,000). The penalty cost is much lower ($600 compared to $83,000) due to the accurate lifetime buy quantity chosen for supporting the system. The average total life-cycle cost was reduced from $193,000 to $155,000. The accurate lifetime buy quantity led to a $38,000 cost avoidance in considering the lifetime buy.

Table 4.3 Cost Analysis Outputs for Lifetime Buy Case Study (3,000 Parts)

| | |
|---|---|
| Total quantity needed, $Q_{TQN}$ | 2,993 |
| NRE cost, $C_{NRE}$ ($) | 22,000 |
| Procurement cost, $C_{PR}$ ($) | 75,000 |
| Inventory holding cost, $C_{IH}$ ($) | 22,039 |
| Administrative cost, $\Sigma C_A$ ($) | 5,552 |
| Replacement cost, $\Sigma C_R$ ($) | 28,517 |
| Disposal cost, $\Sigma C_D$ ($) | 1137 |
| Inspection cost, $\Sigma C_I$ ($) | 523 |
| Penalty cost, $C_{PN}$ ($) | 626 |
| Total cost, $C_{TLC}$ ($) | 155,394 |

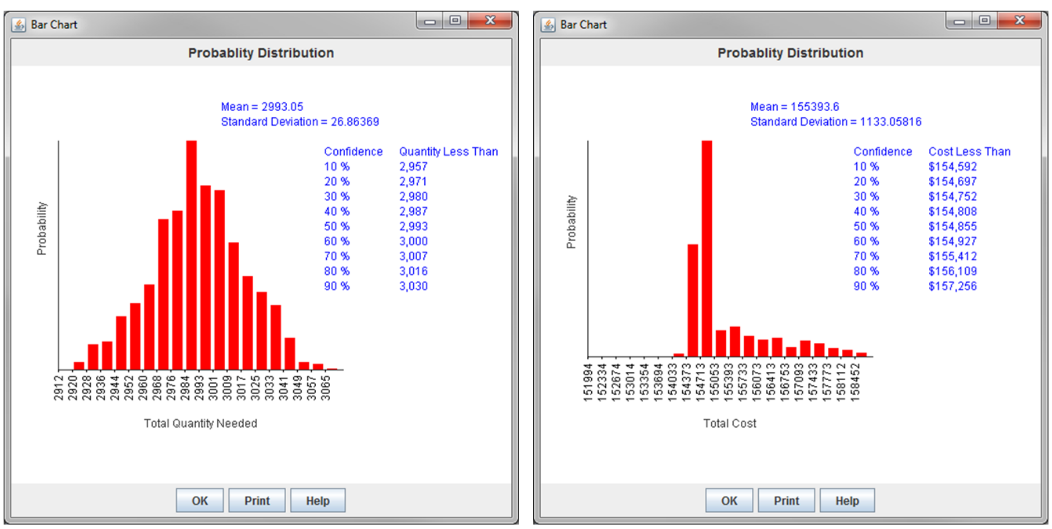


Fig. 4.5 Total quantity needed and total cost for lifetime buy of 3,000 parts

## *4.5 Finding the Optimum Lifetime Buy Quantity*

As previously mentioned, the LTB model can be used to generate lifetime buy quantities that result in a minimal total life-cycle cost associated with the lifetime buy quantity. Therefore, the LTB model can find the optimum lifetime buy quantity (based on assumed constant underbuy/overbuy penalties) that minimizes the total life-cycle cost associated with the lifetime buy (see Fig. 4.6).

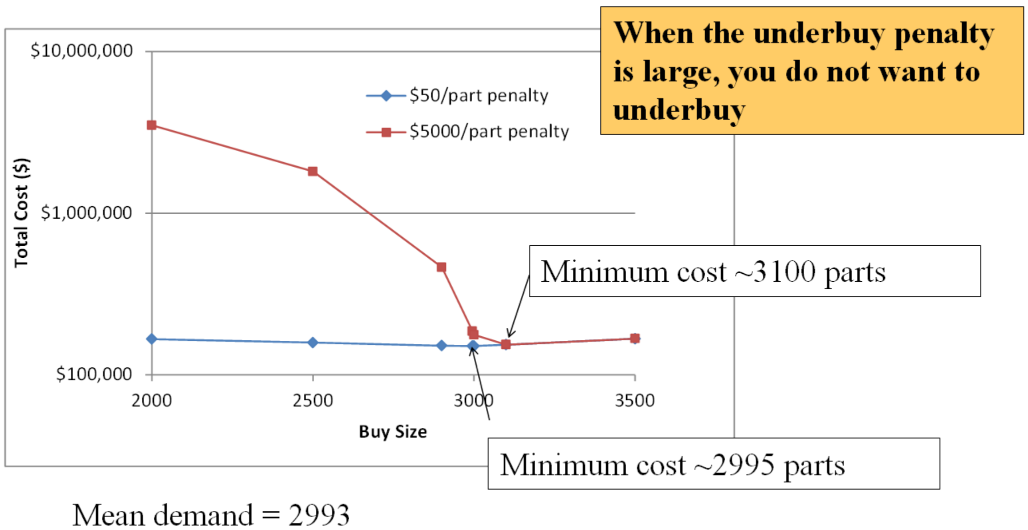



Fig. 4.6 Finding optimum lifetime buy quantity through minimal total life-cycle cost

As the underbuy penalty increases, the minimum total life-cycle cost (and optimum lifetime buy quantity) deviates away from the total quantity needed. Furthermore, if the underbuy penalty is small compared to other costs, then the minimum total life-cycle cost (optimum lifetime buy quantity) approaches the total quantity needed.

# Chapter 5 : Summary and Contributions

*End of Repair* (EOR) is defined in this thesis as "the date that the last repair or manufacturing action associated with a part can be successfully performed." EOR dates are part-specific and may also be card-specific if a particular card can only draw from a subset of the available inventories. Similarly, *End of Maintenance* (EOM) is defined in this thesis as "the earliest date that all available inventories fail to support the demand for one or more specific parts resulting in the loss of system operation." EOM events are caused by a specific part on a specific card.

This thesis described the development of a stochastic discrete event simulation EOR/EOM model, that follows the life history of a population of parts and cards, and determines how long the system can be sustained (and how much it costs to sustain) based on existing inventories of spare parts and cards, and optionally harvesting of parts from existing cards to increase system support life. The EOR/EOM model describes the process of inventory depletion of parts subject to DMSMS-type obsolescence through system operation and tracks the EOR and EOM dates, the critical parts associated with each EOR/EOM event, and the likelihood that these EOR/EOM events will occur for the system.

Reversing the EOR/EOM modeling process, which draws parts from inventories until the inventories are exhausted, a lifetime buy quantity model that filled empty inventories to support a system to a specified end of support date was formed form the same simulation.

*5.1 Contributions*

The research work presented in this thesis makes the following contributions:

1) Developed detailed definitions of End of Repair (EOR) and End of Maintenance (EOM). Although general notions of EOR and EOM previously existed, this thesis articulated detailed definitions that can be applied to actual quantitative analysis.
2) Created a methodology for performing support life and support cost assessments for legacy systems composed of parts and cards based on the systems' existing inventories. This methodology is the first model to specifically target the forecasting and analysis of system-level EOR and EOM.
3) Developed method for harvesting parts to further extend system support life capabilities. This thesis is the first known work to quantitatively model and implement part harvesting (reclamation) activities aimed at electronic system sustainment modeling.
4) Developed a methodology for design refresh planning to ensure system sustainment to a specific end of support date. Sensitivity analyses using individual card refreshes can also be performed to examine effects on the system support life.
5) Created a methodology for generating optimum lifetime buy quantities of parts that minimizes the total life-cycle cost associated with the lifetime buy.

### *5.2 Future Work*

There are many directions that the current work can be extended. These future extensions for the EOR/EOM model (and LTB model) include the treatment of non-standard parts, part dependencies, inventory replenishment, planned obsolescence, and End of Support uncertainty.

#### *5.2.1 Non-Standard Parts*

A major portion of the DMSMS-type obsolescence problem occurs for non-standard parts within mission-critical systems. Non-standard parts are parts that can be used towards multiple applications—this makes tracking the use of these parts difficult. Non-standard parts include Application Specific Integrated Circuits (ASICs) and altered or programmable parts. Non-standard parts create two issues in the analysis of End of Repair and End of Maintenance:

1) The definition of obsolescence for non-standard parts is unclear due to its ability to be used towards multiple applications. When do the inventories of these parts become obsolete?
2) There may not be a one-to-one correspondence between non-standard parts and the inventories from which they draw. Multiple non-standard parts may draw from a single inventory item, or a single non-standard part may need to draw multiple items from multiple inventories.

#### *5.2.2 Part Dependencies*

Part failures are presently replaced under the assumption that the system supporter has perfect knowledge of the part that caused the failure. In some cases, the reason for the failure may be unknown, and groups of parts may need to be replaced to eliminate the problem. Part dependencies may vary among parts (i.e., one part failure might depend on four parts while another might depend on two) and among cards (i.e., the same part located on different cards may have different part dependencies) but may not be linked to every failure that occurs for a part.

### *5.2.3 Replenishment of Inventories via Aftermarket Sources*

The EOR/EOM model follows the life history of a population of parts and cards and determines how long the system can be sustained based on existing non-replenishable inventories of spare parts and cards. However, parts facing DMSMS-type obsolescence can be procured from aftermarket sources (for a higher price) to replenish inventories of parts and further extend the support life capabilities of the system. Multiple procurements for multiple parts may occur at any time throughout system sustainment.

### *5.2.4 Managing Parts with Forecasted Obsolescence Dates*

The EOR/EOM model simulates electronic system sustainment when faced with DMSMS-type obsolescence. The current model stores the obsolescence status of unique parts as input parameters prior to analysis—this means that the forecasted system support life and cost assessments are based on an analysis of the parts that are already obsolete at the start of the simulation. Part obsolescence after the beginning of the simulation could occur and could be modeled. Electronic piece-part obsolescence date forecasts are readily available. The challenge is in assuming (modeling) what inventories of those parts would be put into place when their obsolescence occurs.

*5.2.5 End of Support Uncertainty concerning Lifetime Buys*

Lifetime buys are performed to ensure system sustainment to a specified end of support date. However, the end of support date is seldom known with complete certainty and could incorporate a range of possible dates—thereby have a dramatic effect on the total quantity needed and affecting the optimum lifetime buy quantity. Currently, the LTB model treats the end of support date parameter as a constant value. The LTB model should represent the end of support date as a probability distribution of possible dates.

# Appendix – EOR/EOM Software User's Manual

## A.1 Introduction to the EOR/EOM and LTB Analysis Software Tool

This document is the user's guide for the CALCE EOR/EOM and LTB analysis software tool. The EOR/EOM tool is a stochastic discrete-event simulation that follows the life history of a population of parts and cards determines how long the system can be sustained based upon existing inventories of replacement of parts and cards, and harvesting of parts off of existing cards. In discrete-event simulation, the operation of a system is represented as a chronological sequence of events. Each event occurs at an instant in time and marks a change of state in the system.

The EOR/EOM simulator follows individual parts through their fielded lifetimes. When a part fails, a maintenance event occurs (either to replace the part or the card that the failed part is located on). The simulation ends when the maintenance events can no longer be performed based upon existing non-replenishable inventories of spare parts and cards. In order to capture uncertainties in the characteristics of part failures and in the uncertainties in the characteristics of when the various maintenance events take place, the simulator follows a population of electronic systems through several life histories and determines probability distributions of system the resulting end of maintenance times.

The tool defines EOR and EOM as the following:

***End of Repair*** (EOR): The date that the last repair or manufacturing action associated with a part can be successfully performed.

***End of Maintenance*** (EOM): The earliest date that all available inventories fail to support the demand for one or more specific parts resulting in the loss of system operation

The user provides electronic system(s) information in the forms of unique part and card characteristics as inputs to the tool. For parts with failure history and where no failure distribution is assigned to a unique part, the tool synthesizes part failure distributions based upon past failure data and provides the user with probability distributions for how long the systems can be sustained based upon existing inventories and the identification of particular parts/cards that are the root cause of loss of system operations, as well as their frequency of occurrences (likelihoods).

The Lifetime Buy (LTB) simulator is the reverse-application of the EOR/EOM tool, and is a stochastic discrete-event simulation that determines the total lifetime buy quantity needed to sustain fielded systems to a specific date. The LTB tool can be used to find the optimum lifetime buy quantity that minimizes the total life-cycle cost associated with the quantity purchased at the lifetime buy.

The tool defines lifetime buy quantity and the total quantity needed as the following:

***Lifetime Buy Quantity*** (e.g., Initial Buy Quantity): The quantity of parts purchased at the lifetime buy.

***Total Quantity Needed*** : The quantity of parts required to support the future part needs through the system support life.

The user provides demand distributions and lifetime buy and cost inputs to simulate the requirement for parts demand over the system support life. The tool provides the user with probability distributions of the total quantity needed and the total life-cycle cost associated with the lifetime buy quantity.

IMPORTANT: The EOR/EOM and LTB functionality of the tool are separate, i.e., you must choose to either run the tool in EOR/EOM mode or LTB mode (but not both). Sections A.2-A.4 of this manual describe the EOR/EOM operation of the tool, and Sections A.5-A.7 describe the LTB operation of the tool. The mode (analysis type) in which the tool is run is selected by the users when they startup the tool (e.g., see Figures A.2.1 and A.5.1).

## A.2 EOR/EOM Tutorial

This tutorial includes loading system files into the EOR/EOM tool, running the EOR/EOM analysis, saving an EOR/EOM file, and loading an EOR/EOM file. This tutorial assumes that the user is running an application version of the tool and that the user has the minimum JRE (Java Runtime Environment) installed on their machine. This tutorial also assumes the user is running the CALCE EOR/EOM software on a PC running a Windows operating system, no attempt has been made to adapt the tool's functionality for performance on other platforms.

All fields and file formats are described in detail in Section A.3 of this manual.

### *A.2.1 Running the EOR/EOM Application*

1) Start the EOR/EOM and LTB application by double clicking on the executable. At the "Choose Analysis Type" dialog box, choose "EOR/EOM"

and Select "OK". You should obtain an interface like the one shown in Fig. A.2.1.

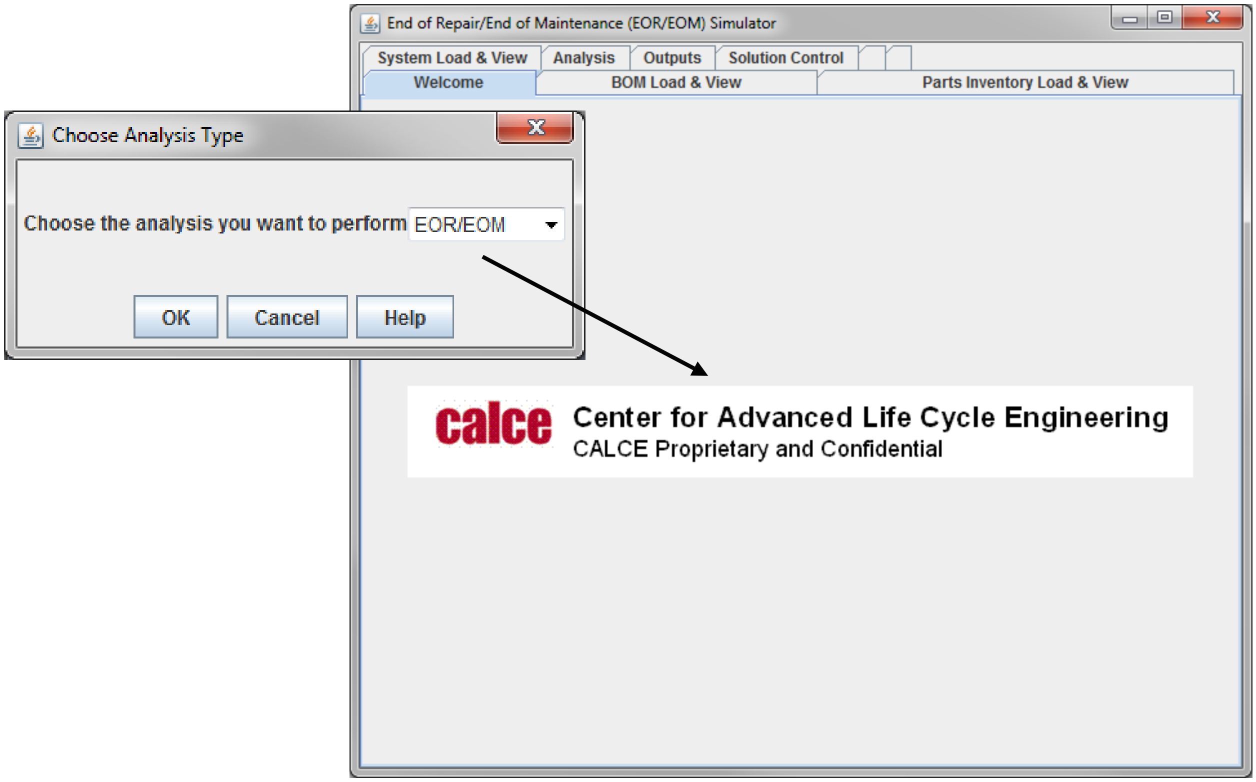



Fig. A.2.1 Initial startup of EOR/EOM tool

2) Select the "BOM Load & View" tab and click the button labeled "Load New Card BOM". Select the CSV file labeled "tutorial_card_1" located in the same directory as the executable. The resulting interface is shown in Fig. A.2.2. All of the part-specific information associated with "Sample Card 1" in the CSV file is now on display and saved within the EOR/EOM simulator. The user can also click on the "Reliability" fields associated with each loaded part and a dialog box will appear detailing the selected part's time-to-failure distribution type and its associated parameters. These time-to-failure distributions can be edited within the interface and are automatically saved when "OK" is clicked in the dialog box. All of the fields and buttons for all tabs within the EOR/EOM tool are detailed in Section A.3 of this manual.

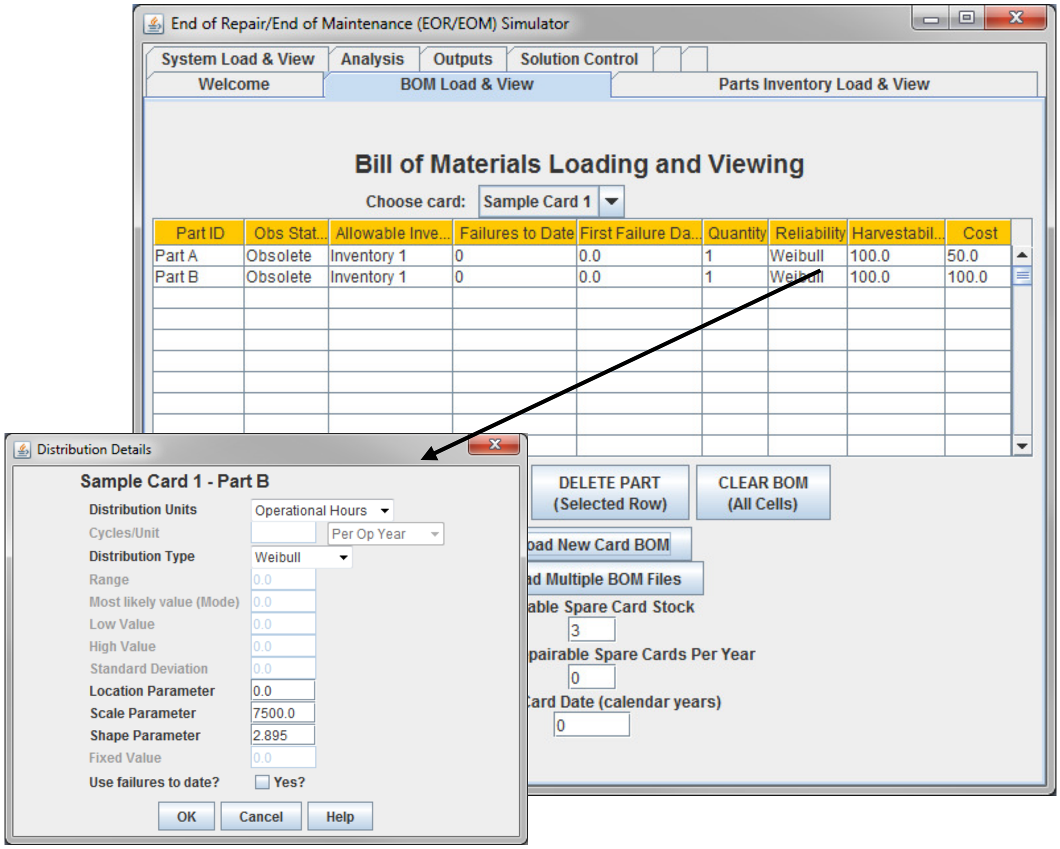



Fig. A.2.2 Sample Card 1 loaded into the EOR/EOM tool

3) Select the “Parts Inventory Load & View” tab and click the button labeled “Load New Inventory”. Select the CSV file labeled “inventory_1” located in the same directory as the executable. The interface should now look like Fig. A.2.3. The table displays the quantity of parts within the loaded inventory. The user can select the column header labeled “Inventory 1” to popup a separate dialog box that describes periodic inspection events associated with the selected inventory. In the same fashion, the user can select the different part quantity fields and dialog boxes will popup that describe the degradation details of the selected part (how often a particular part degrades in inventory). For now, we will leave both of these boxes blank.

4) Select the “System Load & View” tab and click the button labeled “Load New System”. Select the CSV file labeled “system_1”. The interface should now look like Fig. A.2.4. The table displays the quantity of cards that are occupied by the loaded system.

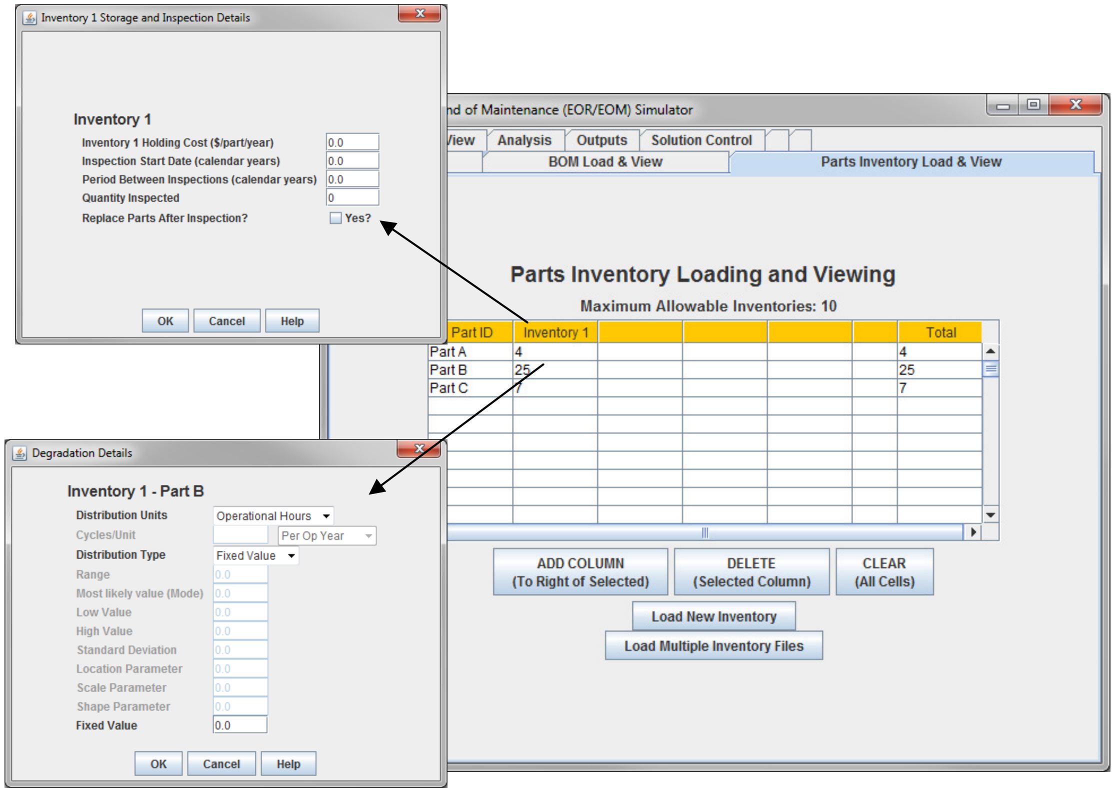



Fig. A.2.3 Inventory 1 loaded into the EOR/EOM tool

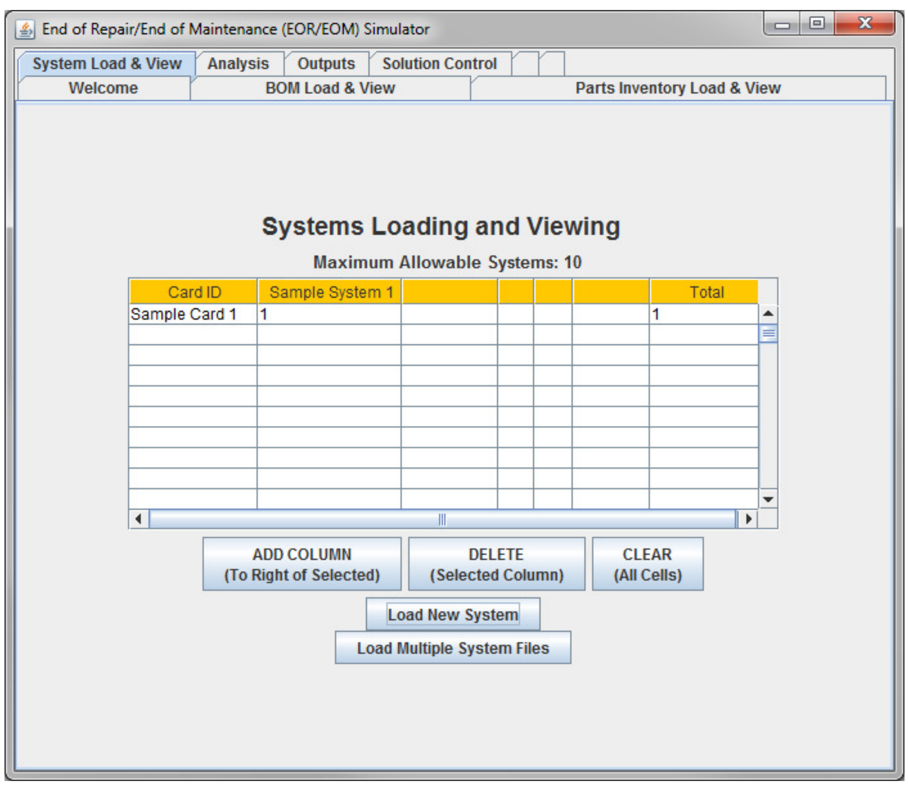



Fig. A.2.4 Sample System 1 loaded into the EOR/EOM tool

5) Once all the CSV files that characterize the system have been loaded into the EOR/EOM tool, the user must then define analysis inputs and solution control options. Select the “Solution Control” tab and type in “2011.0” for the Analysis start date and “8760.0” for the operational hours per year (24/7 operation). The interface should now look like Fig. A.2.5. Next select the “Solution Control” button at the bottom-left corner of the simulator interface. A dialog box will appear that allows the user to control how the EOR/EOM tool will analyze the inputted system(s). The default values are sufficient for this tutorial, click "OK", and close the dialog box.

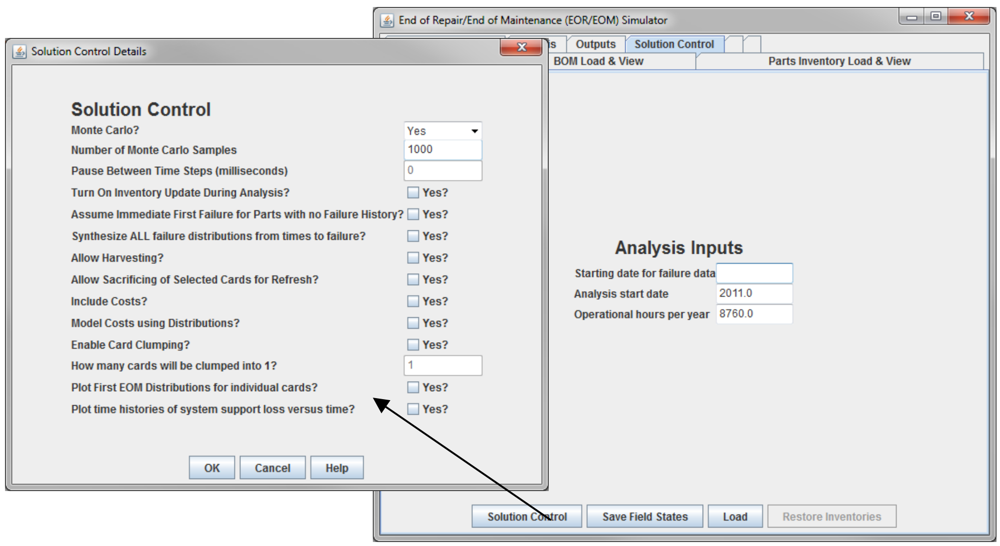



Fig. A.2.5 Solution Control Inputs for the EOR/EOM Analysis

6) Select the “Analysis” tab. The interface should now look like Fig. A.2.6. The user can select the “Card Index” button; this brings up a dialog box that displays all the cards that are loaded in the EOR/EOM tool. The user can then select the cards that they wish to include within the analysis (all cards loaded into the EOR/EOM tool are included in the EOR/EOM analysis by default). Click "OK" to close the Card Index dialog box. Click “Run”.

The application is now running; you can click on the "Pause" button to pause the simulation analysis. Pressing the "Stop and Reset" button will terminate the analysis. There is also a progress bar (shown in Fig. A.2.6) to indicate how many samples have been completed (this feature is only shown when the input number of samples is greater than or equal to 100, by default, the simulator will run 1,000 Monte Carlo samples).

After the simulation has completed, the interface should look similar to Fig. A.2.7. The simulator has run through its 1,000 simulations of the electronic system and the results show that Part A caused the first system EOM to occur at 2030.31 calendar years 53.5% of the time, and that Part B caused the first system EOM to occur at 2031.25 calendar years 46.5% of the time (these values will vary slightly based on inherent random sampling of the parts' time-to-failure distributions, so they may be different for your analysis. Click on the "Plot Dist" button and Click "OK" to accept the default plotting options. This enables the user to view the probability distributions associated with the electronic system's first EOM date (with and without the use of available spare cards)--the results provide the user with a statistical interpretation of the electronic systems' EOM events with regards to its existing inventories of spare parts and cards.

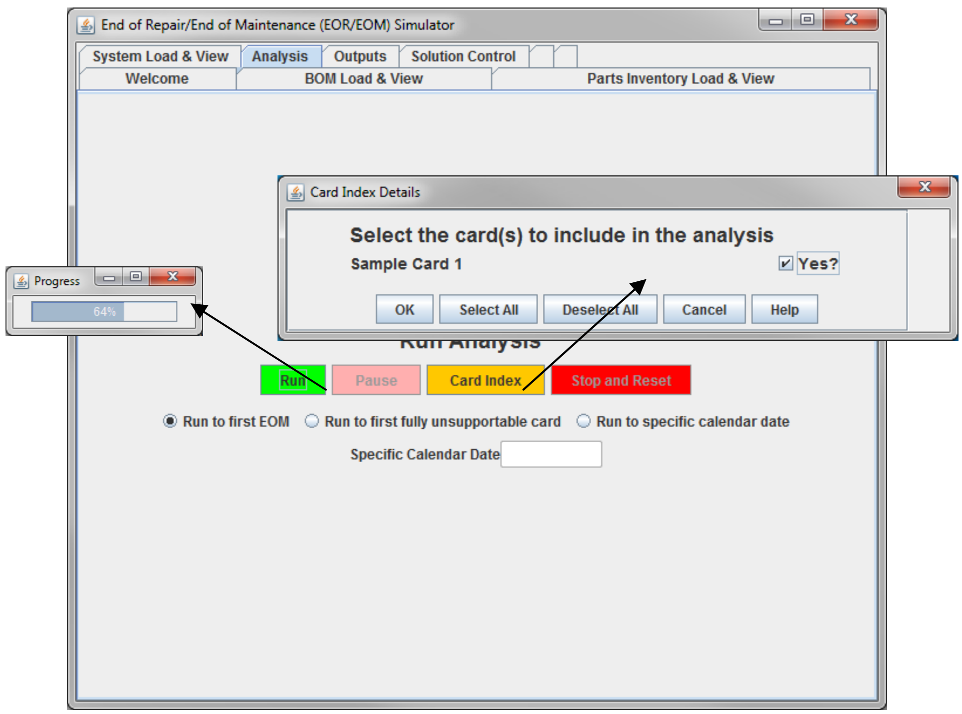



Fig. A.2.6 The EOR/EOM Analysis

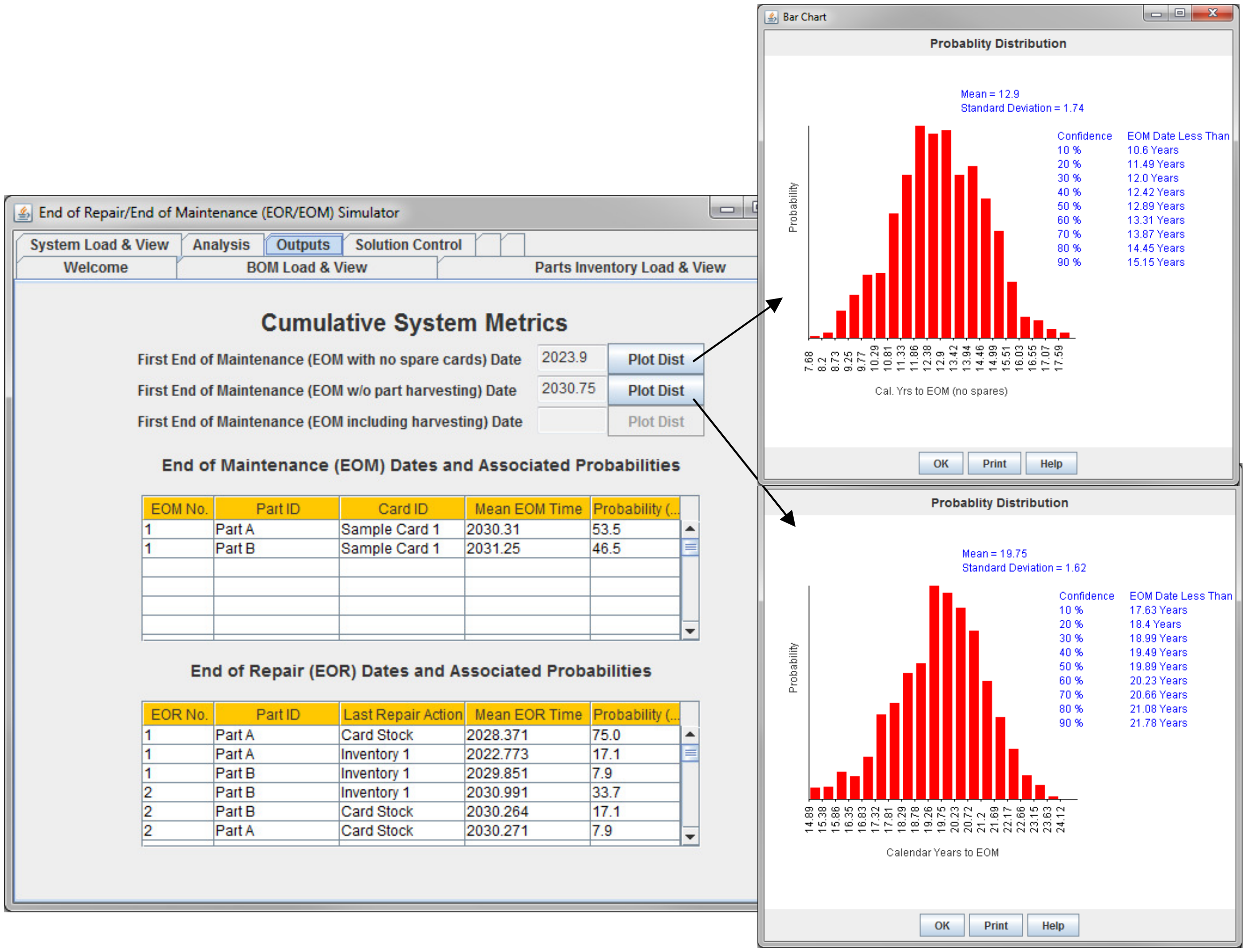



Fig. A.2.7 Cumulative System EOR and EOM Metrics

When the simulation completed, an output file containing all the recorded EOR and EOM information (name format contains "Metrics@Date Time") for each simulated life history is created and saved in the same directory as the CALCE EOR/EOM and LTB application.

*A.2.2 Saving an EOR/EOM File*

An EOR/EOM system file may be saved. Select the “Solution Control” tab and then click on “Save Field States”. Name the file “tutorial_1”, choose a desired saved location, and click “Save”. The EOR/EOM file has now been successfully saved with all the loaded system characteristics to the desired location on your machine.

*A.2.3 Loading an EOR/EOM File*

After your EOR/EOM file has been saved, exit the tool by selecting the red "X" in the top-right corner of the interface. Once the tool has been closed, re-open the CALCE EOR/EOM application. After selecting "EOR/EOM", you should see a screen that looks like Fig. A.2.1. Select the “Solution Control” tab and click on the “Load” button. Locate and select the file labeled “tutorial_1” and click “Open”. After the file is done loading, the loaded application should be in a state that is identical to the last save state.

## A.3 EOR/EOM Input File, Field and Button Reference

This section documents all of the fields within each CSV file that must be loaded into the EOR/EOM tool, and all the buttons and fields located in the EOR/EOM tool.

### *A.3.1 Bill of Materials (BOM) File Field References*

The Bill of Materials file format is shown in Fig. A.3.1.
The first cell reference (A1) is the name of the loaded card.

Cell A2 is the total number of unique parts (rows) located on the card.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Sample Card 1 | | | | | | | | | | | | | | | | | | | | |
| 2 | 2 | Obs status | Inventory | Failures | Quantity | Reliability | Mode | Shape | Range | Low | High | Stddev | Location | Scale | Fixed | First Failur | CardStock | harvestabil | card field date | unrepair percent | Cost |
| 3 | Part A | Obsolete | Inventory 1 | 0 | 1 | Weibull | 0 | 3.587 | 0 | 0 | 0 | 0 | 0 | 25000 | 0 | 0 | 3 | 100 | 0 | 0 | 50 |
| 4 | Part B | Obsolete | Inventory 1 | 0 | 1 | Weibull | 0 | 2.895 | 0 | 0 | 0 | 0 | 0 | 7500 | 0 | 0 | | 100 | 0 | 0 | 100 |

Fig. A.3.1 Sample Card CSV file format

Column B describes the obsolescence status of the part. Allowed inputs are:

Available: This means that the corresponding part is still considered actively procurable from part manufacturers; currently, available parts are not included in EOR/EOM analysis.

Obsolete: This means that the corresponding part is no longer sold or supported by the original equipment manufacturer (OEM) and can only be replaced if necessary by spares that are currently in the inventory—only parts deemed "Obsolete" are included in EOR/EOM analysis.

Column C defines the part's accessible inventory. The name of this inventory should correspond to one of the names of the loadable inventories that will be used in EOR/EOM analysis.

Column D defines the "failures to date" for the part. This is the number of failures observed between the part’s original fielded date (described by either the starting date for failure analysis seen in Fig. A.2.5 or the fielded date of the loaded card) and the beginning of the EOR/EOM analysis (referred to as the analysis start date). This characteristic is only used when the part's time-to-failure distribution is derived from past failure history occurrences rather than assigned a specific time-to-failure distribution.

Column E is the total quantity (number of instances) of the part that appear on the loaded card.

Columns F-O are only used if a predefined time-to-failure distribution is to be entered for the parts. If the failure history to date is going to be used, columns F-O can be ignored.

Column F is the name of the time-to-failure distribution that is associated with the part. There are a number of user-defined distributions (Fixed Value, Uniform, Triangular, 2-and 3-parameter Weibull, Normal, Lognormal, Exponential) to choose from.

Column G is the mode (most likely value) for the failure distribution (only applicable for Uniform, Triangular, Normal, and Lognormal distributions). With the exception of Column H, Columns G-O have distribution units in terms of operational hours (by default) in the EOR/EOM tool. The distribution units can be changed by selecting the cell under the "Reliability" tab after the card has been loaded into the EOR/EOM tool.

Column H is the shape parameter, a specific parameter used in the 2 and 3-parameter Weibull distributions.

Columns J and K are the low and high values of the failure distribution (Triangular distribution only).

Column L is the standard deviation of the distribution (Normal and Lognormal distributions only).

Column M is the location parameter, a specific parameter used in the 2 and 3-parameter Weibull distributions in addition to the Exponential distribution.

Column N is the scale parameter, a specific parameter used in the 2 and 3-parameter Weibull distributions in addition to the Exponential distribution (corresponding to the MTBF of the Exponential distribution).

Column O is the fixed value of the time-to-failure distribution (only applicable if “None” is chosen for the distribution type).

Column P is the *First Failure Date*, the first failure date is the first calendar date where a part failure was observed. This is only used if the time-to-failure distribution is generated from times to failure (see Fig. A.2.5). The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

Column Q is the available card spares that are made available to the particular card for EOR/EOM analysis. This is a card-specific characteristic and as such, only needs to appear in the third row of the spreadsheet.

Column R is "harvestability" of the part. This is indicative of any damage that the part may receive due to the physical action of part harvesting. This value ranges from 0 to 100, where a value of 100 means that the part receives no additional damage due to harvesting and a value of 0 means that the part is non-recoverable (cannot be harvested). It is important to note that this "harvestability" damage has no relation to the remaining relative part life on the part as a result of being fielded. This is only used if the "Allow Harvesting" option is selected in the Solution Control dialog box.

Column S is the field date for the loaded card. This is used if the user loads multiple cards into the tool and some or all loaded cards were fielded on different dates. This is a card-specific characteristic and as such, only needs to appear in the third row of the spreadsheet. The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

Column T is the unrepairable percentage of spare cards per year for the loaded card. This value ranges from 0 to 100, where a value of 100 means that all spare cards become unusable after the first year and a value of 0 means that none of the spare cards for the loaded card become degraded over time.

Column U is the part procurement cost (per part instance) for the loaded card, and is used in system support cost modeling.

*A.3.2 Inventory File Field References*

This file defines the existing inventory of parts, which is shown in Fig. A.3.2.

| | A | B | C |
|---|---|---|---|
| 1 | Inventory 1 | | |
| 2 | 3 | | |
| 3 | Part A | 4 | |
| 4 | Part B | 25 | |
| 5 | Part C | 7 | |

Fig. A.3.2 Sample Inventory CSV file format

The first cell reference (A1) is the name of the loaded inventory.

Cell A2 is the total number of unique parts (rows) located within the inventory. Starting with the third row, each unique part that is identified in the inventory and the total quantity of that part (corresponding Column B) are defined.

*A.3.3 System File Field References*

This file defines the fielded quantity of cards in the system, which is shown in Fig. A.3.3.

| | A | B |
|---|---|---|
| 1 | Sample System 1 | |
| 2 | 1 | |
| 3 | Sample Card 1 | 1 |

Fig. A.3.3 Sample System CSV file format

The first cell reference (A1) is the name of the loaded system.

Cell A2 is the total fielded quantity of unique cards (rows) located within the system. The third row and on describe each unique card that is identified in the system and the total fielded quantity of that card (corresponding Column B) that is located within the system.

### *A.3.4 EOR/EOM Field References*

Upon startup and active use, recall Fig. A.2.1. Upon startup of the tool, the user can select from a number of active tabs that guide the user to the respective panel containing the listed information.

<u>BOM Load & View</u>

Fig. A.3.4 displays the different properties that make up the "BOM Load & View" panel. The drop-down lists (Columns 2 and 3 in the table and the "Choose card" option) allow the user choose from a selection of different attributes. Choosing a different card will refresh the entire panel and display the chosen card's specific characteristics. The BOM table is a representation of how the card is modeled in the EOR/EOM analysis. The user can also select a part's "Reliability" cell and change its time-to-failure characteristics which are then saved to be used in the analysis.

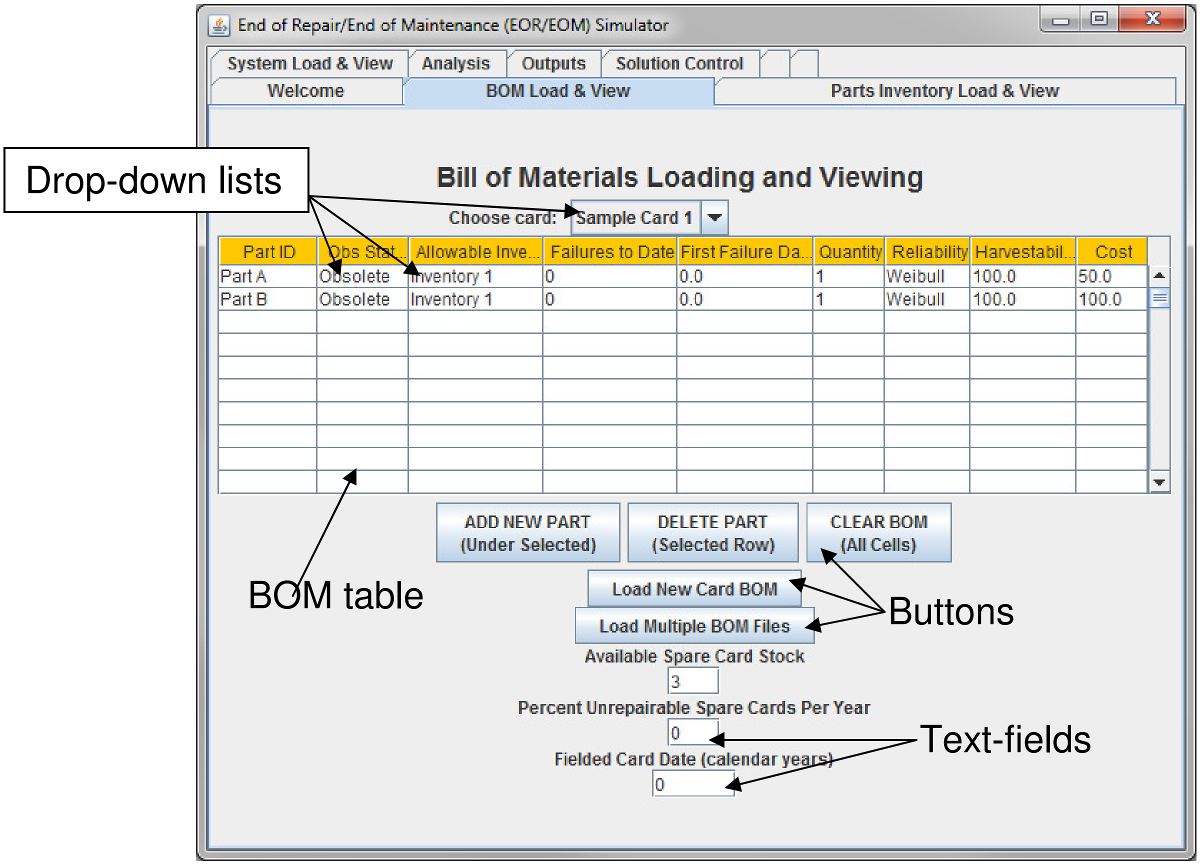



Fig. A.3.4 BOM Panel Field References

The panel also contains the following buttons:

*Add New Part (Under Selected)*: indicates that a part (row) will be added to the BOM table under the currently selected part (row) in the table.

*Delete Part (Selected Row)*: indicates that the currently selected part (row) be deleted from the BOM table.

*Clear BOM (All Cells)*: Clears the contents of the BOM table.

*Load New Card BOM*: Allows the user to load a pre-made BOM CSV file into the current BOM panel (see Fig. A.3.1).

*Load Multiple BOM Files*: Allows the user to load an entire directory of pre-made BOM CSV files. Users should organize all of their BOM, Inventory, and System files into separate individual labeled folders. Open the folder containing ONLY BOM files and select the first file that appears within the folder. All BOM files should fit the layout of the current BOM panel (see Fig. A.3.1).

The panel contains the following text-fields where the user can edit current card properties and they will be saved upon entry:

*Available Spare Card Stock*- defines the number of available spare cards of the currently displayed card.

*Percent Unrepairable Spare Cards Per Year*- defines the percentage of spare cards for the currently selected card that cannot be used towards repair, per year.

*Fielded Card Date*- defines the calendar date that the displayed card was fielded (this text-field only requires information when part failure distributions are derived from past failure history and if there are card-specific field dates--otherwise, the "Starting date for failure data" text-field from the Solution Control tab can be used). The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

<u>Parts Inventory Load & View</u>

This panel (appearing in Fig. A.3.3) contains a table that displays the currently loaded inventories of parts and their quantities of parts within each inventory. Currently, the maximum number of loadable inventories is five, however one can load hundreds of parts within a given inventory. This panel also contains the following buttons:

*Add Column (To Right of Selected)*: indicates that a column will be added to the right of the currently selected column in the table.

*Delete (Selected Column)*: indicates that the currently selected column will be deleted from the table.

*Clear (All Cells)*: Clears the contents of the table.

*Load New Inventory*: Allows the user to load a pre-made inventory CSV file that fits the layout of the current inventory panel (see Fig. A.3.2).

*Load Multiple Inventory Files*: Allows the user to load an entire directory of pre-made inventory CSV files. Users should organize all of their BOM, Inventory, and System files into separate individual labeled folders. Open the folder containing <u>ONLY Inventory files</u> and select the first file that appears within the folder. All Inventory files should fit the layout of the current inventory panel (Fig. A.3.2).

The user can also select a particular cell that displays a part's quantity and a dialog box will appear that enables the user to define a degradation distribution for the selected part.

The user can also select the name of the loaded inventory (in this case "Inventory 1"), and a different dialog box will open, showing possible periodic inspection events that the user can define for the inventory. Here, the user can also load additional inventory manufacturing demands as time progresses during the simulation.

System Load & View

The System Load & View panel is shown in Figure A.2.4. This panel has the following buttons:

*Add Column (To Right of Selected)*: indicates that a column will be added to the right of the currently selected column in the table.

*Delete (Selected Column)*: indicates that the currently selected column will be deleted from the table.

*Clear (All Cells)*: Clears the contents of the table.

*Load New System*: Allows the user to load a pre-made system CSV file to be loaded into the current system panel (see Fig. A.3.3).

*Load Multiple System Files*: Allows the user to load an entire directory of pre-made system CSV files. Users should organize all of their BOM, Inventory, and System files into separate individual labeled folders. Open the folder containing ONLY System files and select the first file that appears within the folder. All System files should fit the layout of the current system panel (see Fig. A.3.3).

Analysis

The Analysis panel is shown in Figure A.2.6. This panel has the following buttons:

*Run*- Begins EOR/EOM simulation analysis of electronic system.

*Pause*- Pause the computation of the analysis.

*Card Index*- Select the card(s) to be included in the analysis (by default, all loaded cards into the tool are included in the analysis).

*Stop and Reset*: Ends and resets the simulation analysis.

The user can also select a number of simulation termination settings for EOR/EOM analysis prior to clicking "Run", and is located on the "Analysis" tab.

*Run to first EOM*: The default termination setting of the EOR/EOM tool. A single simulated life history of the electronic system will be terminated after the occurrence of the first EOM event.

*Run to first fully unsupportable card*: A single simulated life history of the electronic system will be terminated after all instances of a loaded card have been deemed "unsupportable" (the number of failed requests to fulfill a part demand regarding a specific card are equal to its total fielded quantity).

*Run to specific calendar date*: A single simulated life history of the electronic system will be terminated after the specified calendar date has been reached; all EOR and EOM events are recorded until the date of termination.

Outputs

The Outputs panel will appear in different formats depending on conditions that are selected in the Solution Control dialog box. By default, the EOR/EOM tool analysis will track EOR and EOM dates as they occur within the system and display the results in the panel shown in the top of Fig. A.3.5. If the "Individual Card EOM distributions" checkbox in the Solution Control dialog is selected, the results will displayed in the panel shown in the bottom of Fig. A.3.5 and the EOR/EOM tool will track the first EOM dates to occur on each individual card and plot EOM probability distributions specific to loaded cards rather than the order of occurrence within the system. If the "Sacrificing of Selected Cards for Refresh" checkbox in the Solution Control dialog is selected, the simulation termination setting is set to "Run to specific calendar date", and the "Generate refresh plans to end of support date" checkbox is selected, the EOR/EOM tool will track the "just-in-time" completion dates for selected card refreshes required to ensure sustainment of the electronic system to the specific calendar date (see Fig. A.3.5).

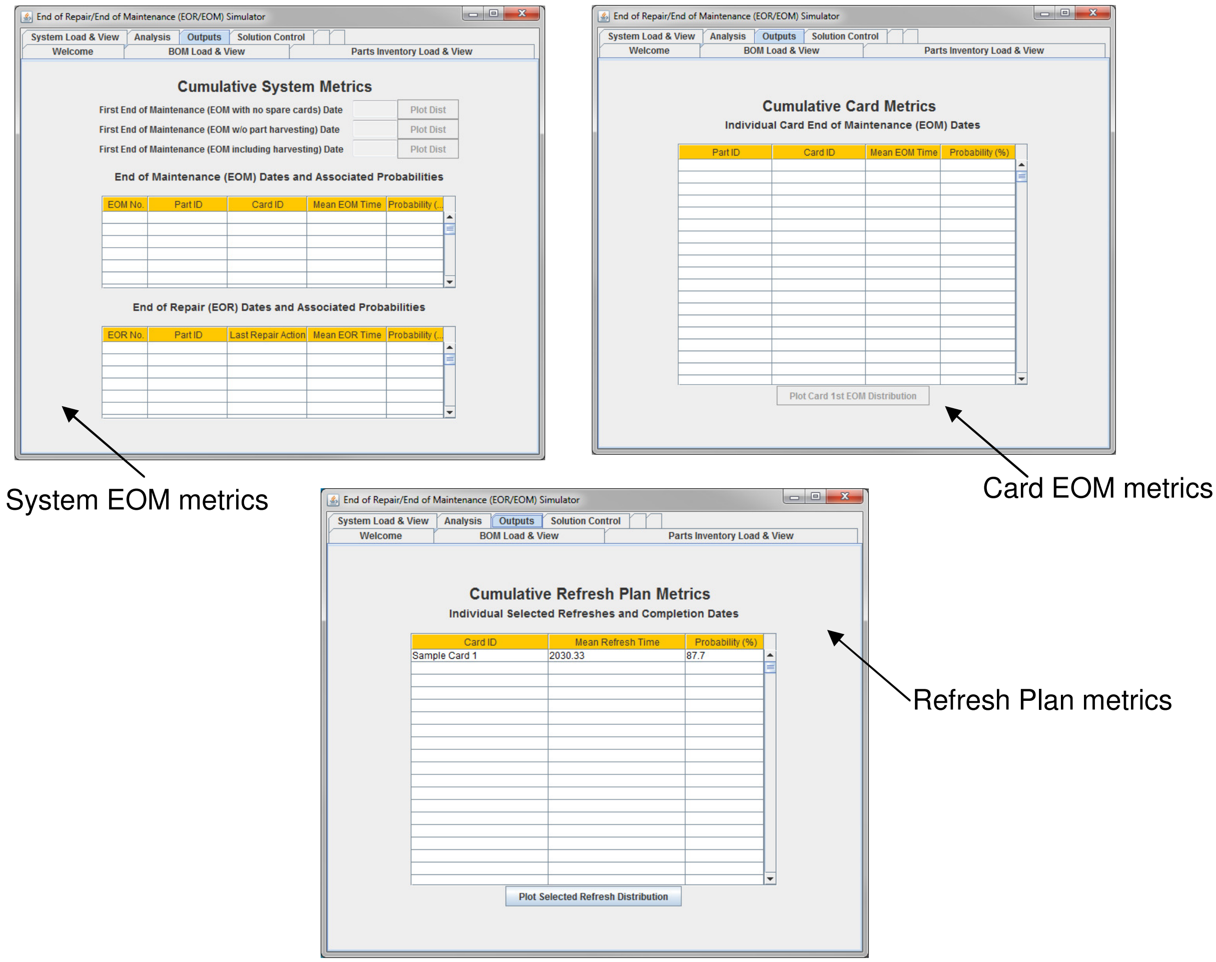



Fig. A.3.5 EOR/EOM outputs panels

<u>Solution Control</u>

There are three text-fields that the user can input information into the Solution Control panel (Fig. A.2.5).

*Starting date for failure data*: Only used for when part failure distributions will be derived from past part failure history. This field requires the user to enter the calendar date when part failures began to be observed. This field should be overridden for specific cards that were not fielded on the same calendar date as other cards (see *Card Field Date* from "BOM Load & View" reference page). The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

*Analysis Start Date*: The beginning calendar date of the EOR/EOM simulation. It is assumed that all part failure observances have been recorded up until this date (unless otherwise specified by specific “Card Fielded Dates”. The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

*Operational hours per year*: This field is used only if part failure distributions are in terms of operational hours. This also assumes that all cards and parts are on the same operational schedule (assumes all cards and parts are operational for the same amount of time each year).

*Solution Control*: Opens up the Solution Control dialog box where various settings may be selected for EOR/EOM analysis.

*Save Field States*: Saves the current EOR/EOM file state.

*Load*: Loads a previously saved EOR/EOM file.

*Restore Inventories*: This option is only available if the user has previously run the simulation with the "Inventory Update" checkbox selected under the solution control dialog box; this refreshes the inventory table located under the "Parts Inventory Load & View" tab.

<u>Solution Control dialog box</u>

The Solution Control dialog box is shown on the left side of Fig. A.2.5. This dialog box contains the following fields:

*Monte Carlo?*: Run repeated random samplings of the analysis, this should always be selected as "Yes".

*Number of Monte Carlo Samples*: The user can input the number of monte carlo samples run for analysis. As with any repeated random sampling algorithm, as the number of samples increase, the more accurate the system and card metrics will be based upon system inputs and random sampling of part failure distributions, but the run time will also increase.

*Pause Between Time Steps*: User can define the pause between time steps within a given life history run, this should be "0", unless Inventory Update is selected and a single simulation is being conducted.

*Turn On Inventory Update During Analysis*: If selected, this option turns on inventory update on the "Parts Inventory Load & View" panel which highlights important actions taking place during the system's life history including:

- Replacement of failed parts with new parts from inventory
- Harvesting existing cards and placing them into a separate harvested inventory
- Degradation of parts in inventory

- Replacement of failed parts with harvested parts from harvested inventory

*Assume Immediate First Failure for Parts with no Failure History*: If selected, the analysis assumes the worst case for loaded parts with no failures to date (assumes the fail right before the analysis date) and synthesizes their part failure distributions off the single failure.

*Synthesize ALL failure distributions from times to failure*: If selected, the analysis synthesizes all part failure distributions from past failure histories (failures to date, fielded start date, and first failure date) and fits it to a Uniform distribution.

*Allow Harvesting*: If selected, the analysis incorporates the action of harvesting existing parts that have not failed off an existing card that is swapped out and replaced with an available spare (occurs when a demand for a part on a particular card cannot replaced with a new part and there are available spare cards). After part and card spares have been exhausted for fielded failing parts, the harvested inventory of parts is accessed to increase the time until end of maintenance occurs.

*Allow Sacrificing of Selected Cards for Refresh*: If selected, the analysis incorporates the action of sacrificing selected cards for a design refresh. Technology or design refreshes are used in the replacement of one or more obsolete parts with non-obsolete parts in order to keep the system sustainable. This option can be used towards:

1) *Selective design refreshes of cards and their refresh completion dates prior to analysis to EOM*
2) *Construction of a design refresh plan to ensure system sustainment to a specific date (if the specified termination setting is pre-selected).*

*Include Costs*: Enables system support cost modeling of the electronic system. An additional tab will be generated if this checkbox is selected.

*Model Costs using Distributions*: Enables input costs to be entered as user-defined distributions rather than as fixed values.

*Enable Card Clumping/How many cards will be clumped into 1*: Used for complex and large systems in an aim to effectively 'clump' populations of specific fielded quantities of cards together for dynamic memory allocation and tool computation efficiency.

*Plot First EOM Distributions for individual cards*: Changes the "Outputs" panel and how the EOR/EOM results are displayed and categorized (probability EOM distributions for individual cards are calculated rather than system EOM distributions).

*Plot time histories of system support loss versus time*: Allows the user to plot time histories of selected cards in the loaded system to observe their corresponding measures of unsupportability over time.

### *A.3.5 EOR/EOM Cost Model Field References*

The following field references are for the cost models used in the EOR/EOM tool. The one cost metric that is not included on the cost analysis inputs tab (see Fig. A.3.6) is the inventory holding cost associated with each loaded inventory. The inventory holding costs ($ per part per year) can be changed by selecting the name of the loaded inventory column (see Fig. A.2.3).

*Administrative cost of a draw* ($ per draw): A cost that is accumulated for every maintenance or inspection event that occurs during EOR/EOM analysis.

*Value added cost of a draw* ($ per draw): A cost that is accumulated for every replacement event that occurs during EOR/EOM analysis (i.e., replacing parts from inventory, using spare cards or harvested parts).

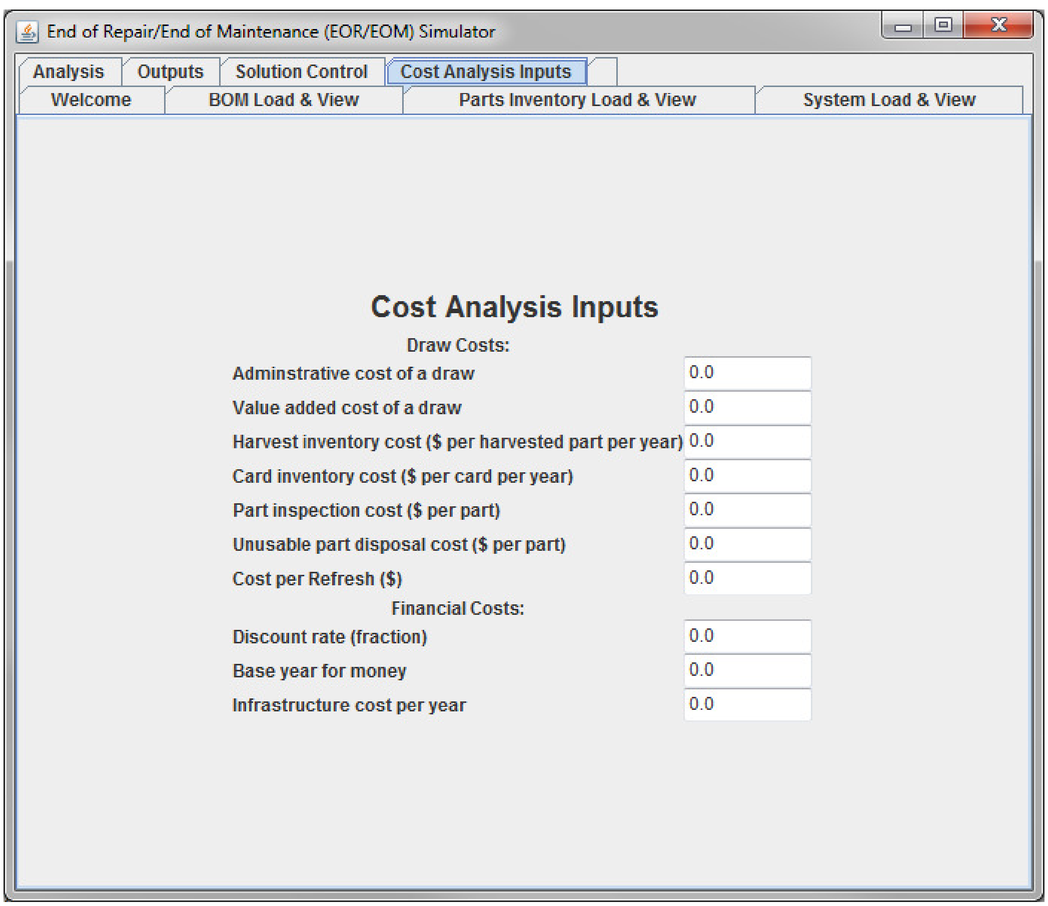


Fig. A.3.6 Cost Analysis Inputs

*Harvest inventory cost* ($ per harvested part per year): A cost that is accumulated for holding a harvested part during EOR/EOM analysis.

*Card inventory cost* ($ per card per year): A cost that is accumulated for holding a spare card during EOR/EOM analysis.

*Part inspection cost* ($ per part): A cost that is accumulated for each inspected part during periodic inspections during EOR/EOM analysis.

*Unusable part disposal cost* ($ per part): A cost that is accumulated for each part disposed of during EOR/EOM analysis.

*Cost per refresh* ($ per refresh): A cost that is accumulated for each completed refresh of a selected card during EOR/EOM analysis.

*Discount rate* (fraction): Discount rate on money per year. A parameter used in determining the net present value (NPV) of costs accumulated during EOR/EOM analysis.

*Base year for money*: A parameter used in determining the net present value (NPV) of costs accumulated during EOR/EOM analysis. The format for the date is represented as a "####.##". For example, May 2011 would be represented by 2011.4.

*Infrastructure cost per year*: A cost that is accumulated for each calendar year during EOR/EOM analysis.

## A.4 EOR/EOM Simulation Outputs

The EOR/EOM tool provides three kinds of outputs:

1. Cumulative System Metrics
2. Individual Card Metrics
3. Time-history plots
4. Cost Metrics
5. Design Refresh Metrics
6. Output Data Files

*A.4.1 Cumulative System Metrics*

Fig. A.4.1 shows the results for a particular systems' cumulative metrics. The first mean EOM date that occurs for the system seen in Fig. A.4.1 is in the year 2018.22, not including the available spare card inventories the system can draw from. If the system draws from available spare card inventories, the first mean system EOM date occurs approximately 11 years later, in other words, the system is able to be sustained for an additional 11 years. The first system EOM date can occur in a variety of different situations (seen in the EOM table from Fig. A.4.1) and the identification of the parts that caused a loss of systems operations and their corresponding likelihoods are also presented.

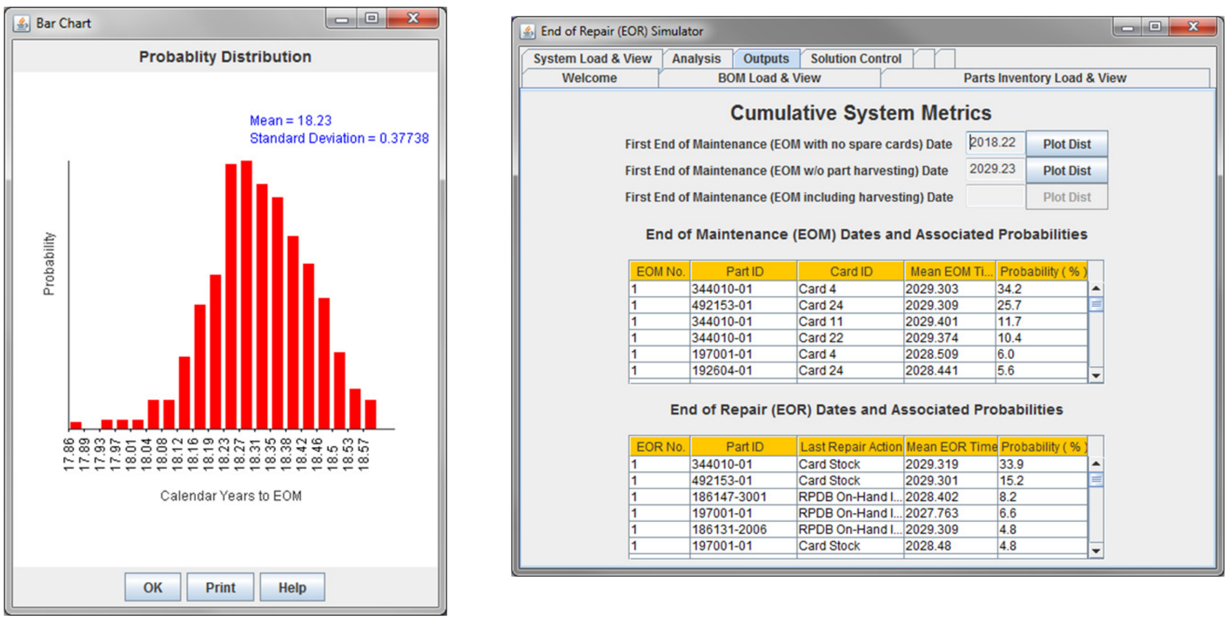
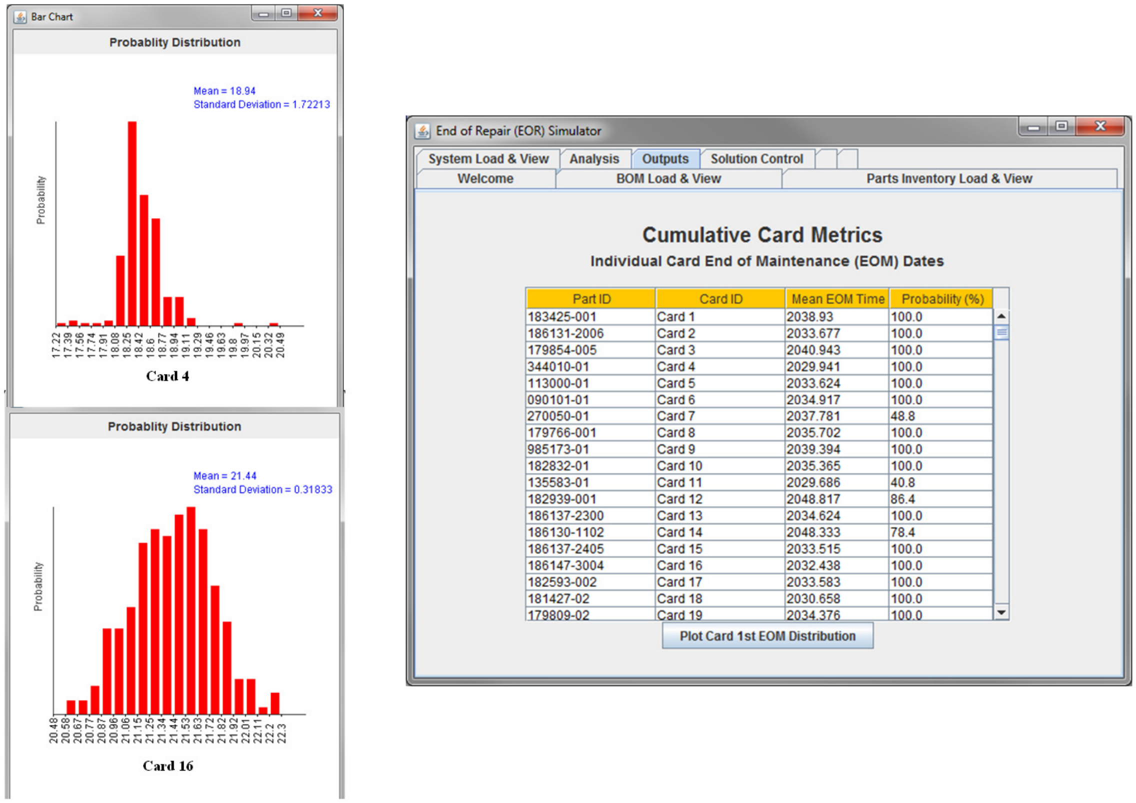



Fig. A.4.1 Cumulative System Metrics

*A.4.2 Individual Card Metrics*

The EOR/EOM tool also can display its results in terms of individual card metrics (as seen in Fig. A.4.2) depending on the pre-selected conditions from the Solution Control dialog box (Fig. A.2.5).




| Part ID | Card ID | Mean EOM Time | Probability (%) |
|---|---|---|---|
| 183425-001 | Card 1 | 2038.93 | 100.0 |
| 186131-2006 | Card 2 | 2033.677 | 100.0 |
| 179854-005 | Card 3 | 2040.943 | 100.0 |
| 344010-01 | Card 4 | 2029.941 | 100.0 |
| 113000-01 | Card 5 | 2033.624 | 100.0 |
| 090101-01 | Card 6 | 2034.917 | 100.0 |
| 270050-01 | Card 7 | 2037.781 | 48.8 |
| 179766-001 | Card 8 | 2035.702 | 100.0 |
| 985173-01 | Card 9 | 2039.394 | 100.0 |
| 182832-01 | Card 10 | 2035.365 | 100.0 |
| 135583-01 | Card 11 | 2029.686 | 40.8 |
| 182939-001 | Card 12 | 2048.817 | 86.4 |
| 186137-2300 | Card 13 | 2034.624 | 100.0 |
| 186130-1102 | Card 14 | 2048.333 | 78.4 |
| 186137-2405 | Card 15 | 2033.515 | 100.0 |
| 186147-3004 | Card 16 | 2032.438 | 100.0 |
| 182593-002 | Card 17 | 2033.583 | 100.0 |
| 181427-02 | Card 18 | 2030.658 | 100.0 |
| 179809-02 | Card 19 | 2034.376 | 100.0 |

Fig. A.4.2 Individual Card Metrics

Here a list of different loaded cards found to have first EOM dates when running the EOR/EOM tool to a specified calendar date (2050.0). The part identifications that caused the particular card to cause an EOM are displayed in addition to the likelihoods that the corresponding card encountered a first EOM throughout the total number of system life histories that were analyzed. Card-specific first EOM probability distributions can also be constructed a displayed and provide the user with statistical interpretations of end of maintenance events at a card-specific level.

*A.4.3 Time-history plots*

The third simulation output are time history plots that provide the user with a graphical interpretation of system loss over time. Fig. A.4.3 displays a system where four cards became *unsupportable* (an instance of End of Maintenance--where a particular part demand could not be met for a particular card) and how the total percentage of those fielded cards became unsupportable over time.

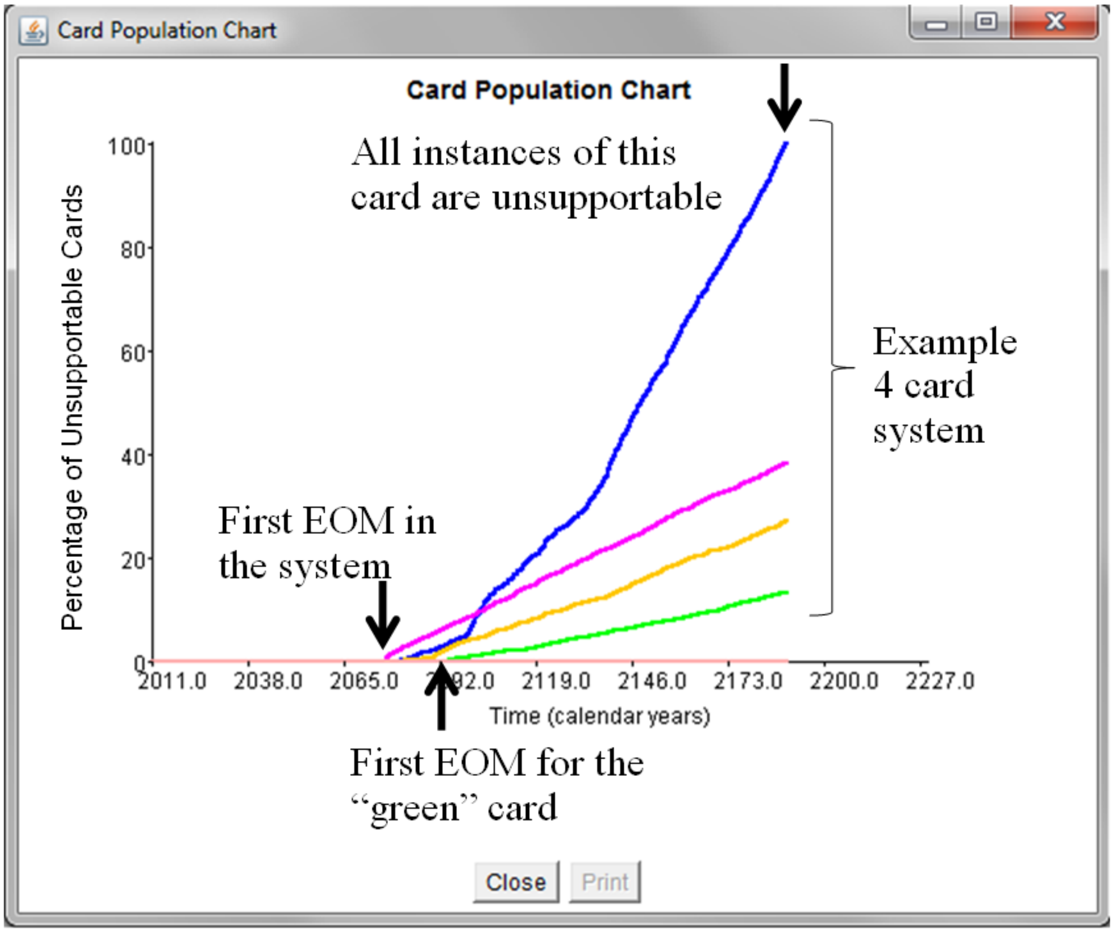



Fig. A.4.3 Time history plot (4 card system)

This provides the user with an understanding of the "loss rate of system operations" at the card-level due to End of Maintenance events occurring and how that rate increases with time. This loss rate will increase over time due to an increase in different parts on the cards causing EOM events as part sparing becomes extinguished.

*A.4.4 System Support Cost Metrics*

The system support cost metrics of the electronic system are provided (seen in Fig. A.4.4) once the simulation has concluded. The cumulative cost metrics (top left) display the total support cost over the system support lifetime. The inventory holding costs (top right) detail how the holding costs are sub-divided between the three possible inventories: spare parts, spare cards, and harvested parts. The cumulative cost metrics (bottom left) show the average sub-costs and total cost over the total number of simulations. A probability distribution of the total costs (bottom right) can also be produced for statistical interpretation.

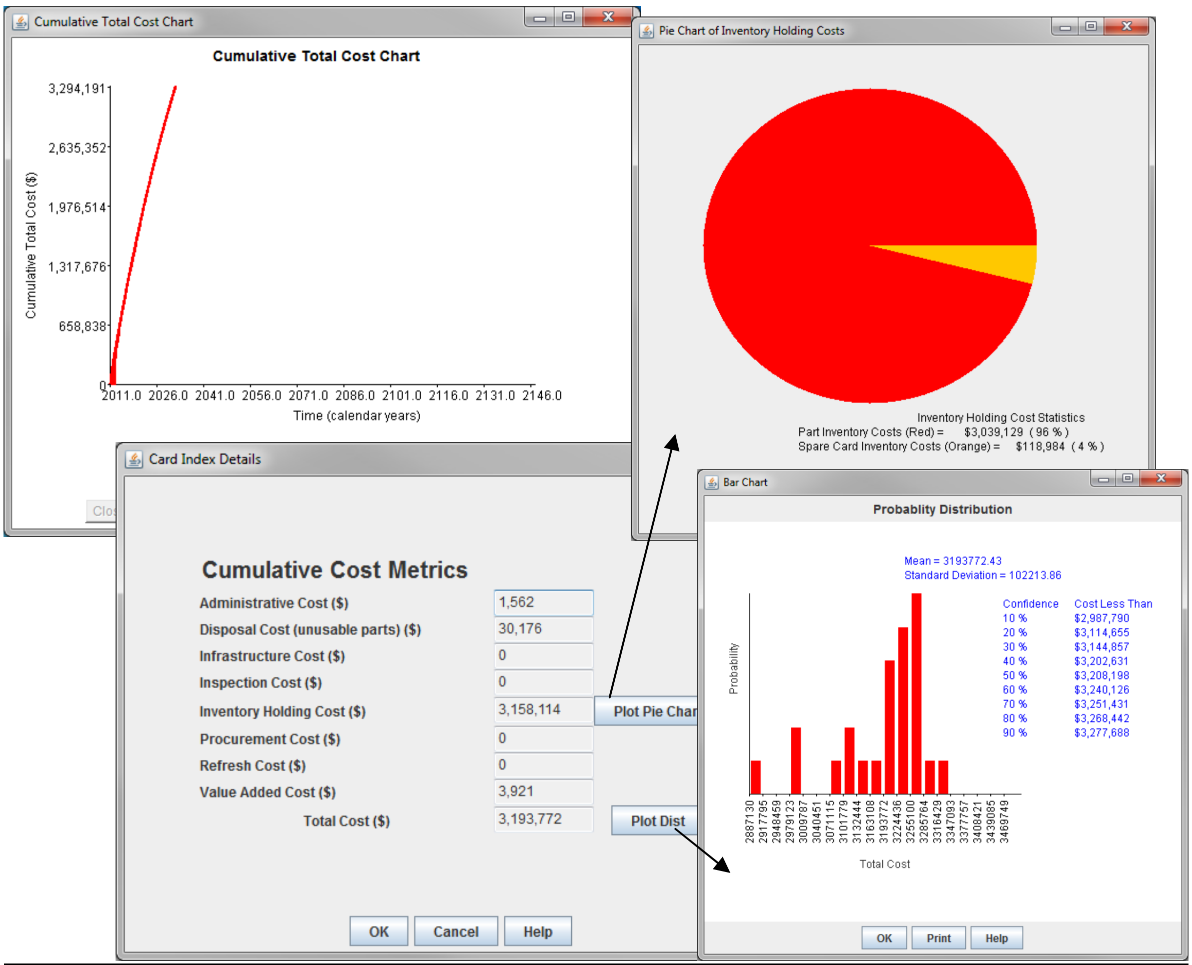


Fig. A.4.4 Cost Metrics

### *A.4.5 Design Refresh Plan Metrics*

The EOR/EOM tool also can display its results in terms of a design refresh plan(as seen in Fig. A.3.5). This assumes necessary design refreshes are completed "just-in-time", on the date which they are required (EOM date). The design refresh metrics are ordered by selected cards that require "just-in-time" refresh and the user can construct probability distributions of the completed refresh dates for selected cards.

### *A.4.6 Output Data Files*

The EOR/EOM tool automatically outputs data files containing the information gathered from the simulated analysis (depending on the type of simulation). The output data file contains consecutive EOR and EOM information, Design Refresh Completion Date information (exclusively for DRP analysis), and Total cost information across all simulated life histories of the system. This output data file is named "Metrics@Date Time" and is located in the same directory as the EOR/EOM and LTB application.

## A.5 LTB Tutorial

This tutorial demonstration includes running through LTB analysis, saving an LTB file, and loading a LTB file. This tutorial assumes that the user is running an application version of the tool and that the user has the minimum JRE (Java Runtime Environment) installed on their machine. This tutorial also assumes the user is running the CALCE LTB software on a PC, no attempt has been made to adapt the tool's functionality for performance on other platforms.

### *A.5.1 Running the LTB Application*

1) Start the EOR/EOM and LTB application. At the "Choose Analysis Type" dialog box, choose "Lifetime Buy" and Select "OK". You should obtain an interface like the one shown in Fig. A.5.1.

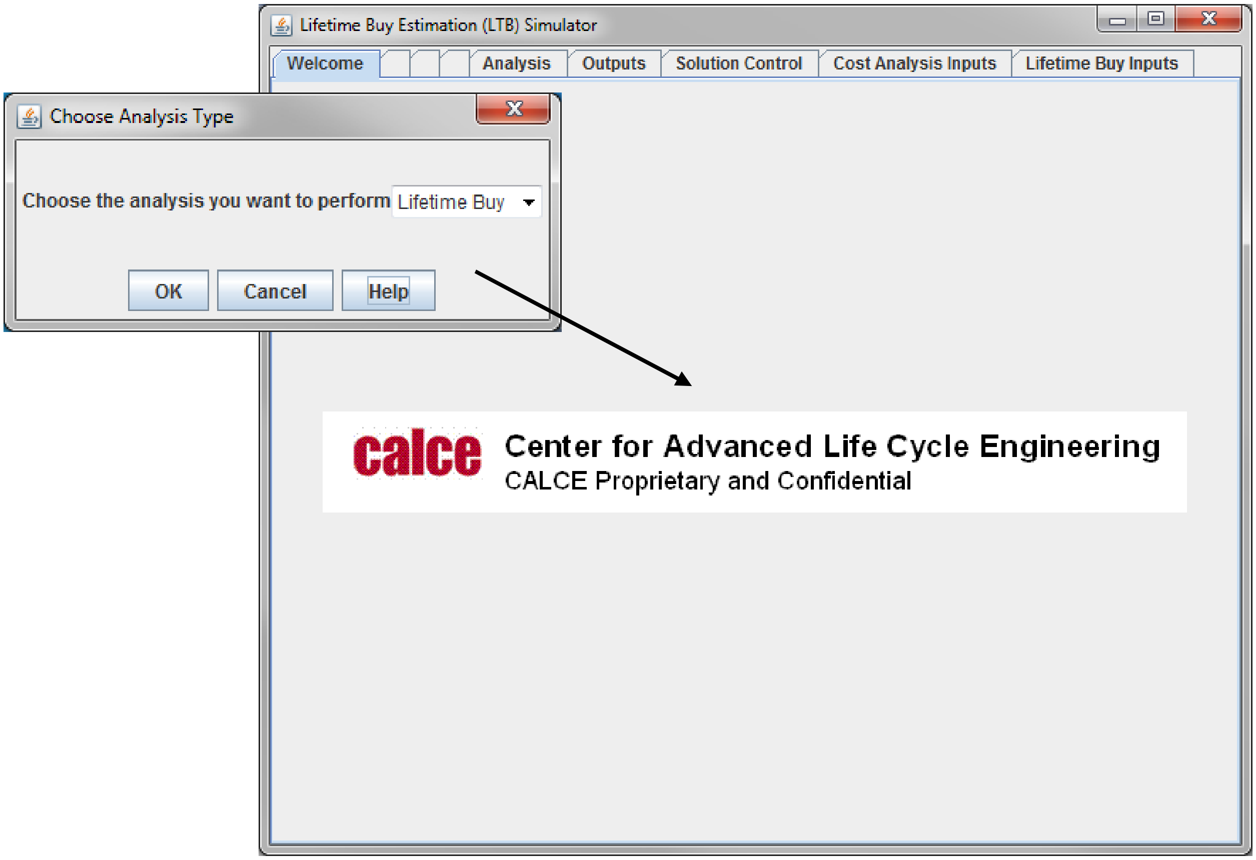



Fig. A.5.1 Initial startup of LTB tool

2) Select the "Solution Control" tab and enter in "2019.0" for the End of support date, "2011.0" for the Analysis start date, and "8760.0" for the Operational hours per year. You should obtain an interface like the one shown in Fig. A.5.2.

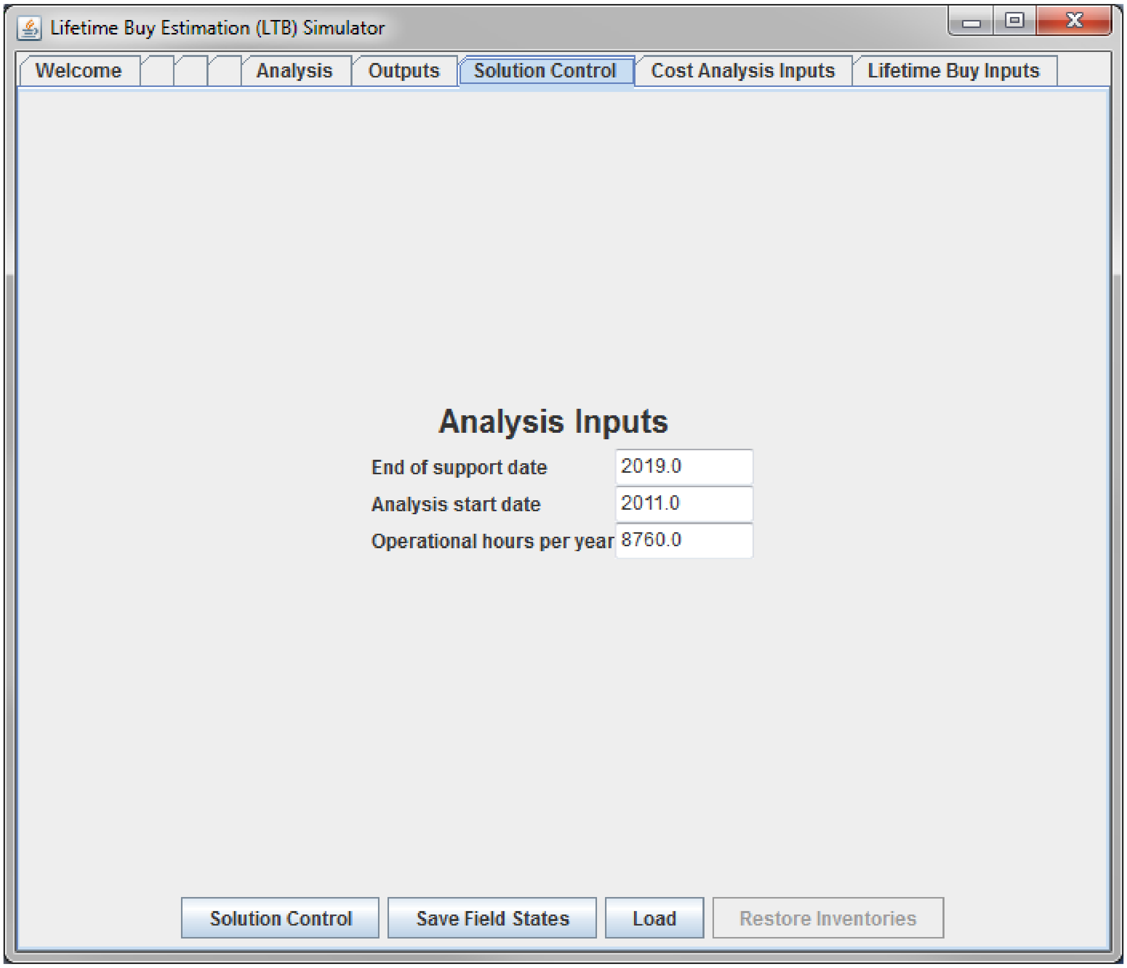



Fig. A.5.2 Solution Control Inputs

3) Click on the "Lifetime Buy Inputs" tab and enter the information shown in Fig. A.5.3. Then click on the "Cost Analysis Inputs" tab and enter the information shown in Fig. A.5.4. The user can also add manufacturing demand or retirement schedules to the electronic system.

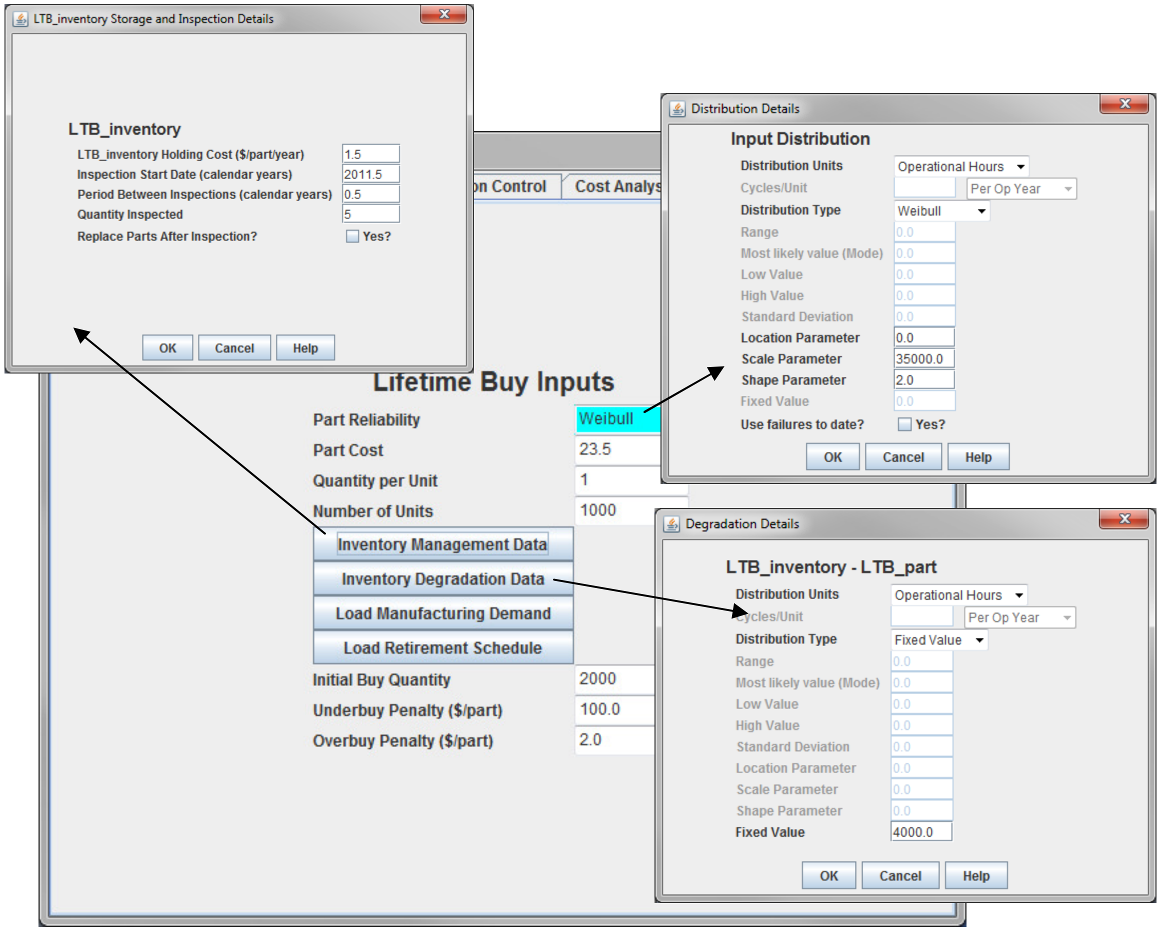



Fig. A.5.3 Lifetime Buy Inputs

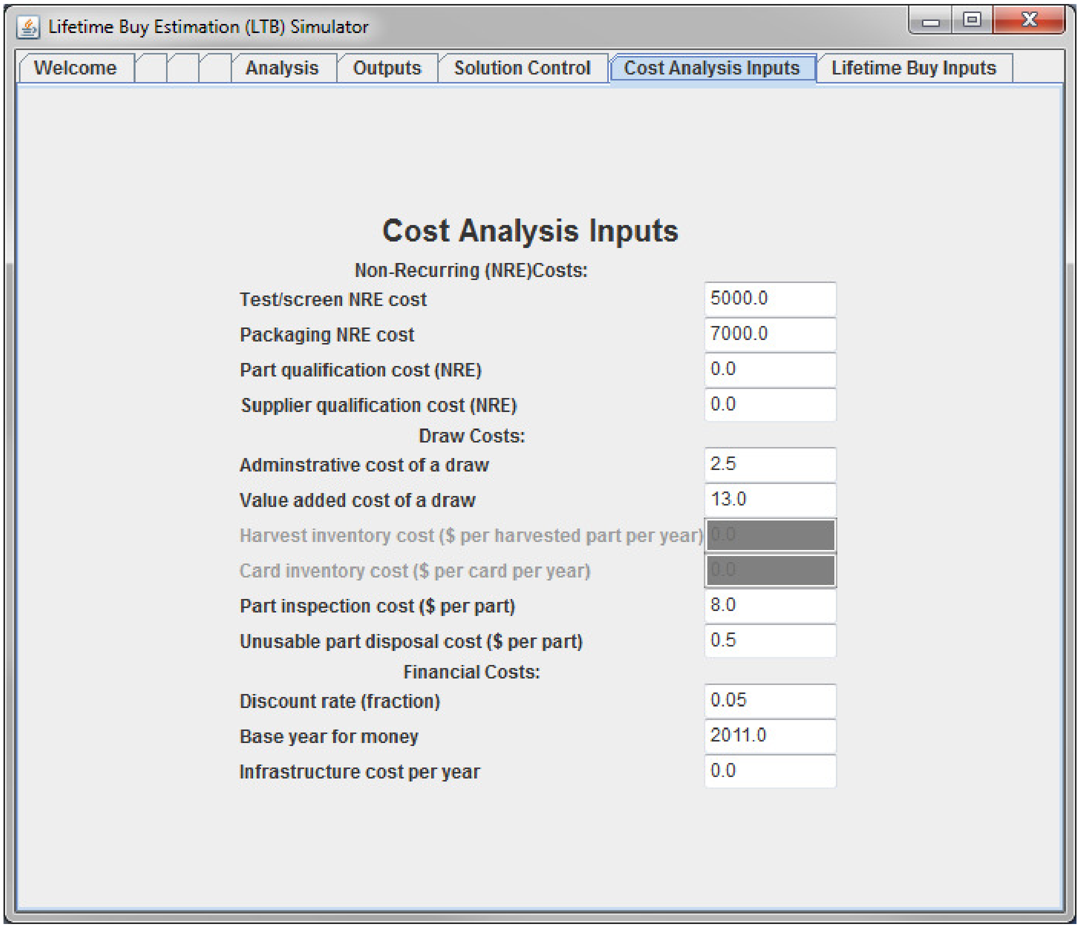


Fig. A.5.4 Cost Analysis Inputs

4) Click on the "Analysis" tab and click the "Run" button. The tool is now conducting 1,000 simulated life histories of the system to determine the total number of demands and associated life-cycle costs for the lifetime buy quantity (referred to as the initial buy quantity) selected in Fig. A.5.3.
5) After the analysis is completed, your screen should appear similar to that of Fig. A.5.5.

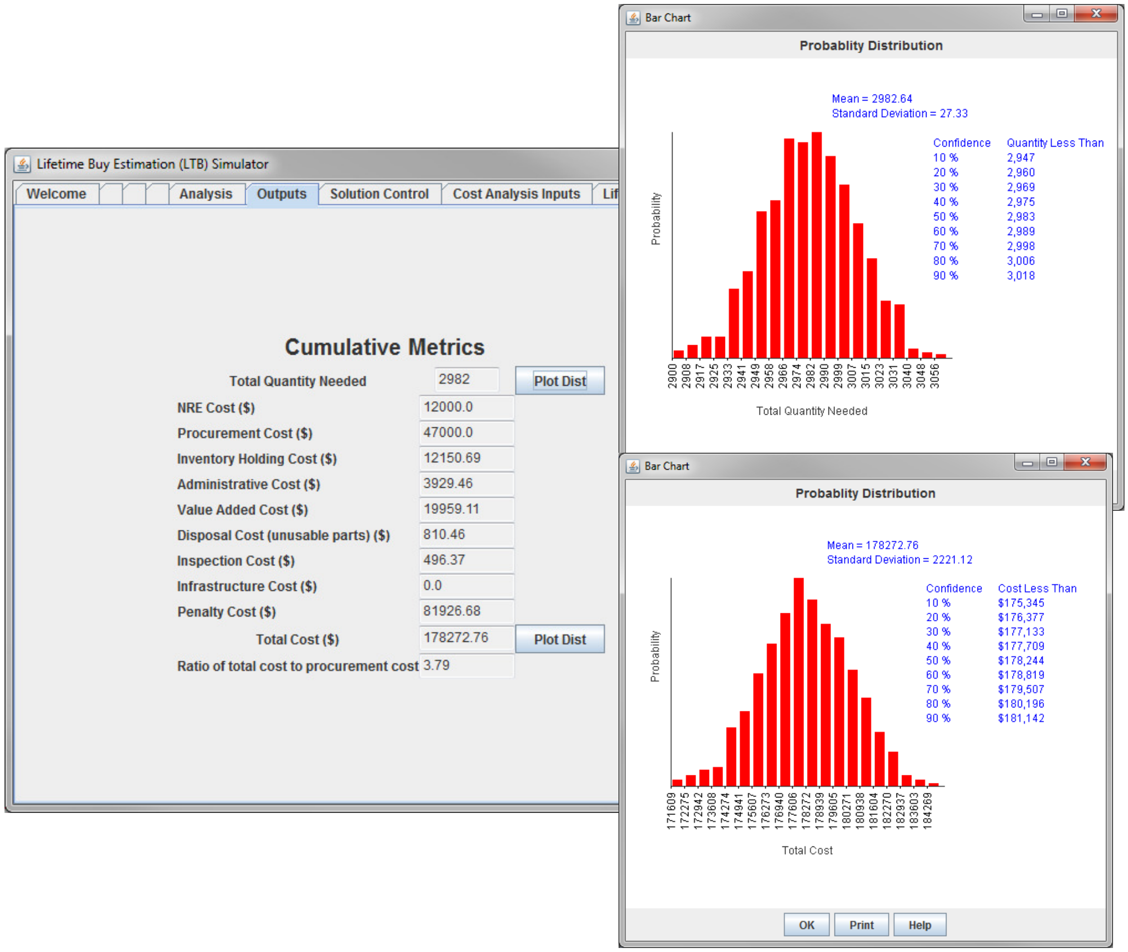



Fig. A.5.5 Cumulative LTB Tutorial Metrics

### *A.5.2 Saving a LTB File*

After the user has input all lifetime buy and cost information, the LTB system file may be saved. Select the "Solution Control" tab and then click on "Save Field States". Name the file "lifetime_1", choose a desired saved location, and click "Save". The LTB file has now been successfully saved with all the loaded system characteristics to the desired location on your machine.

### *A.5.3 Loading a LTB File*

After your LTB file has been saved to a desired location, exit the tool by selecting the red "X" in the top-right corner of the interface. Once the tool has been closed, re-open the CALCE EOR/EOM application. After selecting "Lifetime Buy", you should see a screen that looks like Fig. A.5.1. Select the "Solution Control" tab and click on the "Load" button. Locate and select the file labeled "lifetime _1" and click “Open”. After the tool is done loading, the loaded application should represent the saved application prior to the last save state.

## A.6 LTB Field and Button Reference

This section documents the buttons and fields within the LTB tool. A majority of the fields and buttons within the LTB tool are present within the EOR/EOM tool—this section will only cover those fields and buttons unique to the LTB simulator (see Section 3 for other references).

### *A.6.1 LTB Field References*

The first two references are from the "Solution Control" and "Cost Analysis Inputs" tabs.

*End of support date*: The specified date through which all fielded units of the electronic system must be sustained.

*Test/screen, Packaging, Part qualification, supplier qualification (NRE) costs*: Specific non-recurring costs charged on the analysis start date.

The remaining references are from the "Lifetime Buy Inputs" tab.

*Part Reliability*: User can select the time-to-failure distribution type and distribution parameters for the fielded parts.

*Part Cost ($/part)*: The procurement price of a single part at the time/purchase of the lifetime buy quantity.

*Quantity per Unit*: The quantity of parts per unit (referred to as a "card" in the EOR/EOM tool).

*Number of Units*: The quantity of units that comprise the system.

*Initial Buy Quantity*: Also referred to as the lifetime buy quantity. the quantity of spares purchased at the lifetime buy.

*Underbuy Penalty ($/part)*: The penalty when there are not enough spares to meet the demands (this penalty is included to the procurement price).

*Overbuy Penalty ($/part)*: The penalty when there are a surplus of spares (the procurement price is not recovered).

*A.6.2 LTB Manufacturing and Retirement Schedule File Formats*

The user can also import manufacturing demand or retirement schedules into the LTB simulator for additional events.
The next file defines a schedule of additional manufacturing demands for the electronic system, which is shown in Fig. A.6.1.

| A | B | C |
|---|---|---|
| Inventory 1 | | |
| 1 | Calendar Year | Quantity Manufactured |
| Demand 1 | 2011.00856 | 10 |
| Demand 2 | 2011.75 | 3 |
| Demand 3 | 2012.05 | 8 |
| Demand 4 | 2012.65 | 14 |
| Demand 5 | 2013.05 | 22 |

Fig. A.6.1 Manufacturing Demand File

The first cell reference (A1) is the name of the loaded inventory.

Cell A2 is the total number of individual manufacturing demand events that will be loaded into the tool (note, in this example there are five separate demands; however, only the first one will be loaded into the tool). NOTE: It is assumed that manufacturing demands are chronologically ordered. Cell B2 represents the calendar date (in terms of calendar years) the manufacturing demand will be produced. Cell C2 represents the quantity that will be manufactured at the manufacturing date (B2). Starting with the third row, each manufacturing demand is identified by its corresponding date (Column B) and corresponding quantity (Column C) to be manufactured.

The next file defines a retirement schedule for the electronic system, which is shown in Fig. A.6.2.

| A | B | C |
|---|---|---|
| Inventory 1 | | |
| 1 | Calendar Year | Quantity Retired |
| Demand 1 | 2011.5 | 30 |
| Demand 2 | 2011.75 | 3 |
| Demand 3 | 2012.05 | 8 |
| Demand 4 | 2012.65 | 14 |
| Demand 5 | 2013.05 | 22 |

Fig. A.6.2 Retirement Schedule File

The first cell reference (A1) is the name of the loaded inventory.

Cell A2 is the total number of individual retirement schedule events that will be loaded into the tool (note, in this example there are five separate demands; however, only the first one will be loaded into the tool). NOTE: It is assumed that retirement demands are chronologically ordered. Cell B2 represents the calendar date (in terms of calendar years) the retirement demand will be removed. Cell C2 represents the quantity that will be retired at the retirement date (B2). Starting with the third row, each retirement demand is identified by its corresponding date (Column B) and corresponding quantity (Column C) to be retired.

## A.7 LTB Simulation Outputs

The LTB simulation outputs generated probability distributions of the total quantity needed and total life-cycle cost associated with the lifetime buy quantity (see Fig. A.5.5). The user can then perform subsequent simulations with varying estimations of the lifetime buy quantity in order to determine the optimum lifetime buy quantity that minimizes the total life-cycle cost associated with the lifetime buy quantity.

# References


1. Alford, L., “The Problem with Aviation COTS,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 16, No. 2, pp. 33-37, February 2001.

2. Hall, J. and R. Naff, “The Cost of COTS,” *IEEE Aerospace and Electronic Systems Magazine*, Vol. 16, No. 8, pp. 20-24, August 2001.

3. Sandborn, P., V. Prabhakar, and O. Ahmad, "Forecasting electronic part procurement lifetimes to enable the management of DMSMS obsolescence," *Microelectronics Reliability*, Vol. 51, No. 2, pp. 392-399, February 2011.

4. Song, Y. and H. Lau, "A periodic review inventory model with application to the continuous review obsolescence problem," *European Journal of Operations Research*, Vol. 159, No. 1, pp. 110-120, November 2004.

5. Adams, C., “Getting a Handle on COTS Obsolescence,” *Avionics Magazine,* 1 May 2005.

6. Sandborn, P., and J. Myers, “Designing Engineering Systems for Sustainment,” *Handbook of Performability Engineering*, ed. K.B. Misra, Springer, pp. 81-103, London, 2008.

7. Tomczykowski, W. J., “A Study on Component Obsolescence Mitigation Strategies and Their Impact on R & M,” *Annual Reliability and Maintainability Symposium*, pp. 332-338, January 2003.

8. Blischke, W. R., and D. N. P. Murthy, "Mathematical Techniques for Warranty Analysis,” *Product Warranty Handbook*. New York: M. Dekker, 1996. Print.

9. Soland, Richard M., “A Renewal Theoretic Approach to the Estimation of Future Demand for Replacement Parts,” *Operations Research*, Vol. 16, No. 1, pp. 36-51, January 1968.

10. Croston, J.D., "Forecasting and stock control for intermittent demands," *Operational Research Quarterly*, Vol. 23, No. 3, pp. 289-303, 1972.

11. Willemain, T., C. Smart, and H. Schwarz, "A new approach to forecasting intermittent demand for service parts inventories," *International Journal of Forecasting*, Vol. 20, No. 3, pp. 375-387, 2004.

12. Eaves, A.H.C. and B. Kingsman, "Forecasting for the ordering and stock-holding of spare parts," *Journal of the Operational Research Society*, Vol 55, pp. 431-437, 2004.

13. Regattieri, A., M. Gameri, R. Gamerini, and R. Manzini, "Managing lumpy demand for aircraft service parts," *Journal of Air Transport Management*, Vol. 11, No. 6, pp. 426-431, 2005.

14. Kennedy, W., J. Patterson, and L. Fredendall, "An overview of recent literature on spare parts inventories," *International Journal of Production Economics*, Vol. 76, No. 2, pp. 201-215, 2002.

15. Goyal, S.K. and B. Giri, "Recent trends in modeling of deteriorating inventory," *European Journal of Operational Research*, Vol. 134, No. 1, pp. 1-16, 2001.

16. Brown G., J. Lu, and R. Wolfson, "Dynamic modeling of inventories subject to obsolescence," *Management Science*, Vol. 11, No. 1., pp. 51-63, 1984.

17. Hong, J. S., Koo, H.-Y., Lee, C.-S., and J. Ahn, “Forecasting Service Parts Demand for a Discontinued Product,” *IIE Transactions*, Vol. 40, No. 7, pp. 640-649, April 2008.

18. Eppen, G. D., & Martin, R. K., "Determining Safety Stock in the Presence of Stochastic Lead Time and Demand," *Management Science*, Vol. 34, No. 11, pp. 1380-1390, November 1988.

19. Hua, Z.S., B. Zhang, J. Yang, and D.S. Tan, "A New Approach of Forecasting Intermittent Demand for Spare Parts Inventories in the Process Industries," *Journal of the Operational Research Society*, Vol. 58, No. 1, pp. 52-61, 2007.

20. Teunter, R., A. Syntetos, and M. Babai, "Intermittent demand: Linking forecasting to inventory obsolescence, " *European Journal of Operational Research*, Vol. 214, No. 3, pp. 606-615, November 2011.

21. Prowell, S.J. and J.H. Poore, "Computing system reliability using Markov chain usage models," *The Journal of Systems and Software*, Vol. 73, pp. 219-225, 2004.

22. Kounev, S., "Performance Modeling and Evaluation of Distributed Component-Based Systems Using Queueing Petri Nets," *IEEE Transactions on Software Engineering*, Vol. 32, No. 7, pp. 486-502, July 2006.

23. Van der Aalst, W. and K. van Hee, "Business process redesign: A Petri-net based approach," *Computers in Industry*, Vol. 29, No.1-2, pp.15-26, July 1996.

24. Peterson, J., "Petri Nets," *ACM Computing Surveys*, Vol. 9, No. 3, pp. 223-252, Sept. 1977.

25. Hagmark P-E and H Pernu, "Risk evaluation of a spare part stock by stochastic simulation," *ESREL* Estoril, Portugal, pp. 18–22 September 2006.

26. Siprelle, A., D. Parsons, and R. Clark, "Benefits of using a supply chain simulation tool to study inventory allocation," *Simulation Conference Proceedings*, Vol. 1, pp. 238-245, 2003.

27. Marsan, M., "Stochastic petri nets: an elementary introduction," *Advances in Petri Nets* pp. 1–29, 1989.

28. Kumaraswarny, S. and E. Sankarasubramanian, "A Continuous Review of (S-s) Inventory Systems in which Depletion is Due to Demand and Failure of Units," *The Journal of Operational Research Society*, Vol. 32, No. 11, pp. 997-1001, 1981.

29. Pierskalla, W., "Inventory Depletion Management with Stochastic Field Life Functions," *Management Science*, Vol. 13, No. 11, pp.877-886, 1967.

30. Nahmias, S., "Inventory Depletion Management When the Field Life is Random," *Management Science*, Vol. 20, No.9, pp.1276-1283, 1974.

31. Song, J-S. and P. Zipkin, "Managing Inventory with the Prospect of Obsolescence," *Operations Research*, Vol. 44, No. 1, pp. 215-222, 1996.

32. Cobbaert, K. and D. van Oudheusden, "Inventory models for fast moving spare parts subject to 'sudden death' obsoelscence," *International Journal of Production Economics*, Vol. 44, No. 3, pp. 239-248, 1996.

33. Song, Y. and H. Lau, "A periodic-review inventory model with application to the continuous-review obsolescence problem," *European Journal of Operational Research*, Vol. 159, No. 1, pp. 110-120, 2004.

34. Lee, L., H. Huang, C. Lee, W. Jaruphongsa, Y. Yong, Z. Liang, C. Leong, Y. Tan, K. Namburi, E. Johnson, and J. Banks, "Discrete Event Simulation Model for Airline Operations: SIMAIR," *Simulation Conference Proceedings*, Vol. 2, pp. 1656-1662, December 2003.

35. Warrington, L., J. Jones, and N. Davis, "Modelling of maintenance, within discrete event simulation," *Reliability and Maintainability Symposium Proceedings*, pp. 260-265, 2002.

36. Shaffer, G. and G. McPherson, "FAA COTS Risk Mitigation Guide: Practical Methods for Effective COTS Acquisition and Lifecycle Support Revision 3.2," *Federal Aviation Administration*, 2010, http://www.faa.gov/aua/resources/cots.

37. Herald, T.E., Jr., "Technology refreshment strategy and plan for application in military systems a "How-to systems development process" and linkage with CAIV," *National Aerospace and Electronics Conference Proceedings*, 2000.

38. "Monte Carlo Simulation." *Palisade.com*. Palisade Corporation. Web. 30 May 2012. <http://www.palisade.com/risk/monte_carlo_simulation.asp>.

39. "The Bathtub Curve and Product Failure Behavior Part One - The Bathtub Curve, Infant Mortality and Burn-in." *Weibull.com*. ReliaSoft Corporation. Web. 30 May 2012. <http://www.weibull.com/hotwire/issue21/hottopics21.htm>.

40. Lemon, G., "Maximum Likelihood Estimation for the Three Parameter Weibull Distribution Based on Censored Samples," *Technometrics*, Vol. 17, No. 2, pp. 247-254, May 1975.

41. Cohen, A., "Maximum Likelihood Estimation in the Weibull Distribution Based on Complete and on Censored Samples," *Technometrics*, Vol. 7, No. 4, pp. 579-588, Nov. 1965.

42. Usher, J. and T. Hodgson, "Maximum Likelihood Analysis of Component Reliability Using Masked System Life-Test Data," *IEEE Transactions on Reliability*, Vol. 37, No. 5, pp. 550-555, Dec. 1988.

43. "Maximum Likelihood Estimation." *Engineering Statistics Handbook*. National Institute of Standards and Technology, Web. 19 July 2012.

44. Hill, A.V., "The Newsvendor Problem," *Clamshell Beach Press*, 7 July 2011. Web. 19 July 2012.

45. Edgeworth, F., "The mathematical theory of banking" *Journal of the Royal Statistical Society*, Vol. 51, No. 1, pp. 113-127, Mar. 1888.

46. Teunter, R.H. and Haneveld W.K., "The 'Final Order' Problem," *European Journal of Operational* Research, Vol. 107, pp. 35-44, 1998.

47. Teunter, R.H. and Fortuin L., "End-of-Life Service," *International Journal of Production Economics*, Vol. 59, pp. 487-497, 1999.

48. D. Feng, P. Singh, and P. Sandborn, "Optimizing Lifetime Buys to Minimize Lifecycle Cost," *Proceedings of the 2007 Aging Aircraft Conference*, Palm Springs, CA, April 2007.

49. Leifker, N.W., "A continuous-time examination of the last buy problem." dissertation, University of Iowa, 2010. http://ir.uiowa.edu/etd/697.

50. Sandborn, P., *Cost Analysis of Electronic Systems*, World Scientific, 2012.