A Decision-Making Framework to Determine the Value of On-Orbit Servicing Compared to Replacement of Space Telescopes

by

Mark Baldesarra

B.A.Sc. Engineering Science
University of Toronto (2005)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2007

Author . . . .
Department of Aeronautics and Astronautics
May 25, 2007

Certified by . . . .
David W. Miller
Professor of Aeronautics and Astronautics
Thesis Supervisor

Accepted by . . . .
Jaime Peraire
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Dedicated to the astronauts of Apollo 1, Space Shuttle Challenger*, and Space Shuttle* Columbia*, who gave their lives in the service of space exploration. May we never forget their sacrifice.*

Per aspera ad astra.

A Decision-Making Framework to Determine the Value of On-Orbit Servicing Compared to Replacement of Space Telescopes

by

Mark Baldesarra

Submitted to the Department of Aeronautics and Astronautics
on May 25, 2007, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

The Hubble Space Telescope has demonstrated that on-orbit servicing can provide significant benefits for scientific space programs. Specifically, servicing missions can replace failed components to keep spacecraft operational, and can upgrade onboard components to improve spacecraft performance. Hubble was able to capture these benefits because it was designed to be serviceable; however, many other programs have excluded serviceability from the design due to cost considerations. Often, the value of serviceability cannot be quantitatively justified. This thesis develops a framework to determine the value of including serviceability in a space telescope.

Various principles to evaluate serviceability are proposed throughout the literature, and this thesis incorporates three main principles to construct the framework. First, the costs and benefits of servicing are separated so that the “cost” of servicing is expressed as the maximum price the customer is willing to pay. Second, the value of serviceability will be determined by comparing a telescope servicing program to a telescope replacement program. Third, the value of flexibility provided by servicing is analyzed by a Monte-Carlo simulation and decision rule analysis.

A case study was performed to demonstrate how the framework is used, using representative data from Hubble. For a simple space telescope, the case study calculated the increase in science return gained by servicing and the maximum price for servicing missions. The case study illustrated an important trade between science return and risk of telescope downtime. Finally, the principles and techniques used in this framework are more generally applicable to non-revenue generating spacecraft.

Thesis Supervisor: David W. Miller
Title: Professor of Aeronautics and Astronautics

Acknowledgements

My research would not have been possible without the support of my research advisor, Professor Dave Miller. I am grateful for the opportunity he provided to investigate an area of research in which I am interested, and for the freedom he gave me to chart my own path, in the absence of a supporting organization. I would like to acknowledge The Aerospace Corporation library and Erik Nilsen from NASA Goddard for their contributions to my research.

I would like to thank two professors whose courses have provided the foundation for my research. First, to Professor Richard de Neufville, through his *Engineering Systems Analysis for Design* course. He introduced me to Real Options theory and provided the inspiration to apply real options techniques to determine the value of telescope servicing. Second, to Professor George Apostolakis, through his *Reliability and Applications* course. He provided the statistical tools needed to perform the reliability analysis that is critical to this framework.

My friends have made my experience here at MIT an enjoyable one and kept me on track throughout my studies, for which I must express my sincere thanks. To Bruce, for your always insightful guidance and sense of humour. To Rosie, for providing a safe harbour away from work and always being good for a laugh or two. To Zoe and Laslo, for keeping *The Canadian Contingent* alive and well. And to Angela and Jessica, back at home in Canada, for keeping me connected to my roots and sticking with me despite the distance between us.

And finally, my family has been critical to my success here at MIT and throughout my academic career. They kept me going while I tried to find my footing in first semester, and they have always provided advice on trials and tribulations, both large and small. I am grateful that they were always a phone call away whenever I needed them. Dad, Mom, and Lauren: I couldn't have done it without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Hubble Space Telescope (HST) has provided an unprecedented glimpse into the structure of the universe. Throughout its first eleven years of operation, HST has produced about 420,000 images, observed over 17,000 targets, and contributed to over 3,200 scientific papers [1]. The success of HST lies partly in the use of on-orbit *servicing*. Servicing is the act of physically replacing, modifying, and/or upgrading components on an operational spacecraft in its deployed environment. As of 2007, four servicing missions have been sent to HST, with a fifth currently scheduled for 2008. HST is serviced by astronauts through extra-vehicular activity (EVA) and supported by the Space Shuttle, as shown in Figure 1-1.

The Hubble Space Telescope

Figure 1-1: HST being serviced by astronauts and the Space Shuttle

Servicing can be used to maintain a spacecraft to keep it operational throughout its *nominal mission duration*, which is the minimum lifetime of the spacecraft mandated by its operators. Maintenance is required for complex spacecraft such as telescopes that are intended to last for an extended life, since it is infeasible and uneconomical to be designed to remain operational for decades without assistance; for example, HST was designed with a nominal mission duration of 20 years [1]. Each HST servicing mission replaced failed components with new, and often improved, components. As a result, HST has been almost continuously operational from its deployment in 1990 through to 2007.

Servicing can also upgrade components on the spacecraft to improve performance. In particular, the science instruments can be upgraded with new technology to increase their resolution and sensitivity. The HST servicing missions have replaced the instrument several times, and those currently installed on the telescope are orders of magnitude more accurate than the original instruments. This has kept HST on the cutting-edge of astronomical research, and continued servicing will keep it there for the foreseeable future. The net result of servicing is a telescope that has successfully operated throughout its mission and provided ever-increasing science capabilities.

These benefits of servicing do not come for free. The servicing missions themselves have considerable direct and indirect costs. The fifth servicing mission, which like the previous four will be Shuttle-based, has an expected price tag of $900M [2], and a cancelled plan for a robotic servicing mission to HST would have cost between $1.7B and $2.4B[1] [3]. In order to perform servicing on a telescope, it must first be designed with *serviceability*, that is, the ability to be serviced by an external agent, whether human or robotic. HST was designed with doors to provide access to internal systems, the components were modularized to allow easy removal, and the instruments were designed to be swapped with new ones. Incorporating these design features incur additional costs before the telescope is launched, but without these features, it is extremely difficult (if not impossible) to service the telescope. Clearly, the decision to include serviceability in a telescope design must be made during the design phase.

Serviceability has been excluded from telescope designs in the past because the cost of serviceability couldn't be justified. The Chandra X-Ray Observatory was originally intended to operate in Low Earth Orbit (LEO) and be serviced via the Space Shuttle. When cost constraints required the mission to be descoped, the plan to service the telescope was abandoned, and the intended orbit was switched to a highly elliptical one instead of LEO [4]. This dilemma is at the heart of the issue of telescope servicing. Scientists, engineers, NASA administrators, and policy makers all understand the benefits associated with servicing telescopes, but they often cannot justify the added costs associated with enabling this benefit captures. When budgetary pressures appear, serviceability is dropped from designs.

What is needed is a method to quantitatively determine the *value of serviceability*, so engineers and program managers have the information needed to make the decision. For space systems that generate revenue as a benefit, the value of serviceability can be easily calculated using standard economic valuation techniques. Consider the example of a commercial communications satellite constellation. The benefits of the system are the revenues from subscribers to the service. Servicing operations can modify satellites to support more subscribers, which can potentially increase revenue [5]. Since both the benefits and costs of servicing are measured using the same units (dollars), the decision to service a satellite (and to design the satellites to be

[1]This was the estimated range from the team that developed an HST robotic servicing concept. An independent audit of the program revealed that the actual cost could have been much higher.

serviceable) can be made on a standard Net Present Value (NPV) analysis [6]. In this example, the question becomes: are the up-front costs of serviceability and the costs of servicing missions repaid by the increase in future cash flows due to the improved system performance?

In the case of space telescopes, however, the benefit that the system generates is science data, rather than revenues. Science data has no monetary equivalent, so a standard NPV analysis cannot be used; with the benefits and costs in different units, there is no direct way to combine these metrics into a single quantity that can guide the decision. Absent any budgetary restrictions, scientists would clamour to include serviceability in design and set aside funds to pay for servicing missions in the future to install new, advanced instruments. However, when budget pressures are applied, all telescope features are critically analyzed to determine if they should remain in the design or removed to save cost. Without a method to analyze the value of serviceability, it will often be dropped because the cost cannot be justified.

Furthermore, the telescope program is subject to many sources of uncertainty. For example, the mean time to failure for components can be calculated, but the telescope failure time is not known *a priori*. As well, the future instrument technology that is installed onboard depends on the time of servicing, which affects the overall science return of the telescope. Furthermore, the servicing missions themselves have a likelihood of failure. The mission may fail to service the telescope, or it may inadvertently disable or destroy the telescope; each of these events has a probability associated with it. Any method of calculating the value of serviceability must account for these sources of uncertainty.

There is a need for a method to analyze the value of including serviceability in the design of a telescope. The goal of this thesis is to develop a framework to perform this analysis. The current state of research will be discussed in Chapter 2, which will motivate the specific research questions that must be addressed. The general principles that will be used in the framework to answer these questions are described in Chapter 3. The framework itself will be constructed in Chapter 4. Finally, the framework will be demonstrated through a case study in Chapter 5.

Chapter 2

The State of Research: Literature Review

This chapter will investigate research into telescope servicing, which is a specialized segment of the general on-orbit servicing field. On-orbit servicing in turn falls into the broader class of real options theory. This can be thought of as a funnel which narrows down towards telescope servicing, as illustrated in Figure 2-1.

This chapter is organized by starting at the top of the funnel and gradually working down to the specific field of telescope servicing, and specific areas of interest will be

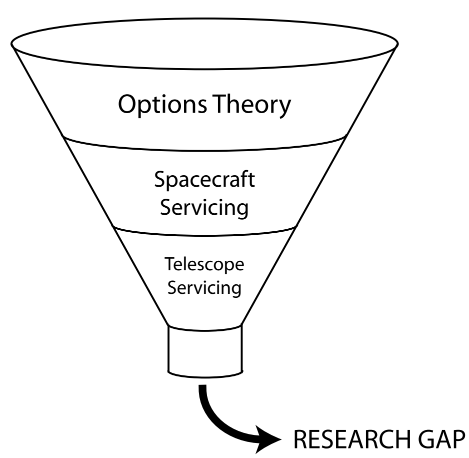


Figure 2-1: The hierarchy of servicing research

highlighted along the way. The topics investigated in this literature review are the following:

1. Define *value* as used in this thesis.

2. How can options theory be used to think about spacecraft servicing?

3. What are the benefits of servicing? How have these benefits been demonstrated on HST?

4. What are the costs of servicing?

5. Examine current research into the value of spacecraft servicing.

6. Examine current research into the value of telescope servicing in particular.

7. Identify the research gap that this thesis will address.

2.1 What is Value?

Spacecraft servicing has not been widely adopted partly as a result of misconceptions on the part of engineers about the concept of *value*. Although engineers may share the view that spacecraft servicing is "valuable", in many cases this is confused with the concept that spacecraft servicing can generate many *benefits*.

This confusion can be seen even in programs where servicing has been clearly demonstrated as valuable. For example, in the NASA media guide for the fourth servicing mission to the Hubble Space Telescope (HST), a short section entitled "The Value of Servicing" states that:

> Hubble's visionary modular design allows NASA to equip it with new, state-of-the-art instruments every few years. These servicing missions enhance the Telescope's science capabilities, leading to fascinating new discoveries about the universe. Periodic service calls also permit astronauts to "tune up" the Telescope and replace limited-life components [1].

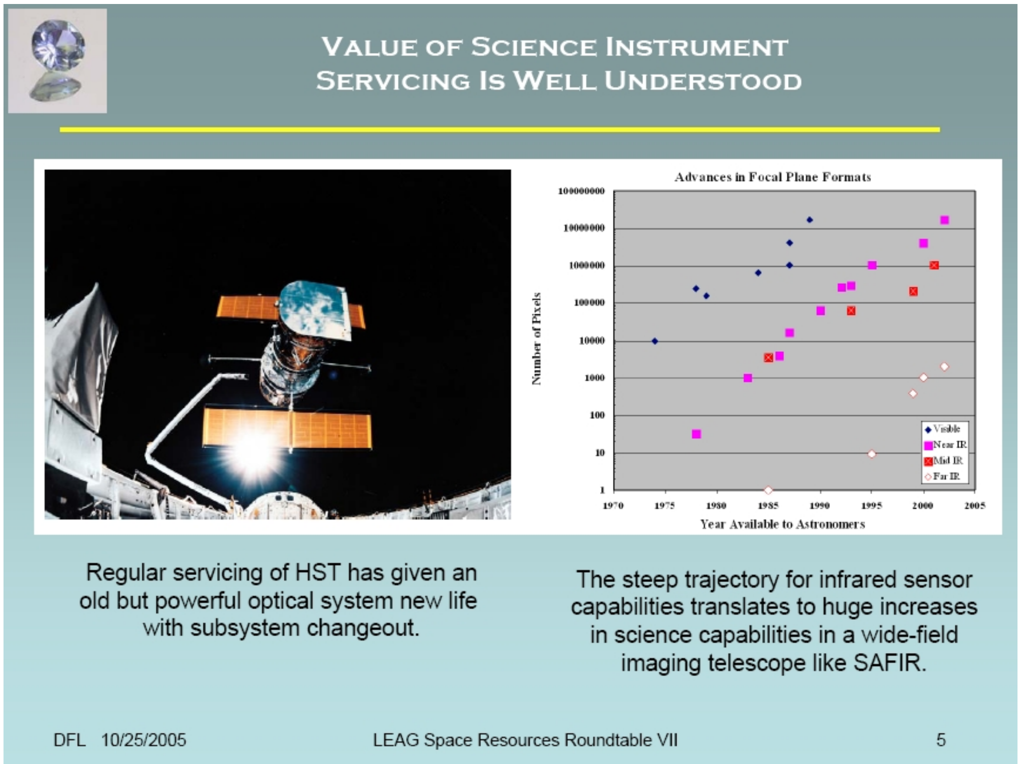


Figure 2-2: Example of focussing on benefits when discussing the value of servicing (from Lester [7])

Another example is from a presentation on the inclusion of serviceability in the design of a future space telescope. Most of the slides in the presentation discuss proposals for the servicing architecture and implications of serviceability on design, but one slide is devoted to the "value of servicing", and shown in Figure 2-2. The value, according to the presenter, is clearly apparent from a graph of increasing instrument capabilities, implying that servicing can allow new and advanced instruments to be installed in the future [7]. This is correct; however the title of the slide, "Value of Servicing is Well Understood", is somewhat misleading.

Both of these examples share a common issue: the "value" of servicing as presented ignores the cost of servicing. Serviceability is worthless (i.e., has no value) if servicing missions are prohibitively expensive, or the necessary modifications to the telescope to enable servicing are too extreme. No matter how much benefit can be gained from servicing, if the associated costs (monetary or otherwise) are too high, servicing is not valuable. In addition, the benefits themselves are often expressed qualitatively. The graph in Figure 2-2 merely demonstrates that instrument technology will significantly improve. How does this change the science return of a telescope, and by how much? It is often unclear exactly how a telescope program will gain from servicing, save for a fuzzy notion that servicing will improve science capabilities.

The net result of these difficulties is that serviceability is often left out of telescope design studies. Even if it is included in the design, it is often removed when budgeting and scheduling pressures start to mount. If engineers cannot quantitatively justify the serviceability, it is unlikely to be present in the final design.

Clearly, a more satisfactory definition of value is needed, and there have been several proposed in the literature. Murman [8] defines value in terms of a business enterprise. He states that value comes from transactions that provide utility to an organization. Specifically, Murman defines value as:

> How various stakeholders find particular worth, utility, benefit, or reward in exchange for their respective contributions to the enterprise.

These transactions both can provide utility (benefits) or payment for those benefits (costs). Likewise, Rouse and Boff define value in the context of systems engineering as "a fair return or equivalent in goods, services or money for something exchanged" [9]. The general theme in these definitions is that value comes from generating benefits while incurring associated costs, neatly summarized by Crawley as "value is benefit at cost" [10]. He states that a "good" architecture is one that delivers benefit at a competitive cost. Thus to calculate the value, one must evaluate both benefits and costs to determine if the benefits received justify the expense in generating them.

2.2 Options Theory

Serviceability can be considered as an *option* that is designed into a telescope program. In its most general sense, an option is the right, but not the obligation, to take an action in a specified time period and for a certain price [11]. Servicing fits well within this framework: telescope serviceability gives managers the right (but no obligation) to service the telescope throughout its operational lifetime. However, serviceability must be designed into the system before it is launched, so engineers and program managers must decide to incorporate it into the initial telescope design. This section discusses the various types of options in use in the business world, which will motivate how the servicing option can be thought about.

2.2.1 Financial Options

Long before options were introduced as tools to analyze projects, they were used as financial instruments. A financial option is a contract between two parties where the option purchaser is given the right (but not the obligation) to either buy or sell an asset in the future at a certain time. The asset to which the option applies to is called the *underlying asset*. Underlying assets for options can be one of a wide variety of financial instruments, such as shares, bonds, mutual funds, and foreign currencies.

Options became widely used in finance and business to either profit from unexpected gains or protect against risk. As a result, options were classified into two primary types: A *call* option is the right to perform an action to take advantage of a favourable opportunity, whereas a *put* option is the right to perform an action to limit losses in a bad situation [12]. In the case of a stock option, a call allows the holder to profit if the stock price goes high, and a put allows the holder to prevent losses if the stock price goes low.

When one performs the action that the option allows, it is called *exercising* the option. For both call and put options, purchasers spend money now to have the opportunity to exercise the option in the future as conditions warrant. To determine the value of an option, various mathematical treatments were developed, which are further described in Chapter 3. The goal of these analyses is to determine how much should be paid up-front to purchase the option and how much benefit can be realized.

The value of a financial option can be determined using the following parameters:

- The *initial cost* of the option.
- The *strike price* of the underlying asset: the set price at which the asset must be traded when the option is exercised.
- The *expiration date* of the option: the time period over which the option can be exercised.
- The *volatility* of the underlying asset: the distribution of potential returns of the asset over the life of the option. [11].

With this terminology, the stock option example can be expressed more formally. A call option on a stock grants the purchaser the right to purchase the stock at the strike price on or before the expiration date. A put option on a stock grants the right to sell the stock at the strike price on or before the expiration date[1] [6].

Financial options are attractive because their value is *asymmetric*. The option holder will only exercise it when it is advantageous; for a call option, the holder will only exercise it when the price of the asset (known as the *spot price*) is above the strike price. Conversely, if the spot price is below the strike price, the holder will not exercise the option, as it would lead to a net loss. So the holder can only have a net benefit from the option; the option is exercised when profitable, and it is not exercised if it would result in a loss [11]. The expected value of an option is always positive. This comes with the important caveat that the option itself must be purchased in the first place. Thus, if the option is not exercised, no loss is incurred, **except** for the initial cost of the option.

Options are more valuable when the underlying asset has higher *volatility*, which is defined in economics as the possible spread of the asset value [6]. In the case of financial options, the underlying asset is more volatile if the distribution of asset price is wider. Returning to the call option example, having a higher chance of the asset price increasing means a greater expected payoff when the option is exercised. Of

[1]This discussion has focussed on *American* options, which can be exercised at any time before the expiration date, rather than *European* options, which can be exercised only on the expiration date. This distinction is not relevant to the discussion, so European options are omitted.

course, along with a higher chance of price increases comes a higher chance of price drops, but since the value of a call option (as with all options) is asymmetric, having a greater chance of low prices is of no importance because no loss is incurred if the price is low. Overall, the expected return of the option is greater for a more volatile underlying asset.

The key insight from financial options theory is that uncertainty and risk are not necessarily negative. In fact, uncertainty itself can be a source of value, since more uncertainty (i.e. volatility) provides a chance of larger payoffs with no downside risk. Options are instruments that can be used to capture this value. Again, this comes with the important caveat that the option must be purchased first. The remaining issue to be resolved with financial options is to determine if the up-front cost of purchasing the option is justified by the potential future payoff.

2.2.2 Real Options

Financial options are limited to actions on financial instruments, but the concept of creating an opportunity to perform actions in the future is more broadly applicable. In the business world, companies often structure contracts to include provisions to act if revenues or profits increase, or escape clauses that allows the company cut its losses if conditions deteriorate. More generally, large business operations, such as factories, refineries, and mines, are not designed and built to be static throughout their lifecycles: changes are made depending on how demand or other business factors evolve through time. Good designs are ones that allow for these changes to be made without incurring very large expenses. All of these examples can be considered as types of options, but instead of an option on a financial asset, these options are on business projects. Thus this particular class of options are called *real options*, so named because they act on real, tangible projects.

Similar to financial options, real options can be classified into one of two basic types: *call-like* and *put-like*. Call-like real options are those that can be exercised to capture benefits when the value of the project increases, such as the ability to increase production at a factory if demand increases. Put-like real options are exercised to limit future losses, such as the ability to slow or halt production at a factory if demand decreases [11]. In both of these cases, expenses may be incurred in order to exercise the option. For the call option example, the factory manager may have to increase wages to increase production or spend capital to expand the plant. The option provides the opportunity to perform actions as future conditions warrant, and without it those actions may be impractical or prohibitively expensive.

Another, complementary classification for real options, proposed by Richard de Neufville, professor of Engineering Systems at MIT, concerns the level of knowledge about the project design that is needed to purchase the real option. Real options, of both the call-like and put-like varieties, can be considered either "on" or "in" a project [11]. Real options "on" a project are activities that can be performed at the project level without regard to its internal design, whereas a real option "in" the system is one that is built into the system design itself. For example, consider a mining project. If managers purchase land to provide a future opportunity to mine natural resources, this is a real option "on" the project, since it is independent of the eventual design of the mine. In contrast, if the mine is designed so that production can be increased if other nearby deposits are found, this is an option "in" the system since the design itself was altered.

This is similar to the ability of financial options to extract benefits from volatility and uncertainty, but whereas financial options manage uncertainty in underlying assets, real options manage uncertainty in real projects. Furthermore, the value of a real option (as with a financial option) stems from the ability to make decisions in the future based on conditions as they happen. The ability to adapt to future events provides management with a measure of *flexibility*. Flexibility is defined in [13] as

> The ability of a system to adapt and respond to changes in its initial objectives, requirements and environment occurring after the system is in operation in a timely and cost-effective manner.

The key here is that built-in flexibility provides managers the ability to make changes that are cost-effective. Many changes can be made after a project starts, but without the real option, the change may be prohibitively expensive.

Just as flexibility is valuable where there is uncertainty, flexibility is worthless in a deterministic world. If future events are known in advance, the project can be designed to maximize utility given this set of events, and the ability to make changes is not valuable. In reality, the future contains uncertainty, so a real option provides the ability to make changes in the future when managers have more information and the uncertainty has been resolved. Furthermore, like their counterparts in the financial world, real options are more valuable with greater uncertainty. Call-like real options are more valuable when there is a higher chance that the conditions surrounding the project improve, and the option can be exercised to capture additional value. In contrast, put-like real options are valuable if there is a larger risk of negative events, and the option can be exercised to cut losses.

The key question that remains is to find the value of the real option so that planners know how much they should pay to incorporate the option into the project. As with financial options, the specific option valuation techniques will be discussed in Chapter 3.

2.2.3 Servicing as a Real Option

Servicing of space systems while deployed in orbit can be considered a real option because servicing provides the flexibility to perform actions to improve benefits or cut losses as future conditions warrant. The servicing real option can be loosely described by the four parameters that describe a financial option. The analogue between servicing as a real option and standard options are described in Table 2.1 and are discussed in more detail below.

First, the *initial cost* of the option corresponds to the cost of engineering, development and fabrication associated with enabling servicing in the space system. In the case of a serviceable satellite, the satellite must be designed with doors for access by astronauts or robotic servicing systems, the replaceable components must be modularized so that they can be removed and swapped easily, etc.

Table 2.1: Parameters for Financial Options and Analogues in Real Options

Parameter	Financial Option	Servicing Real Option Analogue
Initial Cost	Price of option contract on purchase	Costs associated with incorporating serviceability in the initial design
Strike Price	Price that the underlying asset will be bought / sold when option is exercised	Costs associated with performing the servicing mission in the future
Expiration Date	Latest time when the option can be exercised	Time period over which the option to service is available
Volatility	Possible spread of asset prices over life of option	Uncertainty in the performance or other parameters of the space system

Second, the *strike price* of the financial option corresponds to the cost of exercising the servicing real option in the future. Although the servicing option was built in (or purchased) during the design phase, the servicing mission itself will not be free. Costs are incurred when the servicing mission is launched; including the cost of the servicing spacecraft, replacement components, launch, and operations. The costs incurred may not end after servicing is complete: the system may be modified and so the operations costs may change.

Third, both financial and real options have an *expiration date* because there may be only a fixed time interval over which the options can be exercised. Normally, a spacecraft can be serviced throughout its operational life, but there may be cases where servicing cannot be exercised past a certain time. For example, if a spacecraft requires a large amount of propellant to manoeuvre into position to be serviced, it may not be available later into the mission when fuel is depleted, although it still may be able to continue normal operations.

Finally, the *volatility* underlying a financial option corresponds to various sources of uncertainty in a telescope program. The value of the servicing real option increases if the underlying uncertainty associated with the space system is high. Consider a space system whereby servicing missions are sent to repair components that failed prematurely. If there is no uncertainty in the failure rate, it is possible for a spacecraft to be designed with sufficient redundancy to last for the entire mission. In that case, the spacecraft would not require servicing, and the value of serviceability is zero. On the other hand, as the uncertainty in the failure time increases, so does the probability of premature failure, which makes the option to service more valuable.

2.3 Benefits of Servicing

The benefit of the servicing real option is the flexibility it provides to respond to future events. Nilchiani and others [14] categorizes three types of flexibility by the time frame over which these changes occur: short-, medium-, and long-term. In the short-term, components or the entire spacecraft may fail, which requires urgent repair or replacement to ensure that the overall performance of the system is not compromised. This is called *emergency service flexibility*. In the medium-term, changes in demand may require the system to be adapted to support these changes. This is called *volume flexibility*. In the long-term, the type of service demanded may change, requiring more substantial changes to the system to fit the new mission need. This is called *mix flexibility*.

These three flexibility types can be illustrated using the example of a constellation of communications satellites, where the satellites provide service to terrestrial subscribers [5]. In the event that a satellite is damaged, emergency service flexibility allows the satellite to be repaired or replaced in order to maintain service. If demand from the subscriber base sharply increases, volume flexibility allows the constellation to be reconfigured by adding more satellites to meet demand. If subscribers demand different services over time (for example, a shift from telephony to Internet service), mix flexibility allows the satellites to be reconfigured to support the new service. To enable these different types of flexibility, the system must be designed with a real option that can be exercised as needed.

Table 2.2: Benefits of servicing to Earth Observation Missions

Science Benefits • The science data set is increased through more observation time. • The science data set can be used in conjunction with other missions. • Unique capabilities onboard a satellite can remain in operation • Any unexpected science results gained during the nominal mission can be investigated further.
Operational Benefits • The satellite may still be useful and provide valuable data. • Satellite procedures and/or technology can be further demonstrated or validated. • The satellite may be useful in future applications that are currently unanticipated.

In the context of scientific missions, flexibility has already been explicitly recognized as valuable for Earth observation missions (EOMs). At the end of the nominal mission duration of a scientific satellite, the program undergoes a Senior Review to determines if the satellite program should be extended past its nominal duration [15]. The National Academy of Sciences commissioned a report to investigate the circumstances in which EOMs should be extended. The report found seven benefits to extend EOMs, summarized in Table 2.2, categorized as either benefits related to the acquisition of science data, or benefits related to continued spacecraft operations.

In addition to adding benefits to a program, flexibility can be used to protect programs against risk. Joppin [16] identifies four sources of risk are important for space systems that are deployed over medium- to long mission durations, which can be mitigated by incorporating flexibility into the system:

- *Risk of system failure*: The system fails prematurely due to component wearout, random failures, or design errors, and is unable to satisfy its intended mission duration without intervention.

- *Risk of commercial obsolescence* The actual market demand profile is drastically different than the assumed demand profile used during the design phase, and the system either cannot satisfy demand or has too much capacity.

- *Risk of technology obsolescence*: Technology on the spacecraft is made obsolete by new developments on the ground and so the spacecraft becomes less useful to customers.

- *Risk of change in customer requirements*: The desires of the customers change over time, and the new desires cannot be served by the system as originally designed [16].

These four risks can map onto the three types of flexibility described by Nilchiani. System failure is a short-term issue that can be immediately addressed via emergency service flexibility. Commercial obsolescence can either be an inability to meet the level of demand (which can be remedied by volume flexibility) or an inability to provide the demanded type of service (remedied by mix flexibility). For both technological obsolescence and requirements changes, the system needs to be reconfigured, which is possible only with mix flexibility.

In a later paper, Joppin [17] analyzes the value of flexibility specifically for scientific missions. She identifies four primary areas where the ability to service a science mission can provide value over the mission lifetime:

- *Mission salvage*: The system is damaged before becoming operational. Without the ability to service, the mission is an immediate failure.

- *Repair and maintenance*: As components on the spacecraft fail, servicing missions can replace these components to ensure it can continue to operate. This applies to both expected (wearout) failures and unexpected (random) failures.

- *Instrument upgrades* The instruments can be replaced to improve the science return with newer technology. Also, if the objectives of principal investigators change, the installed instruments may not be sufficient. For example, if an instrument is optimized for the infrared wavelength band but scientists want to focus more on ultraviolet, the instrument must be replaced in order to satisfy this new demand profile.

- *Bus component upgrades*: The supporting equipment, such as power systems, onboard computers, or environment control, can be upgraded to drastically improve science return even with the same instruments.

In summary, flexibility provided by servicing can theoretically provide a wide range of benefits to a space mission. Servicing can both protect against bad conditions (failures and risks) and provide increased benefits under good conditions (installation of new technology through upgrades).

2.4 Servicing and the HST Experience

Previous sections have shown that the flexibility to perform servicing can potentially provide significant benefits, but it is difficult to accurately quantify these benefits. Fortunately, the Hubble Space Telescope (HST) has clearly demonstrated these benefits on an actual telescope. HST was intended to replicate a ground observatory in orbit [18]. Ground observatories are designed to be flexible so that they can be upgraded with new instruments to remain on the leading edge of technology. This is possible because ground observatories are readily accessible for maintenance, repair, and upgrade activities. As well, the scientific instruments are generally kept separate from the optical bench, so they can be removed and replaced as needed. An observatory in orbit is more advantageous due to the more stable platform that space provides: free from local vibrations, earthquakes, and most importantly, atmospheric interference [19]. However, a space telescope is much more inaccessible than ground telescopes due to its location, and components must be more integrated to save mass, volume and cost. Despite these difficulties, the advantages of flexibility that are seen with ground telescopes are still valid for space telescopes.

HST was initially intended to be returned to Earth via the Shuttle to undergo periodic maintenance, but in the end HST was designed to allow servicing in orbit by astronauts performing extra-vehicular activity (EVA) [17]. The internal components were modularized and designed such that they could be repaired or replaced by astronauts during servicing missions, which were planned to occur approximately once every three years [20]. As of 2007, four servicing missions have been sent to HST, with a fifth mission scheduled for 2008 [21]. Table 2.3 provides a summary of the components replaced by the four servicing missions [21, 22].

The flexibility to perform servicing operations on HST granted NASA the ability to perform four categories of tasks:

- *Preventive maintenance*: Replace components subject to wearout before they fail and cause suspension of HST operations.

- *Corrective maintenance*: Replace components if they fail prematurely and repair flaws detected after deployment.

- *Bus upgrades*: Replace engineering components with improved hardware to improve the lifetime and/or utility of HST.

- *Instrument upgrades*: Replace onboard scientific instruments with more advanced instruments [16].

These four categories are discussed in more detail below. Although both types of maintenance (preventive and corrective) is usually considered together as one category, the distinction is made between these two types by Waltz [23] since they are each performed under different circumstances.

2.4.1 Preventive Maintenance

Components on HST do not last indefinitely. If the telescope is to operate for its intended 15-year mission duration, these components need to be maintained or replaced. For example, HST depends on a set of six Rate Sensing Units (RSUs). Each

Table 2.3: Summary of major HST components affected during servicing

Servicing Mission 1 • Replaced 4 of 6 Rate Sensing Units • Replaced 2 of 2 Magnetometers • Upgraded the flight computer coprocessor • Replaced solar arrays • Replaced solar array drive electronics
Servicing Mission 2 • Replaced 1 of 3 Fine Guidance Sensors • Replaced 1 of 4 Reaction Wheel Assemblies • Installed Optical Control Electronics Enhancement Kit • Replaced Solar Array Drive Electronics • Replaced Tape Recorder with Solid State Recorder
Servicing Mission 3A • Replaced 6 of 6 Rate Sensing Units • Replaced 1 of 3 Fine Guidance Sensors • Installed Voltage/Temperature Improvement Kits • Installed new computer (Intel 486) • Upgraded Solid State Recorder • Replaced thermal insulation blankets
Servicing Mission 3B • Replaced 1 of 4 Reaction Wheel Assemblies • Replaced solar arrays • Replaced Power Distribution Unit

RSU contains gyroscopes and electronics to detect the orientation of HST and provide data for the reaction wheels to point the telescope [24]. Three of the six RSUs are required to be functional for science operations. Since gyroscopes are subject to wearout, the backup RSUs are brought online when the primaries fail, but without intervention the number of functioning RSUs will eventually drop below the minimum of three, at which point telescope will cease operations. Servicing missions were planned accordingly to replace gyroscopes as they wear out and to prevent the number of functioning gyroscopes from dropping below three [25].

2.4.2 Corrective Maintenance

Not all events can be foreseen, so corrective maintenance operations perform repairs on unexpected failures. The infamous example from HST was the error in the primary mirror discovered immediately after launch, when the first images captured by HST had a lower resolution than expected. After an investigation, NASA concluded that the primary mirror had a slight spherical aberration flaw caused by manufacturing errors, which caused light entering the telescope to converge away from the focal plane [26]. To correct this problem, engineers designed the Corrective Optics Space Telescope Axial Replacement (COSTAR) optics package, which was installed on the first servicing mission to return HST to its designed specifications.

Corrective maintenance was also useful later into the life of HST. The gyroscopes inside the RSUs wore out much sooner than anticipated, and by 1999 only two gyroscopes were functional. This caused HST to go into standby mode and suspend science operations. A servicing mission was originally scheduled to launch in 2000 to perform preventive maintenance on the RSUs. Since the components failed earlier than expected, the servicing mission was split in two, and the RSU replacement portion of the mission was bumped up to December 1999. The original preventive maintenance mission became a corrective one once HST ceased operations due to hardware failure [25].

In both of these cases, the ability to perform corrective maintenance saved HST from having a less productive and shorter lifetime than it has had so far.

2.4.3 Bus Upgrades

Maintenance operations alone would keep HST running at its designed specifications with no increase in science return. To act as a true observatory like its ground-based counterparts, HST components were upgraded during servicing missions. Components such as onboard processors, data storage, solar arrays, and control systems were upgraded with new technology. These upgrades increased the overall telescope performance by improving the characteristics of the supporting systems. For example, upgrades to the onboard computer systems increased the data storage capability from 3 GB at launch to 21 GB at present, which enabled better management of data gathered by the instruments [27]. Furthermore, combinations of bus upgrades can lead to systemic telescope improvements. For example, upgrades to the structure and control systems reduced the peak jitter from 39 mas at launch to 14 mas at present [27]. Clearly, both component-level and system-level improvements to the supporting bus improved the science return of HST.

2.4.4 Instrument Upgrades

From the astronomer's perspective, perhaps the most exciting type of servicing operation is the installation of new telescope instruments. HST has five bays for instruments that can be accessed by astronauts so that instruments can be replaced during servicing missions. Three of the four HST servicing missions included the installation of new, state-of-the-art science instruments, and the fifth servicing mission planned for 2008 will install two more. Figure 2-3 shows a timeline of the progression of science instruments throughout the life of HST. Successive generations of instruments have yielded an enormous increase in resolution and sensitivity in many wavelength bands. For example, the Advanced Camera for Surveys (ACS) is a third generation instrument currently installed on HST. Compared to the first generation instrument that was replaced, the instrument resolution has doubled and the field of view is over 40 times greater [28]. The net result is a huge increase in science return, both in data quantity and quality.

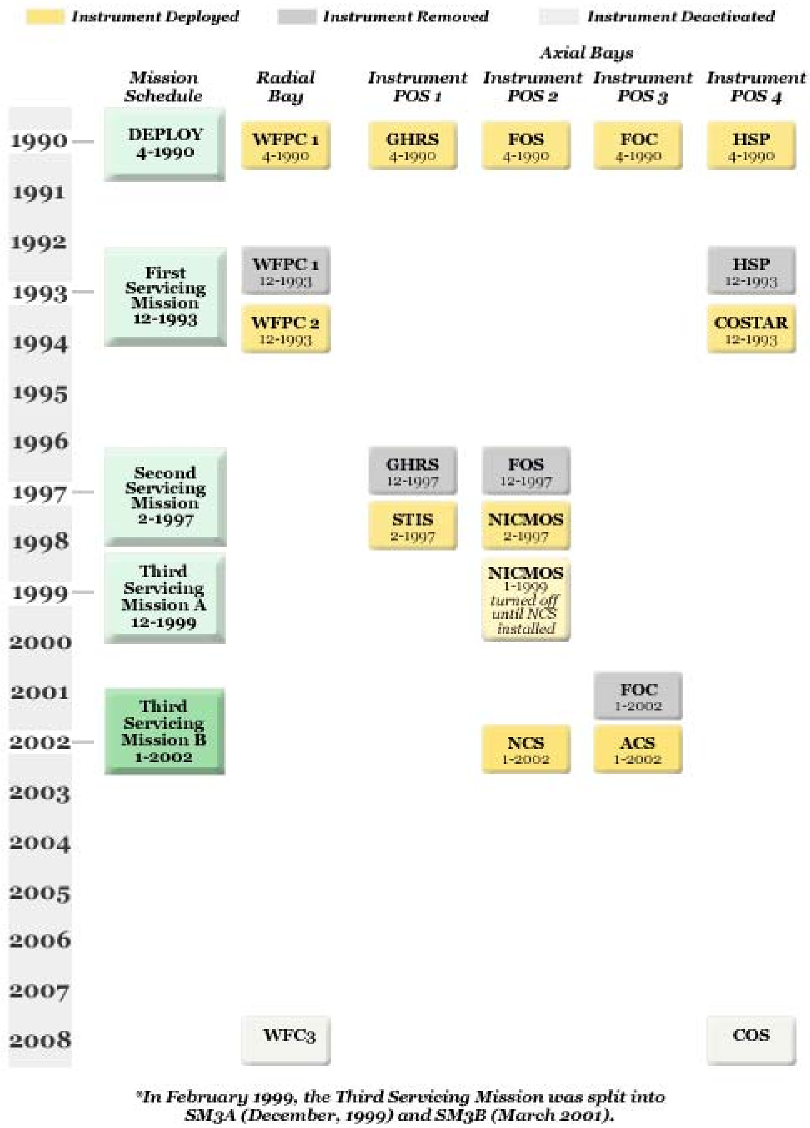


Figure 2-3: Timeline of instruments installed on HST (adapted from NASA [29])

2.4.5 Overall Impact

Through maintenance and upgrades, the servicing missions have significantly increased the science return of the telescope from its original design. Figure 2-4 shows the amount of science data generated by HST per month.

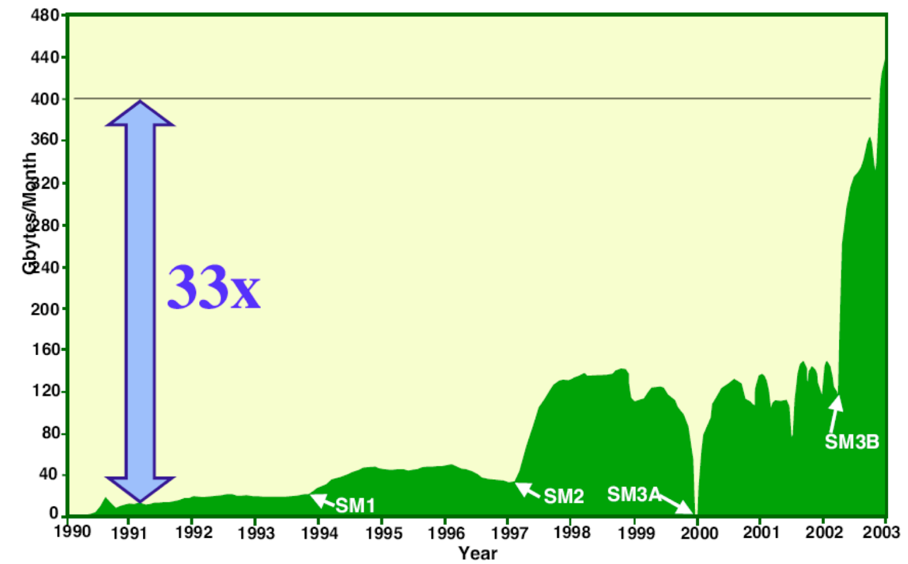


Figure 2-4: Science return from HST between 1990-2003 (from Dedalis [27])

Some important insights from the figure are:

- After each servicing mission, the amount of science data returned increases dramatically. This is because the servicing missions restored HST to full health (via maintenance) and increased the science return (via upgrades).
- By 1999, science operations were halted because four of six RSUs had failed. If servicing were unavailable, HST would not have been repaired, the mission would have ceased, and none of the science data from the year 2000 onwards would have been collected.

The experience of HST has clearly demonstrated that servicing telescopes provided enormous contributions to the scientific community. The benefit of servicing is no longer a theoretical concept.

2.5 Servicing Technology: The *How* of Servicing

Most spacecraft servicing research has been directed towards the technology of servicing. This research has shown that, in order to render a spacecraft serviceable, various requirements are imposed on the system. Satisfying these requirements takes time and costs money. The following is a brief survey of current and planned servicing vehicles and technology demonstrators, which will motivate some of these requirements.

2.5.1 Servicing Technology Testbeds

As of 2007, there are four major programs that are intended to develop and demonstrate technologies that are required for spacecraft servicing.

1. **ETS-7** (1997) was a Japanese testbed to demonstrate docking technologies. It consisted of two spacecraft: a *target* and a *chaser*. The target was a *cooperative* satellite, which means it maintained attitude control and had markers painted on the satellite to aid the chaser in its approach. The chaser used a combination of GPS and LIDAR to approach a target satellite from up to several kilometres away, and optical sensors to detect the markers on the target for final docking procedures within 2 metres. The test demonstrated technologies needed to perform operations such as refuelling, structural deployment, and component replacement. [30, 31, 32]

2. **Demonstration for Autonomous Rendezvous Technology** (2005), or DART, was a NASA testbed for proximity algorithms and operations. The DART spacecraft was to perform a sequence of manoeuvres around a pre-deployed MUBLCOM satellite. MUBLCOM acted as the target and was equipped with retroreflectors as navigation aids for the servicer. The mission was a partial success, but the DART spacecraft inadvertently collided with the target and ended the mission prematurely. [33, 34]

3. **Orbital Express** (2007) is a DARPA/Boeing spacecraft to test orbital servicing technologies. The mission consists of two spacecraft: a servicer (ASTRO) and a

target (NEXTSat). ASTRO is a servicing technology testbed, and NEXTSat is a prototype for future serviceable spacecraft. The main servicing tasks that will be tested include fuel replenishment and replacement of modularized spacecraft components [34].

4. **Spacecraft for the Universal Modification of Orbits** (2010), or SUMO, is a DARPA spacecraft that can dock with satellites without specialized markings or fixtures. It uses a large set of cameras to detect the target satellite and uses multiple robotic arms to attach itself to launch fixture holes on the target. SUMO is often called an "orbital tow truck": it will dock with satellites in GEO and provide sufficient delta-v to modify their orbits. SUMO operates autonomously due to the large time delay between Earth and GEO, which precludes teleoperation of the precise, time-dependent motions that the spacecraft is required to perform [35, 36].

2.5.2 Technical Requirements for Servicing

The programs described above have demonstrated that the target spacecraft have several requirements placed on them in order to enable servicing, including:

- *Failure Identification*: The target spacecraft must be able to determine which components failed, so that the servicer can repair all necessary components.

- *Docking Mechanism*: The target spacecraft must allow the servicer to dock with it. Generally this is accomplished by attaching a docking interface to the target spacecraft exterior.

- *Docking Cooperation*: The target spacecraft must be equipped with docking aids to assist the servicer during proximity operations. Docking aids can either be passive (retroreflectors, optical targets, etc) or active (sensors, beacons).

- *Attitude Control*: The target spacecraft must be stabilized prior to servicer approach. Prior to servicing, the target can either fix its attitude relative to inertial space (as telescopes often do), or the target can control its attitude in tandem with the approaching servicer spacecraft. If the target is free-floating (which may occur as a result of a major systems failure), it is very difficult for the servicer to dock, and the servicer would require very specialized equipment to attempt docking.

- *Accessibility*: The target spacecraft must grant the servicer access to internal systems. This may include ports for fuel and electrical power, as well as external doors to provide access to internal components that will be repaired.

- *Modularization*: Replaceable components onboard the target must be designed to be removeable by the servicer. Modular designs must consider the capabilities of the system that perform the replacement, whether it is robotic or an astronaut on EVA. Note that modularization may incur cost and mass penalties due to a lower packing efficiency and an increased number of interfaces.

The lesson here is that servicing places major design requirements on a serviceable spacecraft. Satisfying these requirements will incur additional mass and costs in the design phase. If these costs cannot be justified, then program managers will drop serviceability from the spacecraft design.

2.6 Previous Spacecraft Servicing Studies

Servicing operations are invariably expensive, so to be incorporated in future space programs, servicing must be *economically* justified to program managers, particularly when faced with budget and schedule pressures. Various studies have attempted to analyze the economics of servicing. Reynerson [37] created a mathematical model to analyze the value of servicing for a constellation of satellites. The constellation includes multiple satellites in different orbits and planes, and it is serviced by a reusable servicer spacecraft which transfers to the orbit of the satellite needing repair. Servicing operations consist of replacing a percentage of the satellite mass, either by component replacement or by refuelling. When the servicer is not in use, it is parked at an orbital servicing depot, which contains replacement units and extra fuel.

Reynerson uses the metric of lifecycle cost savings to determine whether or not servicing is valuable for a particular combination of constellation and servicing architectures. In the model, servicing operations can both increase and decrease the lifecycle cost of the constellation. On the one hand, servicing extend the life of satellites in the constellation. To maintain the constellation at full capacity, satellites that fail must be replaced by spares launched from Earth, so over the life of the program, if the satellites operate for longer, less spares must be sent into orbit. As less spares are needed, the constellation lifecycle cost goes down. On the other hand, the development and deployment of the servicing spacecraft and orbital depot themselves incur additional costs. If the overall effect of these two cost drivers is to decrease the lifecycle cost of the mission, then servicing is declared valuable.

This is a common approach used to analyze servicing, but it suffers from two limitations. First, it does not account for any benefits associated with satellite upgrades. The analysis implicitly assumes that servicing maintains the constellation at its original capacity, with no regard for changes to the constellation that could increase benefits. Indeed, Joppin notes that maintenance operations alone generally cannot justify servicing programs [16]. Second, the approach explicitly requires the cost of servicing to be modelled. Servicing technology, particularly robotic servicing, is highly uncertain because it is still under development, so cost models of servicing missions also have a high degree of uncertainty. Thus, if the value calculation uses a servicing cost quantity, then the results are suspect. Modellers must aim to avoid including the cost of servicing in order to decrease the uncertainty in the results.

2.7 Previous Telescope Servicing Studies

Most research on the value of servicing has been for programs with revenue-generating satellites, since standard Net Present Value (NPV) calculations can be applied on such programs. Space telescopes do not generate revenue, and as such there are fewer studies that have focussed on the value of telescope servicing. There have been a few notable studies, which are described in this section.

2.7.1 To Service a Space Telescope

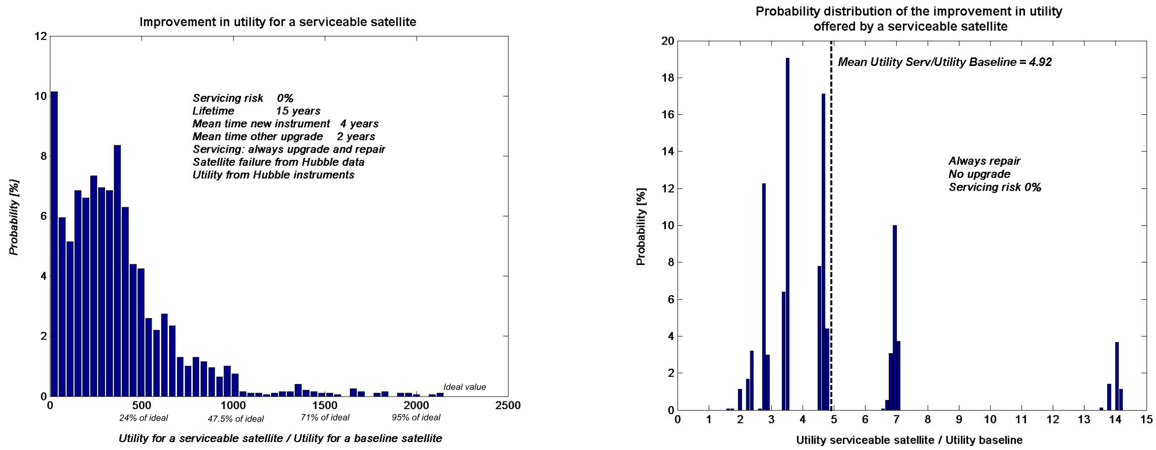
Joppin [17] developed a computer simulation to determine the change in telescope science return over its lifetime with different types of servicing operations. Three servicing types were considered:

- Repair any failed components,
- Upgrade bus instruments when new technology becomes available, and
- Upgrade the science instruments when new technology is available.

Joppin determined the increase in science return with combinations of the above tasks as compared to a baseline case where the telescope receives no servicing.

The simulation captures the effect of both uncertainty and management decision-making on science return. It incorporates four sources of uncertainty: failure time of the telescope, failure of servicing mission, arrival time of new bus technology, and arrival time of new science instrument technology. As well, the simulation captures the actions of program managers through a decision model. The decision model evaluates the program at discrete time steps and determines when servicing missions are sent based on a set of pre-defined decision rules.

The result of the simulation was a set of probability distributions of the cumulative science return of the telescope program for each servicing type. Figure 2-5 shows the probability distribution for two cases. In both cases, the science return is normalized to one, which corresponds to the return of a telescope with no servicing. Case (a) is where servicing includes both repairs and upgrades, and the mean science return is about 300 with a maximum return of 2100. Case (b) is where servicing includes repair only (no upgrades), and the mean science return is about 5 and with a maximum return of 14. Joppin concludes that servicing for the purposes of maintenance alone



(a) Both repairs and upgrades during servicing

(b) Only repairs during servicing

Figure 2-5: Probability distributions of telescope science return (from Joppin [17])

probably does not justify the costs, but servicing to upgrade bus and/or science components provides enormous benefits.

The simulation uses an aggregate reliability curve to provide the probability of operation in the future. Figure 2-6 shows the reliability curve through time, where each data point (t, p) represents the probability p that the telescope will be functional at t years into the future. This reliability data is a first-order approximation of the behaviour of the telescope. An improvement would be to model the individual components. This would make the simulation much more realistic, and engineers could use the simulation to determine the effect of either changing the number of components on the telescope or the design life of each component.

The simulation is a solid foundation for analyzing the benefits of servicing, but it is less useful for aiding decision-making. The simulation has demonstrated that servicing increases both the science return and lifecycle cost of the telescope. This result can be summarized as "if you spend more, you can get more science", which was already known intuitively. Although this result is true, this does not help program managers make decisions. Specifically, managers still have to justify increased spending to enable servicing, and the simulation does not avoid the problem where serviceability is dropped due to budgetary pressures. The difference is that managers now are armed

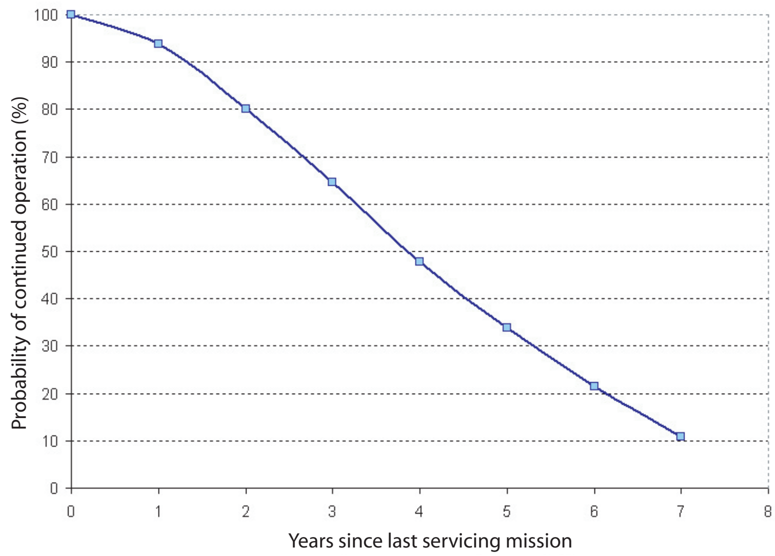


Figure 2-6: Aggregate HST reliability curve (from Joppin [17])

with concrete, quantitative data to demonstrate the benefits of servicing, which can aid in persuading others to keep serviceability in a telescope design. For this reason, Joppin's work is a significant contribution to the understanding of telescope servicing value. This thesis aims to build on her work.

2.7.2 The SAFIR Experience

The Single Aperture Far Infrared (SAFIR) spacecraft is a future telescope where engineers are currently investigating the potential to include servicing. SAFIR will operate at the Earth-Sun L2 Lagrange point and its design features a deployable sunshield to keep the optical telescope assembly (OTA) at extremely cold temperatures (4-7 K) in order to observe light in the mid- to far-infrared spectrum (30-800 μm) [38]. Moe [39] and Lester [40] have identified many potential areas where servicing could provide benefits to the telescope program. A summary of these benefits is shown in Table 2.4.

Table 2.4: Identified Benefits of SAFIR Servicing (from Lester [40])

Service Function	Example	Comments
Replace ACS hardware	Attitude control hardware, such as gyros or computers.	As for HST.
Replace cryo fluids	Liquid helium, cooling line fluids	Although cryo fluids are not now baselined, the ability to replace them and extend mission life could be considered.
Replace cryocoolers	ACTDP-type cryocoolers now baselined for SAFIR.	While instrument cryocoolers (e.g. CADRs) could be packaged with the instrument, observatory cryocoolers need larger scale plumbing connections.
Replace solar panels	Deployable rigid panels, baseline InGaP/GaAs/Ge or thinned Si	New technology, as well as replacement of UV-degraded panels. As for HST.
Replace sunshade	Baseline aluminized Kapton, deploy after attachment	Replacement of UV and micrometeorite degraded panels. Tears, holes, etc.
Replace/Upgrade science instruments	Science or wavefront sensing components	Respond to technology advances in sensors and optical design and aging of the original components.
Replace Propellants	Fluid propellants used for orbit maintenance	In-space fluid transfer technology now being tested (e.g. Orbital Express). Could also just replace entire thrusters instead.
Inspection	Small cameras orbit the observatory and provide imagery	Loose shielding, tiedowns, etc.
Diagnosis	Retrieve sampling coupons placed on the observatory for analysis	Info for next servicing mission. Engineering lessons on contamination.
Replace optical components	Damaged mirror or mirror coating could necessitate replacement of a mirror segment	In-space optical recoating should be considered as an advanced alternative capability
Replace comm. Tx/Rx systems	Ka band w/directional antenna is baselined	Bandwidth upgrades as necessary to match larger sensor formats.
Replace batteries	Baseline Li-Ion	As for HST. Batteries used for safemode only in SAFIR

Mission planners have investigated several potential SAFIR servicing strategies. Since SAFIR will be parked at the Earth-Sun L2 point, it is unlikely to return to LEO for servicing, due to the large delta-v requirement of 3.5 km/s [40]. There are two primary candidate servicing architectures that have been identified [7]. First, SAFIR can be serviced robotically in-situ. This has the advantage of not losing observation time during transit. Second, SAFIR can be transferred to the Earth-Moon L1 point and be serviced either by robots or astronauts. This has the advantage of being closer to Earth, and thus more accessible. With these two strategies, researchers analyzed the engineering design requirements imposed on SAFIR in order to enable servicing. The primary challenges relate to the protection of delicate components such as the sunshields and the cryogenically cooled OTA, as well as the thermal and mechanical interfaces between the servicer and the telescope while docked [39].

As many studies have done in the past, this research into SAFIR servicing has focussed on the benefits associated with servicing without determining whether the costs associated with those benefits could be justified. The example shown earlier in Figure 2-2 comes from a presentation on the value of SAFIR servicing, but they made the common mistake of focussing on benefits instead of value. For example, if servicing SAFIR will cost many billions of dollars, it is unlikely to be of any value, since replacement (launch of a completely new SAFIR) may be a more cost-effective option. Clearly a more comprehensive analysis, which takes into account both costs and benefits, is needed to evaluate whether or not to plan for servicing operations on SAFIR and modify the telescope design.

2.8 The Research Gap

This chapter has summarized the current state of research in the field of telescope servicing, both in the technical methodology (the *how*) and the economic considerations (the *why*). As discussed in Chapter 1, the goal of this thesis is to develop a framework to analyze the value of including serviceability in a telescope design. Many of the elements necessary to conduct this analysis are scattered throughout the literature, and this chapter has highlighted some specific areas that must be addressed to construct the framework. These areas can be summarized in three Research Questions that will be the focus of the remainder of this thesis.

First, it is clear that servicing can provide considerable benefits to a telescope program by maintenance, life extension, and increased science performance. It is also clear that in order to enable telescope servicing, costs are incurred both during the design and operations phase. Both benefits and costs of servicing must be analyzed in order to decide whether or not to incorporate servicing into a new telescope design. The difficulty comes from the fact that the benefits are not measured in the same units as dollars, and this difficulty has stymied efforts to properly determine the value of incorporating servicing of space missions.

Research Question 1

How can the costs and benefits of telescope servicing be compared to determine the value of serviceability?

Second, most research into servicing has focussed on the technology, or *how* to service, and comparatively little research has investigated the value, or *why* to service. No study thus far has taken a comprehensive look at how to justify the incorporation of serviceability in a telescope design. Designers and program managers are thus incapable of quantitatively determining the value of serviceability, which in turn has led to serviceability not being incorporated into many missions.

> **Research Question 2**
> How can the incorporation of serviceability in a telescope be justified?

Finally, the value of servicing comes from the flexibility to perform beneficial actions in the future, as uncertainty is resolved. This can be done either by protecting against losses or taking advantage of advances on the ground. Special analysis techniques are needed in order to accurately account for the value of this flexibility.

> **Research Question 3**
> How can the value of flexibility provided by servicing be analyzed?

The next chapter will discuss general principles that can address these research questions and will be incorporated into the framework.

Chapter 3

Addressing the Gap: Framework Principles

The previous chapter discussed the current state of research into spacecraft and telescope servicing and motivated several research questions to be addressed. In order to develop a framework to analyze the value of serviceability, some basic principles are needed to answer these questions. These principles will be discussed in this chapter, and are summarized below.

- The costs and benefits of servicing will be separated.
- The value of serviceability will be analyzed by comparing a telescope servicing program to a telescope replacement program.
- The value of flexibility will be incorporated by employing decision rule analysis and Monte-Carlo simulation.

3.1 Separating Costs and Benefits of Servicing

The standard method used to determine the value of serviceability in a space program is to calculate the cost savings that would result from incorporating servicing into the program [41]. There are many ways that servicing can decrease the cost of a space program. For instance, components installed on a serviceable spacecraft can be designed for a shorter life, since servicing provides the opportunity to replace failed components through maintenance. Components with shorter design lifetimes have lower costs than those with longer lifetimes [42]. Alternatively, the amount of redundancy built into a serviceable spacecraft can be decreased, since fewer backups would be needed on the spacecraft [43]. The cost savings can be computed by comparing the lifecycle cost of a program without servicing to the lifecycle cost of a similar program that uses servicing. Several studies have demonstrated that programs can achieve significant cost savings through the use of servicing [37, 43, 44].

The difficulty with this approach is that the cost models for servicer spacecraft are highly uncertain. As described in Section 2.5, there have been many testbeds for robotic servicers, but they are still under development. To estimate the cost of these spacecraft, some researchers have used cost-estimating relationships based on historical spacecraft [45]. Unfortunately, servicers are very different than most satellites, so these relationships are not readily applicable. The few cost models that were made specifically for servicers have large error bounds on their cost estimates. As a result, the cost savings incurred through servicing are often less than this uncertainty, as illustrated in Figure 3-1 [44]. Therefore, any analysis that directly use servicer cost models produce results that are often inconclusive.

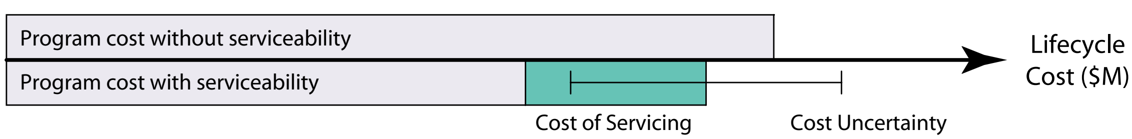


Figure 3-1: Cost savings with highly uncertain servicing costs

Notwithstanding these issues, Saleh [43] noted that if servicing missions were free, a program that used servicing would realize significant cost savings. These savings would not have high uncertainty because they do not include any highly uncertain cost models. His strategy was to exclude the cost of servicing from the overall lifecycle cost of the program. Then, the gap between the cost of the program with and without servicing represents the maximum price that one is willing to pay in order to perform servicing missions, as illustrated in Figure 3-2. Saleh calls this the *customer-centric* approach. Rather than calculating the cost of the servicing mission (which is the cost from the servicing provider), the focus is shifted to how much cost the *customer*, or the organization that purchases servicing, is willing to tolerate in order to perform servicing missions.

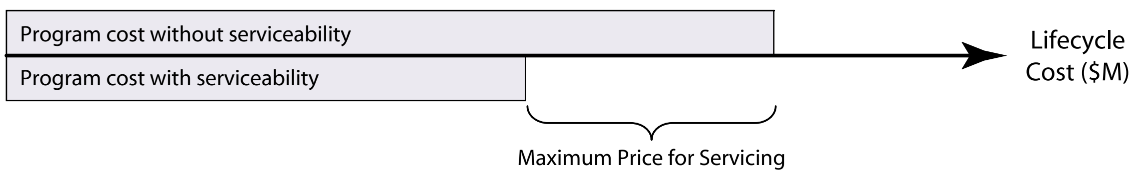


Figure 3-2: Cost savings as the maximum price of servicing

This maximum price can be compared to the quoted price of a servicing mission from a contractor. Consider the perspective of a telescope developer such as NASA, which acts as a customer for servicing. If the quoted price is less than the maximum price, then servicing is valuable to NASA: the servicing missions can be purchased within budget, so the telescope should be designed with serviceability to take advantage of the benefits of servicing. If the quoted price is higher than the maximum price, then either the serviceability is not valuable (because NASA would be unable to pay for servicing missions, so there's no point in adding serviceability), or the budget should be increased to pay for the missions. In the latter case, the framework provides NASA data on how much the budget should be increased, and what the expected science return is in exchange for the budget increase.

There are three main advantages of separating the servicing cost from the rest of the lifecycle costs of the spacecraft. First, as discussed earlier, excluding servicing costs dramatically reduces the level of uncertainty in the overall estimate of lifecycle cost. Second, the maximum servicing price calculated by this method is independent of the servicing architecture. The price is valid for whichever servicing type is available, whether human or robotic [45]. Third, it provides quantitative information to decision makers on how much money should be spent on servicing. This will help justify the additional cost of serviceability in telescopes, which will make it more likely that serviceability will be retained in the face of budgetary pressures.

3.2 Using Program Comparison to Calculate Value

The method described above requires a telescope program without servicing against which the servicing program will be compared. One possibility is a program with a telescope designed to last for the entire nominal mission duration. While this may be possible for space programs with shorter lifetimes and less complexity, for large space telescopes with long mission durations of 20 years or more (such as HST), it is often infeasible to design a telescope to operate for that duration without intervention.

The feasible alternative to servicing a spacecraft is to replace the spacecraft with a new copy when the initial spacecraft fails. This was contemplated for the Solar Maximum Mission (SMM), which was launched in 1980 for a two-year mission [46]. Midway through its life, the pointing system onboard SMM failed prematurely, preventing it from accurately pointing towards the Sun. NASA had two options: replace the spacecraft or service it via the Shuttle, and found that the cost of servicing would be 25% that of replacement [45]. NASA decided to send the Space Shuttle *Challenger* on a mission to service SMM. In the end, servicing prolonged the life of SMM and enabled it to operate until December 1989 [46].

3.2.1 Comparison Cases

This framework evaluates programs with servicing and replacement as two separate *cases*, which are strategies employed in programs to ensure the telescope is operational throughout its nominal mission duration. The two cases in the framework are:

1. Replacement case

2. Servicing case

The Replacement case involves two identical telescopes over the program lifetime. At the start of the program, the first telescope is launched and operates until component failures cause it to become inoperative. When this occurs, a second telescope is constructed and launched. This telescope is based on the original design, so no development costs are incurred, and the cost of the second telescope is solely the cost of construction. The telescope is identical except for the science instrument, which is upgraded with the newest technology.

In contrast, the Servicing case involves a single telescope that would operate throughout the entire program lifetime. The telescope would be periodically serviced to keep the bus components operational and install new science instruments with the latest technology to increase the telescope science return. In order to allow servicing to take place, the telescope must be designed with serviceability. As discussed in Section 2.5, these modifications will increase the initial cost of the telescope, but servicing will preclude the need for a second copy of the telescope to be launched. The major cost tradeoff is between building two telescopes (Replacement case) versus building a serviceable telescope and paying for servicing missions (Servicing case).

Within these two cases, there are several ways that program managers could incorporate replacement or servicing missions into the program, called *subcases*. The two subcases, applicable for both the Replacement and Servicing cases, are:

1. *As-Needed* subcase

2. *Fixed Schedule* subcase

In the *Fixed-Schedule* subcase, missions are executed at predetermined times throughout the mission life. The advantage of a fixed schedule is that the program can get regular updates and ensure that any problems on the telescopes are fixed or mitigated before they cause failures[1]. In contrast, in the *As-Needed* case, missions are executed only when the telescope fails or is near failure. The advantage of sending missions only as needed is that they can be delayed as long as possible, which provides

[1] From a programmatic standpoint, a fixed schedule for replacements or servicing missions may also be easier to budget for. This is not included in the analysis

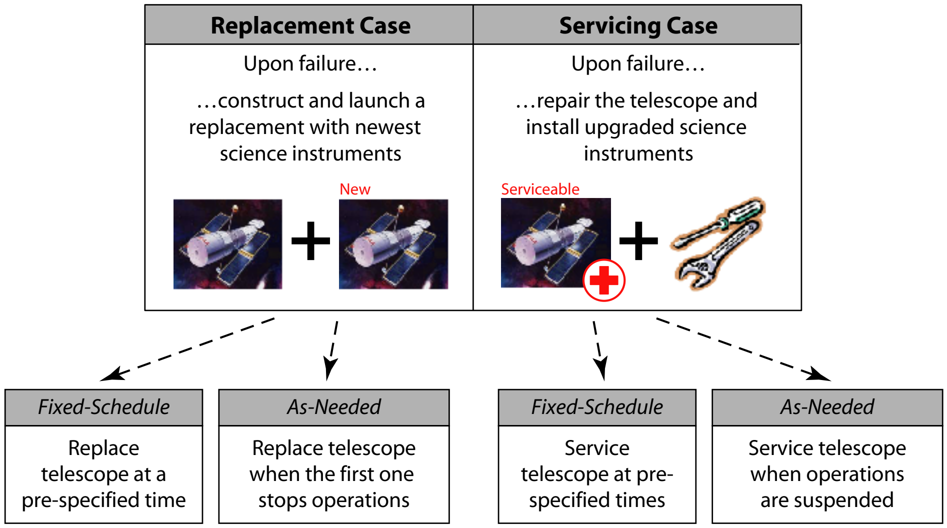


Figure 3-3: Summary of cases and subcases in the framework

two benefits. First, since telescope instrument technology rapidly progresses, servicing or replacing the telescope later in its life will result in a more powerful science instrument being used. Second, delaying missions will ensure that the telescope is not attended to unnecessarily. Since both replacement and servicing are invariably expensive, program managers want to ensure that sending the mission was worthwhile based on the state of the telescope. For example it would be unacceptable to send a mission only to repair one defective gyroscope; rather, the mission should be sent only when the telescope needs significant maintenance. The cases and subcases in the framework are illustrated in Figure 3-3.

Note that the *Fixed-Schedule* Replacement subcase has the opportunity to have two telescope operating at the same time. If the replacement telescope is launched before the initial telescope has failed, then both will operate until the initial telescope fails. The framework allows for two telescopes to operate simultaneously, and the benefits and costs of both telescopes are counted.

The comparison method as described above has one primary limitation: there is no *do-nothing* option. There is no way the analysis would come to the conclusion that neither replacement nor servicing should be implemented. The analysis is set up to calculate the value of servicing as compared to the baseline Replacement case.

The only conclusions one could draw would be one of the following:

1. There is sufficient budget to pay for servicing missions, so incorporate serviceability into the telescope design.

2. There is insufficient budget to pay for servicing missions, so the telescope will be replaced instead. Do not incorporate serviceability into the design.

This assumption exists because there was insufficient data to analyze the case where there is a single telescope that operates for the entire, extended mission duration. Indeed, as of 2007 no telescope has operated in space for decades without intervention. Nevertheless, this limited scope can provide important insights into the value of serviceability and how to proceed to determine its value.

3.2.2 Comparison Parameters

To properly compare programs, the comparison method must have three properties. First, the comparison must be made on a common baseline[2]. If one compares two programs that are not equal on some level, then the comparison itself is meaningless. For example, comparing the costs and benefits of a short-duration Earth observation program and a long-duration infrared telescope program cannot produce meaningful results. Second, the comparison must use a metric that can be used to discriminate between programs. This means the metric must be applicable to the program under analysis: it is not useful to use number of images as a metric if the programs produce spectra rather than images. Finally, the metric must have a sense of direction; that is, more attractive programs are defined by either higher or lower metric values.

One possible method to compare telescope programs is to minimize the lifecycle cost for a given science return. This is often the case for scientific space missions with a well-defined, quantifiable science objective. For example, the goal of the Terrestrial Planet Finder (TPF) project was to observe a minimum number of solar systems to attempt to detect extrasolar planets[3], and to observe certain properties for each

[2] This is colloquially referred to as comparing "apples to apples"

[3] Extrasolar planets are planets outside our own solar system

detected planet [47]. This science objective is a definitive goal, and when the goal is met, the mission could be called a *success*. TPF could be certified as successful once the spacecraft has succeeded in studying the required number of systems and has taken the required data for each system. Furthermore, this science objective is one that can be designed towards. Engineers designing the spacecraft can tailor the amount of consumables and expected life of components so that the system can meet the objective with a high degree of certainty. In the design phase, different architectures are compared based on their ability to meet the science objective, and the best architecture is the one that can meet the objective for the least cost.

This comparison method is not appropriate for general space telescopes since they do not have set science objectives that must be reached to define success. Both space and ground observatories exist to provide a platform for astronomers to continually gather information for a wide variety of studies. There is no quantifiable point at which a telescope is considered successful. Indeed, scientists define a successful telescope as one that has contributed (and continues to contribute) towards a body of research. The appropriate metric for the telescope is science return, where a better telescope provides more science return over its life.

For defined-science missions such as TPF, the goal was to minimize the cost required to achieve a set amount of science return over the mission. For a general space telescope, a more appropriate goal is the converse; namely, maximize the science return for a fixed lifecycle cost. This is reasonable because space telescope programs are often subject to cost constraints rather than fixed science return. As well, a fixed nominal mission duration is an important baseline quantity. Astronomers want to be assured that they will have an operational space observatory for at least a set amount of time. As well, telescope program planners (such as NASA) want to know the lifetime of a telescope in order to fit it into a larger observation program.

In summary, this framework will compare the cases and subcases based on the amount of science return produced, where more science return is better. The baseline for comparison is fixed lifecycle cost, which comes from the Replacement case, and nominal mission duration, which is a specified parameter.

3.3 Incorporating Flexibility and Decision-Making

Section 2.3 described how the value of servicing comes from the flexibility to react to uncertain events in the future. Servicing provides the ability to make decisions as this uncertainty is resolved [45]. Standard economic evaluation techniques such as NPV cannot capture this flexibility, so an alternative modelling method is needed to incorporate decision making into the value analysis. There are several potential methods available that are currently used in industry and literature. This section will discuss each of these in turn.

3.3.1 Methods from Financial Options

Subsection 2.2.3 demonstrated that serviceability can be considered as a real option in a space telescope. Real options are closely tied to their analogues in the financial world, where elaborate theories have been developed to determine the appropriate price of a financial option. Can these financial analysis techniques be used to evaluate the real option of spacecraft servicing?

The fundamental theory of options pricing is the Black-Scholes model, initially proposed by Merton in 1973. The theory considers the performance of an underlying asset by assuming the price follows random Brownian motion with a known volatility σ, where volatility is defined by the variance in the asset price [12]. The model consists of a set of partial differential equations whose solution under various conditions represents the value of an option. The option value represents the maximum price that one should be willing to pay to purchase the option; otherwise the expected return would be insufficient to recover your costs. A more complete treatment of the Black-Scholes model is provided by de Neufville [11] and Trigeorgis [12].

There are two primary difficulties in trying to apply financial option pricing techniques to the spacecraft servicing. First, the model assumes that the benefits of the option are monetary, since the theory is designed to analyze financial instruments such as stocks and bonds. It is therefore difficult to apply the techniques to the real option of space telescope serviceability since the benefits are non-monetary. Second, and more fundamentally, many options pricing techniques (including the Black-Scholes model) rely on the existence of a *replicating portfolio*. A replicating portfolio is a set of assets whose value tracks the value of the option [12]. Selecting the replicating portfolio for financial options is straightforward: it consists of the financial instrument for which the option was purchased. For real options, there may be no underlying asset that is appropriate. In the case of a space telescope program, which is run by a university or government agency and produces science data, there is no conceivable financial asset whose value rises as the science return from the telescope increases. Thus financial options theory, although attractive and mathematically rigorous, is not appropriate for analyzing servicing as a real option.

3.3.2 Decision Tree Analysis

Another method that can be used to analyze the value of servicing is *decision tree analysis*. A *decision tree* is a sequence of decisions at discrete points through time called *decision nodes*. Between each decision node is a *chance node*, which is where uncertain events may occur. The decisions are not made at the start of the program; rather, they are made as time progresses and uncertainty is resolved [45]. Each decision and chance node pair represents one time step. For example, if a decision tree had a time step of one year, then a decision node represents choices that could be made at year one of a project, and the chance node captures all random events that can occur within one year of the decision. From each node, branches extend forward that represent different paths the system can take, whether they occur due to decisions or random events. A sample decision tree is shown in Figure 3-4.

At each decision node, managers must choose which decision to make, and the choice is determined by which decision will maximize the expected value of the node. The expected value of a decision node at time step n is the weighted sum of the expected values of each of the downstream decision nodes at time step $(n+1)$, weighted by the probabilities of the chance events. In this way, the analysis of the decision tree is recursive, and an optimum path can be found to maximize the expected value at the start of the project. The net result of a decision tree analysis is the overall

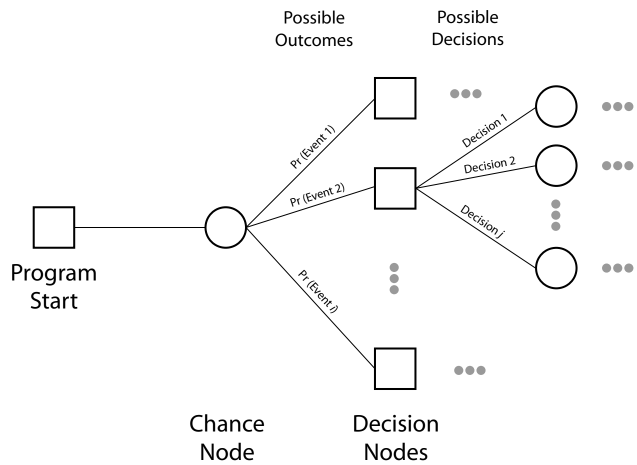


Figure 3-4: Structure of a decision tree

NPV of the project and the optimum set of choices that should be taken in order to achieve this value.

The difficulty with decision trees is that they can become prohibitively large when there are many potential chance events and/or decisions that can be made. The number of nodes in a decision tree is exponential in the number of time steps, decision branches, and chance branches, so even a small number of branches at each node can result in a very large tree [48]. This becomes an issue with space telescopes, since there are a large number of branches from each chance node. Telescopes have many components onboard, and each has a probability of failure, so there are many combinations of random events that can occur, resulting in many chance branches. Evaluating such a tree becomes very computationally expensive.

3.3.3 Decision Rules and Monte-Carlo Simulation

To avoid the computational complexity of analyzing a large decision tree, the system under investigation can be *simulated* through time. The simulation discretizes the program lifetime into time steps, and at each time step, the set of chance events are evaluated. Effectively, the simulation "rolls the dice" to determine the outcomes of chance events at every step. The difference is how decisions are made. Whereas a

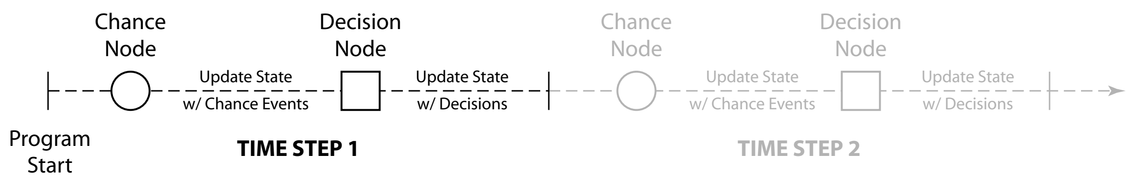


Figure 3-5: Steps in a decision rule analysis

decision tree analysis computes the optimal strategy to determine each decision, the simulation makes decisions on the fly based on a set of pre-defined *decision rules*. These are if/else statements that are evaluated at discrete times to determine which choice to make. Decision rules are often formulated as thresholds that must be met in order for an action to be taken. The system being simulated is represented by a *state*, which captures the effect of random events and decisions during the simulation. This state may change based on probabilistic events and decision-making. When the program terminates, the simulation calculates the net result of all the decisions made and the states that the system passed through.

Each time step in the simulation has four stages, as shown in Figure 3-5.

1. All chance events are evaluated to determine their result.

2. The system state is updated to reflect these chance events.

3. Based on the current system state, the decision rules are evaluated to determine the actions to be taken.

4. The system state is updated to reflect the actions performed.

The simulation moves on to the next time step, where these four stages are repeated. This continues until the system reaches the end of its lifetime, at which point the simulation terminates.

The simulation can be demonstrated through a simple thought experiment. Consider a factory that has an option to expand capacity if demand increases. The state associated with the factory is its capacity and its production rate. The production rate is subject to random failure events, and the market demand is probabilistic. A possible decision rule is to increase capacity if demand increases by 200% above the initial demand. Each time step in the simulation would progress through the four stages as follows:

1. Determine the current market demand and if any failures occured in the plant.
2. Update the system state (production rate) if these failures occur.
3. Evaluate the decision rule: has the demand increased sufficiently to warrant a factory expansion?
4. Update the system state (capacity) if the option to expand is exercised.

The result of the simulation is dependent on the sequence of events that occurred during the simulation. In other words, the result is specific to the particular set of dice rolls that happened during the simulation. To obtain a more general sense of the option value, the simulation must be repeated many times in order to remove the dependence on particular events. This method is called *Monte-Carlo analysis*[49]. It was named after the famous casino in Monte Carlo because the analysis could be thought of as a long series of dice rolls. The number of iterations that is required in a Monte-Carlo simulation depends on the amount of elements in the system that are subject to uncertainty. With more uncertain elements, more iterations are required to average out the effects of each of these elements.

After running the simulation many times, properties of the system can be described as a probability distribution. The probability distribution can be visualized using a histogram, an example of which is shown in Figure 3-6. The histogram divides the results for each property into bins, and provides the probability that the property value falls within each bin. A histogram constructed in this manner is strictly speaking a *probability mass function* (pmf) [50], but in this thesis it is called a *probability distribution*. An example is shown in Figure 3-6. The third bar from the left signifies there is a 30% chance that the parameter falls between 20 and 30, and the probabilities of all bars add up to 100%. The dashed line denotes the mean value.

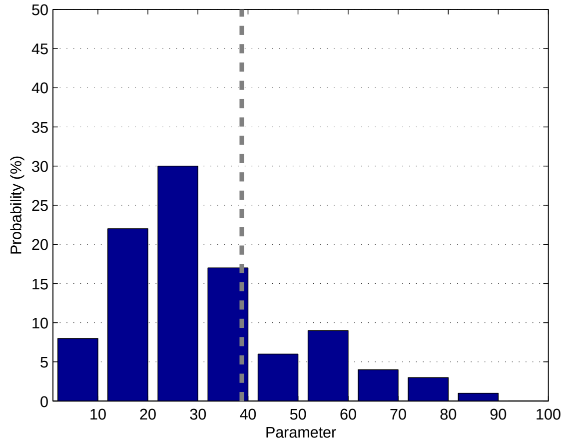


Figure 3-6: An example probability distribution displayed as a histogram

3.4 Principles Summary

This chapter has discussed how the three research questions in Chapter 2 can be addressed by three main principles. From these principles came six major ideas:

- To avoid the issue of uncertainty in servicing cost models, the cost of servicing missions is excluded from the lifecycle cost of a telescope program.
- The value of servicing is the maximum price that the customer is willing to pay in order to perform servicing missions.
- The servicing program is compared to a program where the telescope is replaced with a copy of the original design.
- The comparison is made on the common baseline of fixed lifecycle cost and fixed nominal mission duration.
- The metric for comparison is science return, where more science is better.
- The value of flexibility will be included by using decision rule analysis and Monte-Carlo simulation.

The next chapter will discuss how these ideas will be incorporated into a simulation framework to determine the value of telescope serviceability.

Chapter 4

Raising the Scaffolding: Framework Construction

The previous chapter detailed the principles that will be used in the framework to resolve some of the major issues surrounding the evaluation of telescope servicing. With these principles established, this chapter will detail the construction of the framework itself.

The framework consists of five models that capture important aspects of the telescope program. Each model addresses a necessary element that must be included when calculating the value of servicing.

- *Benefit Model*: Determines the science return of the telescope through time.
- *Cost Model*: Determines the total lifecycle cost of the program.
- *Telescope Model*: Represents the telescope as an simplified set of components that can fail and are affected by servicing and replacement operations.
- *Stochastic Model*: Evaluates uncertain quantities in other models, such as failure times and risk probabilities.
- *Decision Analysis Model*: Simulates the decision-making of a program manager regarding when to send a servicing or replacement mission.

This framework provides the basis for creating a computer *simulation* of a telescope program; that is, a simulation is the framework implemented using numerical models. When this framework is used during an actual design project, each of these five models would be implemented using code that captures design information and company-specific knowledge held by the organization that is considering telescope servicing. This chapter will discuss MATLAB functions for each of these five models to demonstrate how the framework can be implemented as a simulation. It is important to remember that the emphasis in this chapter is placed on the framework itself, not on the simulation code used to implement the framework. Chapter 5 will use the simulation in a case study to illustrate how the framework can analyze the value of serviceability.

To avoid confusion, consistent terminology is used in this section to describe various parts of the simulation. A *model* is an individual module that models a distinct process, such as a cost source or failure occurrence. The collection of models that are used together to calculate the value of servicing is the *simulation*. One iteration of the simulation is a *run*, and the set of runs that is used for analysis is called the *simulation results*.

4.1 Measuring Benefits: Telescope Science Return

One of the primary benefits of servicing is the ability to install new instruments and thus increase the science return of the telescope. To capture this phenomenon, the framework requires a quantitative metric to measure telescope science return, and a method to model the changes of the science return through time. There are three metrics that are widely used in the telescope community to measure science return: productivity rate, number of papers generated, and discovery efficiency. Each of these three metrics is investigated below, with the aim of determining which is most appropriate for use in the framework.

4.1.1 Productivity Rate

The simplest metric to measure the science return of a telescope is the *productivity rate*, which is the number of images captured by the telescope per unit time period. The overall science return from the telescope is therefore the integral of the productivity rate over the life of the telescope [51]. The advantage of using productivity rate is its simplicity. It is relatively straightforward to determine the amount of images that can be captured by the telescope based on its design. Furthermore, the productivity rate is affected by the health of the telescope: if operations are affected due to failures or wearout, the number of images that can be taken over a given period decreases, which decreases the productivity rate.

However, there are two primary disadvantages to this metric. First, the productivity rate has no sense of the quality or worth of each image. There is an implicit assumption that an image gives the same amount of information, whereas image data has properties such as sensitivity, pixel size, field of view, etc., which are not captured with this simplistic approach. If the science instruments onboard are upgraded via servicing, these parameters would change but the metric would remain the same. Second, the metric is not applicable for instruments other than cameras because it measures the rate of capture of discrete images. Instruments such as photometers and spectrometers, which are commonly installed on space telescopes, do not capture discrete images; instead, they collect data on intensity of light at particular wavelengths, which cannot be easily converted into an analogue of a discrete image.

4.1.2 Number of Papers

Astronomers may argue that the science generated by a telescope is not necessarily the raw data that is observed; rather, it is the scientific papers that are written based on that data. Accordingly, a metric commonly used by scientists to evaluate telescope programs is the *number of papers*. The number of papers resulting from a telescope can be counted in a variety of ways.

- Number of papers published in refereed journals (as shown in Figure 4-1)
- Number of papers presented at conferences
- Number of citations to a data set generated by a telescope
- Number of *high-impact papers*, defined as papers that are among the 200 most cited refereed papers in the field [52].

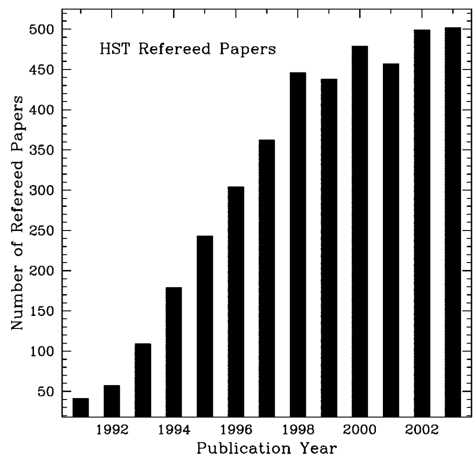


Figure 4-1: Number of refereed papers based on HST data by publication year (from Meylan [52])

In all these forms, program managers often use this metric as a predictor for how productive a telescope will be in the future based on its productivity in the past. This is used to persuade funding agencies to grant the telescope program additional funding to continue its mission.

Although this metric arguably best reflects the actual science return of a telescope, its primary difficulty is that it cannot be computed *a priori*; that is, before the telescope has been launched. The metric can only be evaluated after the telescope has been in use for some time, since there is a delay between data collection and production of research papers. For this reason, the metric has a systemic bias towards telescopes that are older or have been operating for longer durations, since the data set is larger and scientists have more time to analyze it [53]. Furthermore, data comes from specific instruments, but for telescopes with multiple instruments, it is often unclear how to assign papers or citations to specific instruments. Papers are often written on data collected from different instruments so as to use information in different wavelengths on the same observed object [52]. For these reasons, using the number of papers metric is not useful in the initial design phase, since it is restricted to an *a posteriori*, or after-the-fact, analysis.

4.1.3 Discovery Efficiency

Another science metric more prevalent in the design of telescope instruments is the *discovery efficiency*, which is defined as the product of two instrument properties:

- *Field of view*: the angular viewing area (arcsec^2) visible to the instrument.

- *Throughput*: the fraction of photons detected by the instrument [54].

This captures the intrinsic value of data collected by the instrument: a larger field of view and higher sensitivity represents more information, which corresponds to increased discovery efficiency. In contrast to the number of papers metric, discovery efficiency can be calculated *a priori* because it depends only on instrument specifications that are known prior to launch (and perhaps before the instrument is built).

This metric has two limitations. First, it can only be used with cameras, since instruments such as spectrometers cannot be described by a field of view. Second, discovery efficiency is often a function of wavelength, because science instruments are often optimized for specific wavelength bands of interest (such as ultraviolet or infrared). Thus, there may not be a single number that characterises the discovery efficiency of an instrument. Figure 4-2 shows the discovery efficiency curves for several HST science instruments. The problem of wavelength dependence can be avoided by selecting the discovery efficiency at specific wavelengths to be the representative value. For example, Joppin used two representative wavelengths (400 nm and 700

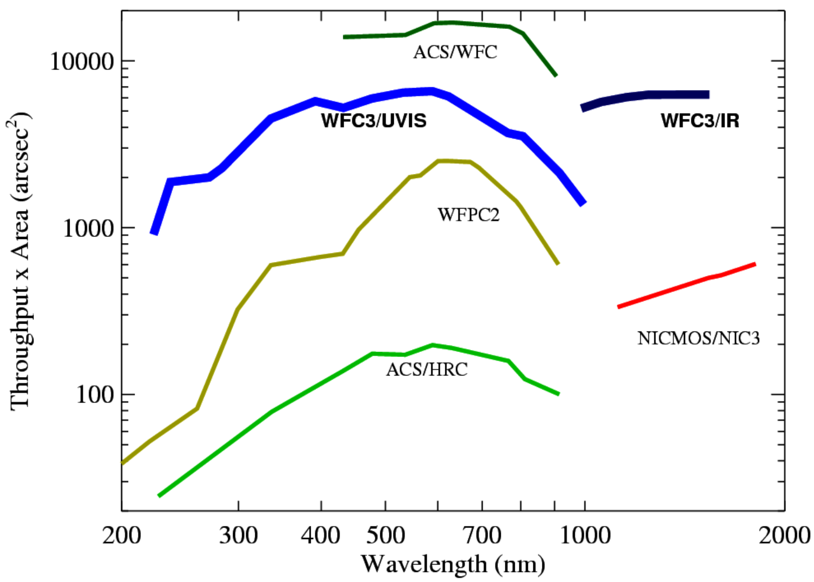


Figure 4-2: Discovery efficiency of select HST Instruments (from STScI [55])

nm) to determine the discovery efficiency [53]. Alternatively, one can use the average discovery efficiency as the representative value.

In the end, the discovery efficiency metric was selected for the telescope benefit model because it is widely used in the community, it can be calculated *a priori* from the design, and it captures some sense of the science worth of the instrument. The analysis will be restricted to telescopes with cameras installed.

4.1.4 Modelling Technology Advancement

Servicing missions can upgrade instruments on a telescope and increase the discovery efficiency of the complement of installed instruments. Servicing can become extremely valuable because instrument technology has progressed exponentially, and future instruments can be orders of magnitude better than the initial ones installed.

This progression has been observed for *charge-coupled devices* (CCDs), which are widely used on telescopes as image capture devices. A CCD consists of arrays of metal-oxide-semiconductor (MOS) capacitors that can detect the excitation of electrons caused by incoming photons [56]. Similar to computer chips, CCD technology has improved dramatically in recent years. The size of CCD chips has decreased and the number of detectors (pixels) per unit area has increased exponentially [57].

A major difficulty in developing a future technology model is how to calculate the numerical value for the discovery efficiency. Quantifying the discovery efficiency requires design specifications for a particular instrument, which is not easily generalizable. Fortunately, as described in Chapter 3, this framework investigates the marginal change in science return between the Replacement and Servicing cases. It is therefore not important to express the *absolute value* of the discovery efficiency, but instead express the *relative change* of discovery efficiency from an older to a newer instrument. To do so, the initial instrument is normalized to a non-dimensional discovery efficiency of 10, and future instruments are assigned a discovery efficiency based on how much better the instrument is compared to the initial one. For example, if the field of view of the replacement instrument is four times larger than that of the initial instrument, then the discovery efficiency of the replacement is set to be 40.

The discovery efficiency of new instrument technology through time was modelled using a power law as shown in Equation 4.1, where DE is the discovery efficiency of the best instrument technology at t years after program start. Note the discovery efficiency is normalized to 10 at $t = 0$ years.

$$DE = 10e^{pt} \tag{4.1}$$

The trend depends on a parameter p, which controls how quickly technology evolves over time. The value of the parameter depends on the specific technology trend data that is being used. The case study in Chapter 5 demonstrates the application of example trend data to the discovery efficiency power law.

4.1.5 Modelling Telescope Science Return

The previous subsection discussed the advancement of instrument technology through time, but this represents the technology that is available on Earth. The benefits of advanced technology are transferred to the telescope only when the telescope is serviced or replaced, so the framework requires a system to track the discovery efficiency of the telescope instruments through time, independent of the technology available on the ground.

In this framework, the discovery efficiency of the instrument follows several simple rules:

1. Without servicing or replacement, discovery efficiency remains constant through time. Experience has shown that science instruments on space telescopes degrade by less than 1% per year[1] [19], so the discovery efficiency can be approximated as constant.

2. Increases in discovery efficiency can only occur if a new instrument is installed, either via a servicing or replacement mission.

3. When a new instrument is installed, the discovery efficiency of the new instrument lags the technology available on the ground.

This last point deserves further clarification. When a telescope is upgraded, it does not get the best technology available at the time of upgrade. The technology used in the instrument does not match the state-of-the-art at the time of servicing because the instrument must be constructed in advance of the servicing mission. If the instrument is installed at time T, then the instrument discovery efficiency is frozen some time S before the instrument is sent to the telescope. For example, if a servicing mission is launched in year $T = 5$ of the telescope program, but the instrument takes $S = 1$ year to build, the discovery efficiency of the instrument is frozen at the technology level available in year four. The instrument construction latency S is represented in the simulation as a parameter called `upgrade_latency`.

Using these rules, the simulation tracks the telescope discovery efficiency throughout its entire lifecycle, as shown in Figure 4-3. The jumps in the telescope discovery efficiency represent the installation of new instruments, and the gap between the state-of-the art and technology discovery efficiency is due to instrument construction latency.

[1]Note this does not include failures that permanently disable the instrument.

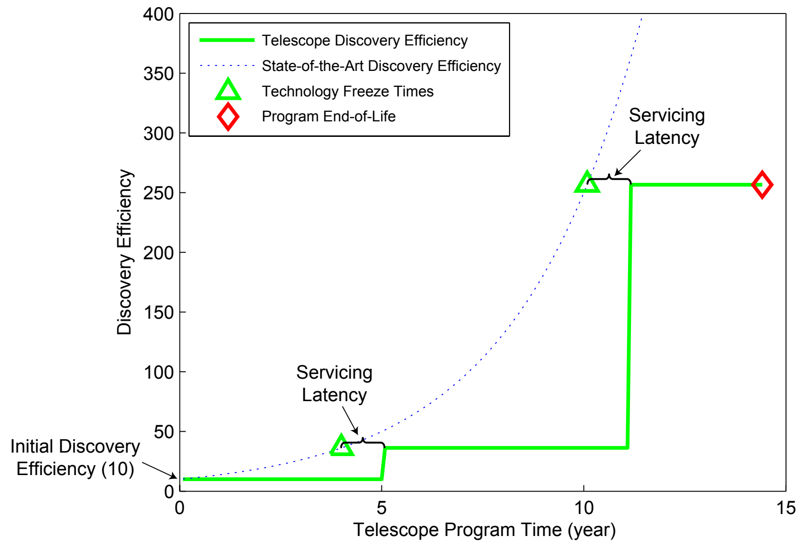


Figure 4-3: Discovery efficiency of state-of-the-art technology and the telescope

The integral of the telescope discovery efficiency curve is called the *cumulative science output* (CSO) and represents the total science return of the telescope. This will be used as a metric in Section 4.6 to compare the relative merits of Servicing and Replacement cases.

4.2 Measuring Costs: Cost Models

In order to build and operate a space telescope, there are four primary sources of cost that will be considered in this framework:

1. *Initial Cost*: The cost of program definition, research, development, fabrication, and testing of the initial telescope, including all engineering components.

2. *Science Instrument Cost*: The cost of the instrument that generates science data, and constitutes the payload of the telescope.

3. *Launch Cost*: The cost of launching the telescope into its operational location.

4. *Operations Cost*: The annual cost of operating the telescope from a mission control station on Earth.

Once the initial telescope is deployed, both the Replacement and Servicing cases incur additional costs. As described in Section 3.2, the Replacement case requires the construction of a new telescope from the same original design (but with an upgraded science instrument), plus launch and operations costs. The Servicing case requires the purchase of replacement components and new science instruments that are installed onboard. In addition, before a telescope can be serviced it must be built to enable servicing, which increases the initial costs. Finally, as described in Section 3.1, the cost of the servicing missions themselves is not directly modelled and will be included when the two cases are compared.

When this framework is implemented by an organization during telescope design, they can use proprietary, high fidelity cost models; however, to show how these models would be used in a simulation, this section will describe some first-order cost models that were developed for demonstration purposes only.

The list of cost models are summarized in Table 4.1. The input to most of the cost models is *MLEO*, or Mass to Low Earth Orbit. This metric was selected because several NASA cost estimation tools are functions of launch mass, and this makes the costs become functions of the size of the telescope. The output of all cost models is in FY2000 US dollars.

Table 4.1: First-order cost models and their inputs

Cost Model	Input
Initial Telescope	MLEO
Instrument	MLEO
Launch	MLEO
Operations	Initial Cost
Replacement Telescope	MLEO
Replacement Components	Number of Components
Serviceability Cost	MLEO

4.2.1 Initial Telescope Cost

All costs incurred before the initial telescope is launched are classified as *initial costs*, which includes research, development, construction, and testing,. In the simulation, initial costs are calculated using the Advanced Missions Cost Model (AMCM) [58]. This provides rough order-of-magnitude estimates of the cost of constructing a variety of vehicle types for different types of missions. It is implemented as a web-based application, which was ported into MATLAB for use in the simulation.

The AMCM, shown in Equation 4.2, is a power law defined by a set of parameters.

$$InitCost = a \times Q^b \times W^c \times d^s \times e^{0.01} \times B^f \times g^D \times INF91 \tag{4.2}$$

Different vehicle types are represented with different parameters (represented as lower-case letters). To analyze telescopes, the law was used with the *Physics and Astronomy Spacecraft* parameter set, which was generated by fitting a curve through the historical set of spacecraft that performed physics and astronomical experiments. In addition, the *INF91* parameter is a multiplicative factor that inflates the cost data from 1991 dollars to 2000 dollars. These parameters are summarized in Table 4.2.

In addition to the fitting parameters, there are also three user-defined parameters:

- *Block Number (B)*: The level of design inheritance in the system. A block number of n implies that the telescope in question is the nth iteration using the same design. The telescope is assumed to be an all-new design, so $B = 1$.

Table 4.2: Parameters for the Operations Cost model

Parameter	Value	Parameter	Value
a	0.000504839	f	-0.355322218
b	0.594183076	g	1.554982942
c	0.653947922	s	2.170
d	76.99939424	*INF91*	1.414
e	1.68051e-52		

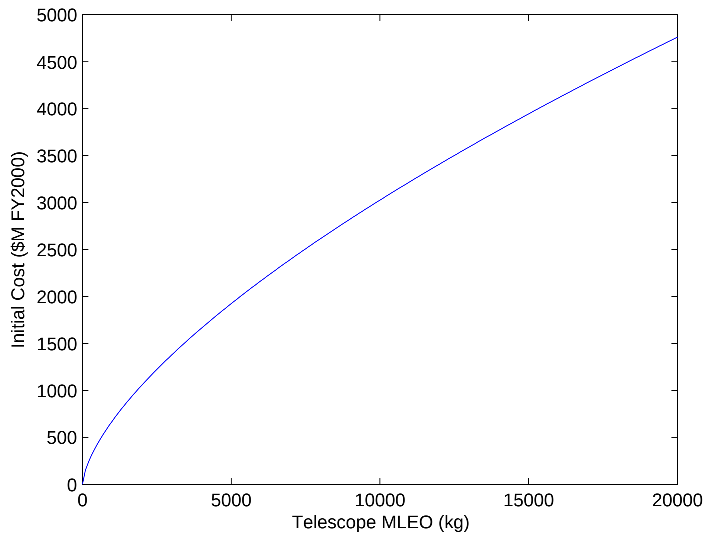


Figure 4-4: Initial Cost as a function of MLEO

- *Difficulty (D)*: The level of programmatic and technical difficulty anticipated for the new system compared to previous, similar space systems. Difficulty is rated on a scale of [-2 -1 0 1 2], corresponding to [very low/low/avg/high/very high] difficulty. Telescopes are highly complex spacecraft and the difficulty is at least "high", so $D = 1$.

- *Quantity (Q)*: The total number of units to be produced. The initial telescope is built only once (not including any replacements, which are treated separately), so $Q = 1$.

It gives the initial cost of a telescope as a function of the telescope launch mass (MLEO), represented in the equation as W (pounds). With all the fitting and program-specific parameters specified, the model gives an estimate for the initial cost of the telescope given its initial mass. Figure 4-4 shows the output of this model for telescopes with MLEO between 0 kg and 20,000 kg.

4.2.2 Instrument Costs

Spacecraft can be generally separated into two segments: the *payload* (the components that perform the intended function of the spacecraft), and the *bus* (the components that supports the payload). In this framework, the bus consists of all engineering components and the optical telescope assembly (mirrors, etc), and the payload consists of the science instrument(s). The bus construction cost is calculated with the Initial Costs model, but a separate cost model is needed to model the cost of the science instruments.

Instrument costs can increase or decrease through time depending on how technology evolves. Costs may increase due to the use of highly advanced technology in an instrument, or costs may decrease as previously expensive components come down the manufacturing learning curve. The experience of HST, however, shows that there was no general trend for the cost of installed instruments, despite the fact that they became increasingly powerful through time. Table 4.3 shows a selection of instrument costs throughout the HST program [17, 59]. The later instruments, despite being far more advanced than the ones initially installed on HST, do not have correspondingly large cost increases. On the contrary, in some cases the instrument cost decreases from one generation to the next. In addition, the instrument cost can be dependent on the size of the telescope, as larger telescopes can support larger and more complex instruments.

In the simulation, the cost of the telescope is defined to be directly proportional to telescope mass and constant through time. From HST, the mass of the telescope is approximately 11,000 kg and has a historical cost per instrument of approximately $100M (in constant dollars). The resulting instrument cost model as a function of telescope MLEO is shown in Figure 4-5. As for the lack of time-dependence, HST has shown that the cost of the instrument can be approximated as constant. Of course, the model can be improved dramatically with more accurate relations for projected future instrument costs.

Table 4.3: Cost of selected HST Instruments (WFC3 to be launched in 2008)

Parameter	Year Installed	Cost
WFPC1	1990	\$130M
WFPC2	1993	\$127M
NICMOS	1999	\$105M
STIS	1999	\$125M
ACS	2002	\$75M
WFC3	2008	\$83M

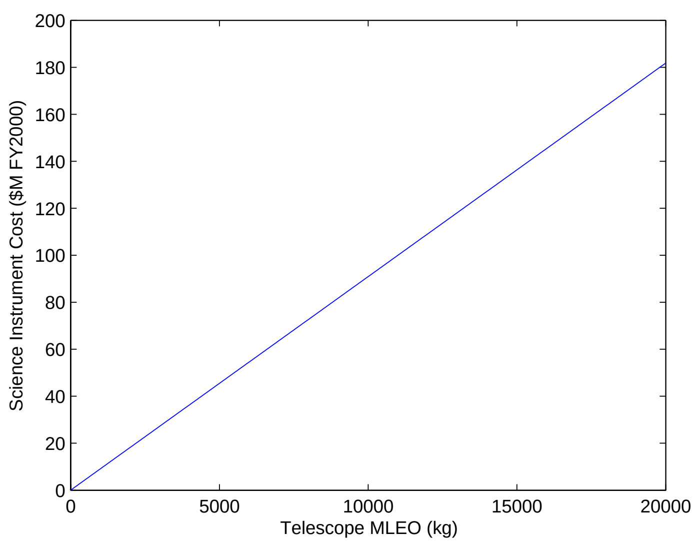


Figure 4-5: Instrument Cost as a function of MLEO

4.2.3 Launch Costs

Once the telescope is constructed and equipped with science instruments, it must be launched from Earth to its operational location. For the simulation, a launch cost model was developed using cost data from current launch vehicle systems [60, 61]. Launch vehicles are often available in a *family*, which is a set of configurations for a launch vehicle to support different MLEO quantities. Each of these configurations has a set price. For example, the Delta II launch vehicle family has two variants: a Delta 7320 and a Delta 7920, with maximum MLEOs of 2760 kg and 5045 kg, and total launch costs of $40M and $50M, respectively [61]. Table 4.4 shows the set of major launch families available in 2007, along with configurations and launch prices (expressed in FY2004 dollars)

For each family, a linear or polynomial equation was fit through the set of (MLEO, cost) data points in that family, as shown in Figure 4-6. The gap for MLEO between 5100 and 8600 kg exists because no launch vehicles are designed to launch single payloads of this size into LEO. To account for this deficiency, a linear interpolation is computed between the two points that bound the gap. There is also a gap for telescopes up to an MLEO of 1220 kg, which is resolved by setting a constant launch price of $20M for this range. The final model consists of a set of lowest launch costs for telescopes with MLEO up to 20,000 kg, as shown in Figure 4-7. In order to convert the cost into FY2000 dollars, the launch cost data discussed above is deflated by a factor of 0.908 [62].

Note there is an implicit assumption in this model that the telescope is compatible with the launch vehicle fairing. The model assumes that if a telescope has a specified MLEO, the design engineers will ensure that the telescope fits into the fairing. A more complete model would use both the dimensions and mass of the telescope to determine the appropriate launch vehicle.

Table 4.4: Launch vehicle family data

Vehicle	Max MLEO	Cost (FY2004)
Delta 7320	2867	$40M
Delta 7920	5139	$50M
Delta IV M+	8600	$133M
Delta IV M+ (4.2)	11700	$138M
Atlas V 402	12500	$138M
Atlas V 532	17250	$192M
Atlas V 552	20050	$252M
Ariane 5G	16000	$180M
Athena I	820	$40M
Athena II	2065	$45M
ARPA Taurus	1220	$20M
Taurus	1300	$24M
Taurus XL	1500	$28M
Taurus XLS	1900	$32M
Pegasus XL	443	$20M
Minotaur	607	$19M
Falcon I	668	$6M

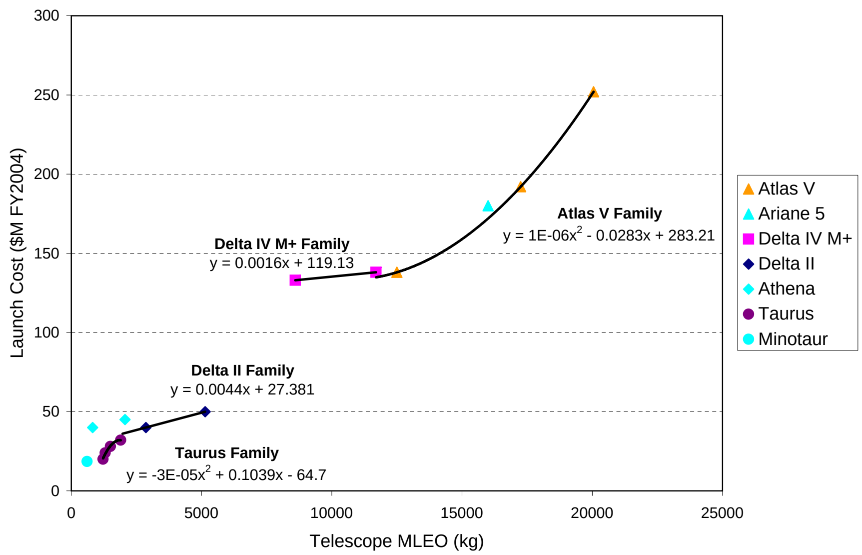


Figure 4-6: Launch vehicle families with trendlines

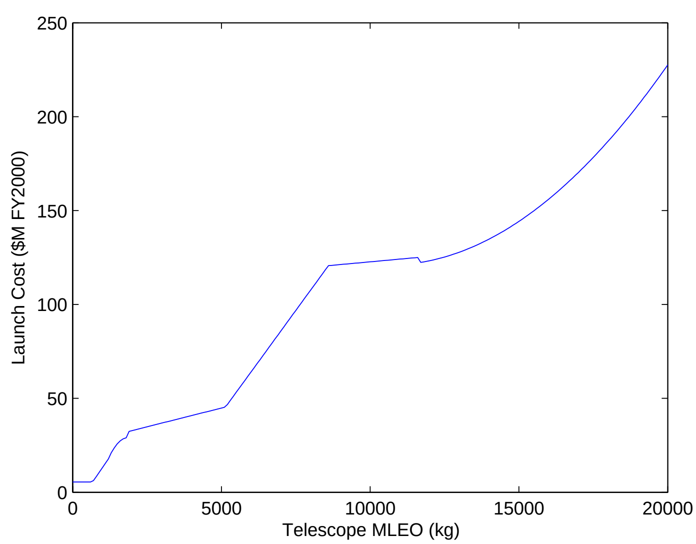


Figure 4-7: Launch Cost as a function of MLEO

4.2.4 Operations Costs

After the telescope is deployed, it must be operated from the ground throughout its life. For the first-order approximation, the annual operations costs of a telescope program is provided by the Mission Operations Cost Model (MOCM) [63], which provides a rough order-of-magnitude estimate of aggregate annual operations costs based on historical data of spacecraft in operation between 1962 and 1990.

As with the AMCM, the MOCM consists of a power law defined by a set of parameters. Unlike the AMCM, however, the input to the cost model is not the telescope MLEO; rather, the *investment cost*, which is the total development and production cost of the telescope. This quantity is output from the Initial Cost model described in Section 4.2.1. By using the output of the initial cost model as input to the MOCM, the annual operations cost can be calculated.

The cost per year ($opsCost$) is given by the power law in Equation 4.3.

$$opsCost = a \times \left(\frac{initCost}{INF87} \right)^{b} \times INF87 \tag{4.3}$$

$initCost$ is the total initial cost of the telescope, a and b are constants whose values come from the *Physics and Astronomy* parameter set. The data set is expressed in FY1987 dollars, so $INF87$ is a multiplier to inflate the costs to FY2000 dollars.

The parameters are summarized in Table 4.5, and the output of this model for a range of MLEO values is shown in Figure 4-8.

Table 4.5: Parameters for the Operations Cost model

Parameter	Value
a	0.047
b	0.878
$INF87$	1.689

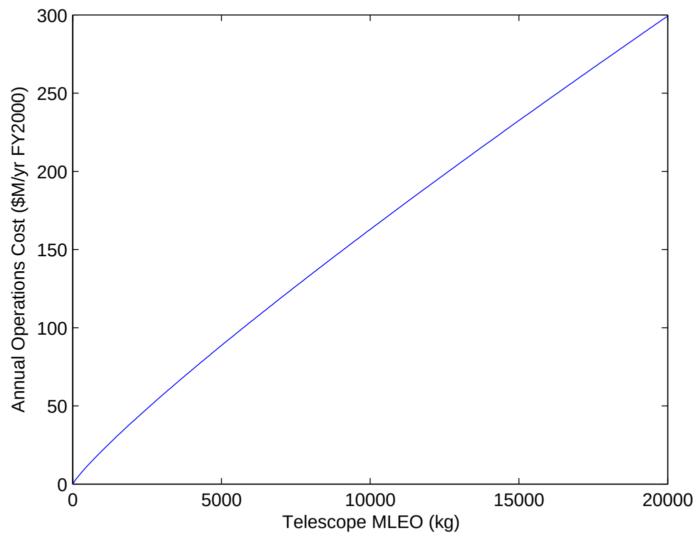


Figure 4-8: Operations Cost as a function of MLEO

4.2.5 Costs for the Replacement Case

In the Replacement Case, when the initial telescope fails, it is replaced with an exact copy. Replacement incurs three primary costs:

- The fabrication of the replacement telescope.
- The construction of the instrument installed on the replacement telescope.
- The launch of the replacement telescope.

The launch cost and the cost of the science instrument can be determined using the Launch Cost and Science Instrument Cost models described above. For the construction cost of the second (replacement) telescope, recall that for the Replacement Case, the second telescope is a copy of the initial telescope. This means that the telescope is constructed and tested without having to repeat all the design and development work that was required for the initial telescope. Since the second telescope is identical to the first telescope (from the same design), the cost of the second telescope is equal to the fraction of the initial cost of the first telescope that was spent on fabrication. This fraction is called the *fabrication fraction*, which is represented in the simulation as the `fab_fraction` parameter. In addition to saving money by reusing the design, fabricating the replacement telescope will also benefit from *learning effects*, which capture the experience gained by building a product, which improves productivity and decreases costs when building subsequent copies [62]. The learning effect is represented in the simulation as the `learn_curve` parameter.

Combining the fabrication fraction and learning effects, the cost of the replacement telescope is expressed in Equation 4.4 [64] as a fraction of the initial cost (*initCost*) of the first telescope

$$\begin{aligned} \mathit{replaceCost} &= \mathit{initCost} \times \texttt{fab\_fraction} \times (2^B - 1) \\ B &= 1 - lg\left(\frac{1}{\texttt{learn\_curve}} - 2\right) \end{aligned} \tag{4.4}$$

4.2.6 Costs for the Servicing Case

In the Servicing case, when the telescope fails, the worn-out components are replaced and the science instruments are upgraded. Servicing incurs four costs:

- The telescope must be designed with serviceability, which incurs additional costs in the initial design phase.
- The construction of the science instrument installed during servicing.
- The purchase of components to replace the ones that have failed, or are likely to fail soon, on the telescope.
- The servicing mission itself.

Incorporating serviceability involves additional up-front design effort to modularize components into replacement units and includes servicing infrastructure such as docking mechanisms, servicing hardpoints, access doors, etc. This is an extremely difficult quantity to estimate since it requires detailed information about the design of a telescope. In the simulation, a simple, multiplicative parameter represents the increase in initial costs to include serviceability in the design. This factor is applied to the initial cost of the telescope to determine the new initial cost of the serviceable telescope, and is represented in the simulation as `serviceable_inc`. For example, if the initial cost must be increased by $X\%$ to enable servicing, then the `serviceable_inc` is equal to $(1 + X)$.

The cost of the new science instruments uses the Science Instrument cost model discussed above in Section 4.2.2. The cost of replacement components is represented in the simulation as `cost_replace`. The cost of the servicing mission itself will be calculated in accordance with the method discussed in Section 3.1, where the servicing cost is represented as the maximum price that program managers are willing to pay in order to perform servicing missions.

4.2.7 Cost Model Summary

The cost models described above were simplified versions of code that would be placed into the simulation framework. This was done to demonstrate how the framework can be implemented. If the framework were used to make decisions on an actual telescope program, much more accurate (and often proprietary) cost models are required. The framework was designed to be modular so that higher fidelity models can be added as needed.

The results of these cost models are used to determine estimates of the total lifecycle costs for telescope programs with either replacement or servicing. When adding up all program costs, no discount rate applied, since NASA budgeting practices for space programs do not incorporate discounted cash flows for multi-year projects. These cost estimates, coupled with the science benefits, will be used when comparing the Replacement and Servicing cases, with the goal of determining the value of serviceability in a telescope design. There is no discount rate applied, since NASA budgeting practices for space programs do not incorporate discounted cash flows for multi-year projects.

4.3 Telescope Model

A telescope consists of many subsystems that work together to support the science-gathering instruments. In this framework, a telescope is represented as a collection of *component sets*, where each component set is a collection of individual *components*. This hierarchy is shown in Figure 4-9. For example, HST is equipped with a set of six rate-sensing units (RSUs). The RSUs contain gyroscopes and electronics that provide orientation information to the attitude determination and control system (ADCS), so that HST can be stabilized and accurately pointed at celestial objects.

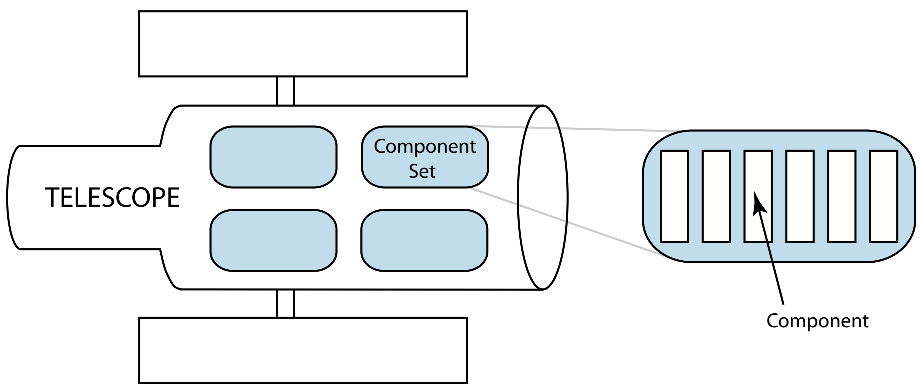


Figure 4-9: Hierarchy of components in the telescope model

A component set is represented by three variables in the simulation:

- *Total* number of components initially installed in the component set.
- *Active* number of components in the total that are operational at any time under nominal conditions.
- *Minimum* number of components that must be functioning for the telescope to be operational.

The distinction between *total* and *active* is needed because each component is not necessarily active at the start of the mission. Spacecraft subsystems often are designed with redundancy, so that the telescope can remain operational in case of single component failures. Returning to the HST example, not all RSUs are needed at all times. Of the six RSUs onboard, at least three are needed for normal operations. When one of the active RSUs fail, a backup RSU is brought online to replace it [25]. As discussed in Subsection 3.3.3, the components each have a state that can be influenced by probabilistic events. In the framework, each component in the component sets can be in one of three states:

- *Active*: the component is currently operational and subject to failures.
- *Standby*: the component is not operational, but can become active later if needed (has not already failed).
- *Failed*: the component was formerly active and suffered a failure event, and cannot become active again.

At the start of the mission, the required number of components in each set is set to active status, and the rest are set on standby. In the HST RSU example, a total of six are installed and four of those are normally active, so at the beginning of the mission, four were set to active status, and the remaining two were on standby. As the mission progresses, active components will fail, and standby units are activated in order to replace these failed components. In this simulation, standby refers to the concept of *cold standby* in reliability engineering. Components in cold standby are not subject to the specified failure distributions and cannot fail until brought to active status. In contrast, components that are on *hot standby* are operating just as if it were active, and are subject to failure.

The transitions between these three states, and the events that cause these transitions, are shown in the state diagram in Figure 4-10. The transition of components from one state to another are described in Section 4.4 (component failures) and Section 4.5 (servicing decision model). Note that failed components can only be returned to a non-failed status (active or standby) via servicing.

The components in each component set are *independent* and *identically distributed*. The failure of one component does not affect the failure of other components (independent), and each component in a component set has the same failure distribution

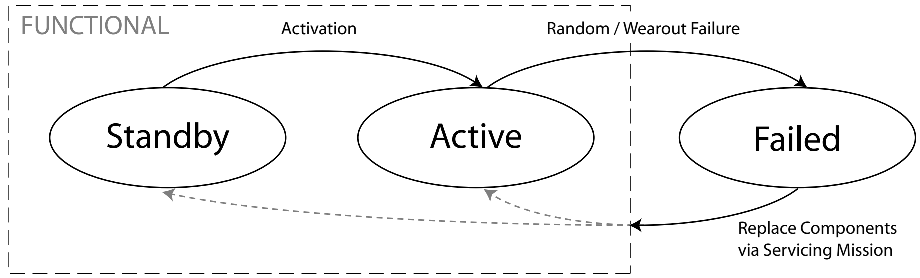


Figure 4-10: Possible states and transitions for each component

(identically distributed). It is reasonable to assume that components from the same manufacturing process are identically distributed. Information about the distribution can come from actual failure rate data from flight hardware or from design specifications from the manufacturer. The independence of components, however, is a major simplifying assumption that appears in the simulation twice. First, each component set is assumed to be independent of other sets; that is, failures in one subsystem do not cascade to others in the telescope. This can be justified based on the HST experience, as most of failures thus far in the HST program are the result of random or wearout failures in individual components. Second, components within a set are assumed to fail independently of others in the same set, which may not be reasonable due to common-cause failures[2] [49]. Common-cause failures were observed in HST, since a manufacturing defect in the RSUs caused them to wear out earlier than expected [65]. The exclusion of common-cause failures is a limitation that should be addressed in future work.

4.4 Modelling Uncertainty: Stochastic Analysis

The stochastic model can handle two types of failure distributions: exponential and non-exponential. For exponentially distributed components, the failure rate $\lambda(t)$, does not depend on the time t that the component has been operational (i.e., $\lambda(t) = \lambda$). For this reason, exponentially distributed components are often called *memoryless*;

[2]A common-cause failure is a failure that affects multiple components at once

that is, they have no memory of prior events and are only subject to random failures. The probability that the component will fail in the next time step is given by Equation 4.5, where Δt is the size of the time step.

$$Pr(\text{failure in next}\Delta t \mid \text{operational at } T) = 1 - exp(-\lambda \cdot \Delta t) \tag{4.5}$$

For non-exponentially distributed components, the stochastic model accepts user-defined functions to compute failure probabilities. For example, the Weibull distribution is commonly used in reliability analysis to model the effects of wearout. Unlike the exponential distribution, the failure rate $\lambda(t)$ is not constant through time [66]. A Weibull distribution depends on two parameters: a scale parameter η (with dimension of time), and a dimensionless shape parameter β [67]. For small time steps Δt, the probability of failure in an operational component from time T to $T+\Delta t$ is expressed in Equation 4.6.

$$\begin{gathered}\lambda(T) = \left(\frac{\beta}{\eta}\right)\left(\frac{T}{\eta}\right)^{\beta-1} \\ Pr(\text{failure in next}\Delta t \mid \text{operational at } T) = 1 - exp(-\lambda(T) \cdot \Delta t)\end{gathered} \tag{4.6}$$

At each time step in the simulation, these models are evaluated to determine the probability that the component has failed. The results are used to update the state of the component sets on the telescope. The updated telescope state is used in the Decision Analysis portion of the simulation discussed below.

4.5 Modelling Management: Decision Analysis

In the two *As-Needed* subcases, decisions on when to service or replace the telescope are made during the program (on-the-fly) rather than during the design phase. To model this on-the-fly decision making, a decision analysis model was developed as a series of decision points through time. The decision of whether or not to replace or service was evaluated at each discrete time step in the simulation until the telescope ceases operations at the end of the program, which is designated the *end of life* (EOL) condition. The decision analysis is executed in three separate stages.

4.5.1 Decision Stage One: Mission Eligibility

The first stage of the decision analysis, shown in Figure 4-11, starts after the stochastic analysis evaluates random events and updates the telescope state. The decision analysis first determines if the telescope is eligible to be serviced or replaced. Eligibility is determined by two criteria. First, the simulation has a specified maximum number of servicing or replacement missions, represented by the `num_serv` and `num_repl` parameters, respectively. If the telescope has already received the maximum number of missions allowed, it is ineligible for further missions. Second, the simulation checks if the telescope has entered the *servicing blackout period*, which is a set time before the end of the nominal mission duration where no servicing or replacement can occur. It is conceivable that mission planners will not want to service the telescope if it is nearing the end of its nominal mission phase, since in their mind the mission is "almost done" and thus servicing is not worthwhile. The length of the blackout period is represented in the simulation as `no_service_period`.

Telescopes that are ineligible for servicing will continue to operate until they fail, at which point the program is terminated (and the EOL condition is signalled). If, on the other hand, the telescope is eligible, then the decision analysis checks if the telescope is operational. The telescope is operational if, for each of the component sets, the number of functioning components is greater than or equal to the minimum required. Conversely, the telescope is not operational if any of the component sets do not meet this minimum requirement. For example, HST requires a minimum of three RSUs to perform science operations. If the simulation showed that only two RSUs are functioning (and the rest have failed), the telescope is inoperative and a servicing or replacement mission must be pre-emptively sent. However, if the telescope is operational and has not yet failed, a servicing or replacement mission may still be needed if the telescope is *near failure*.

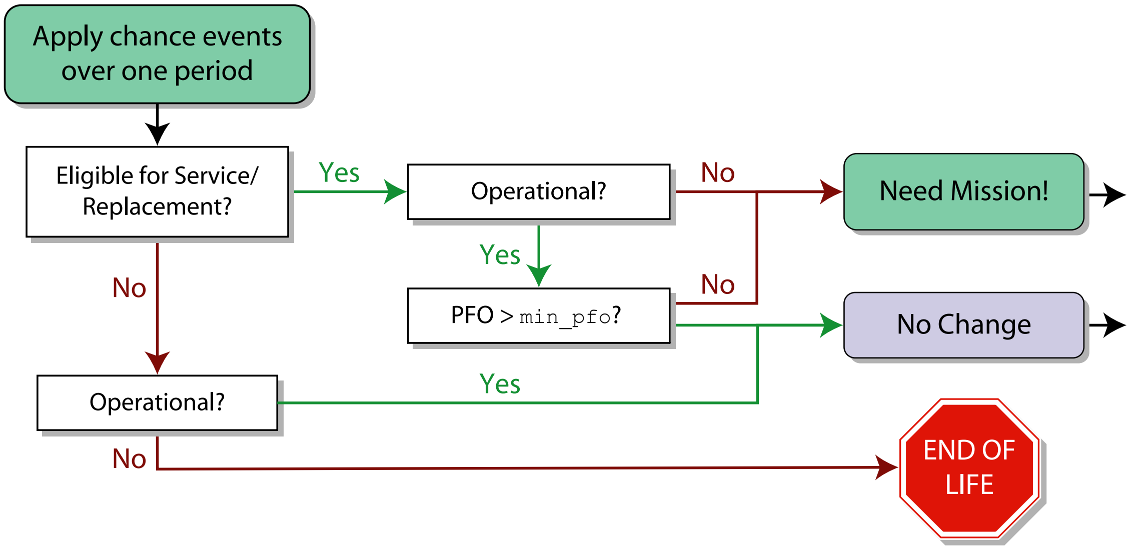


Figure 4-11: Decision Analysis Stage One

Near failure is defined in terms of a quantity called the *probability of future operations* (PFO). The PFO is the probability that, given the telescope is operational at time T, the telescope will be operational at time $T + X$. Thus a telescope is near failure when the PFO drops below a specified threshold. This signifies that there is a significant chance that the telescope will fail in the immediate future, and a servicing or replacement mission must be sent. If the PFO is higher than the threshold, it is unlikely to fail soon, so no mission is required. The threshold used to make decisions about sending missions is represented as a parameter in the simulation called `min_pfo`, which can be changed depending on the program manager's preference.

The parameter X is called the *time horizon* of the reliability analysis and represents how far into the future program managers look to see if the telescope is still operational. This parameter is critical to ensure servicing and replacement missions are effective. If the time horizon is too short, then the decision to service or replace may be made too late: the telescope may already be near failure, and by the time the mission is executed, the telescope has failed. On the other hand, if the time horizon is too long, then a servicing or replacement mission may be sent too early and will service or replace a telescope that is nowhere near failure. This forwardlooking technique was also used in a National Research Council study of future HST servicing missions [68]. The time horizon parameter is represented in the simulation as `future_rel_yrs`.

The probability that the telescope is operational at a future time is the probability that *all* the component sets are operational at that future time. If there are q component sets in a telescope, and the ith component set is represented as S_k, then the probability that the telescope is operational is the product of the probabilities that each component set is operational, as shown in Equation 4.7:

$$Pr(\text{Operational at time } t) = Pr(S_1 \text{ ops}) \times Pr(S_2 \text{ ops}) \times ... \times Pr(S_q \text{ ops}) \quad (4.7)$$

The kth component set contains n_k components, where at least r_k of them must be active for the set to be operational. The probability that the kth component set is operational is the probability that *at least* r_k components in the set are active; that is, there are at least r_k components that have not failed. The probability that this occurs is given by Equation 4.8:

$$Pr(\text{at least } r_k \text{ active}) = \sum_{i=r_k}^{n_k} Pr(\text{exactly } i \text{ components active}) \quad (4.8)$$

The probability that exactly i out of n_k components are active is the sum of probabilities of each combination where i components are active and $n_k - i$ components have failed. There are $\binom{n_k}{i}$ such combinations. For example, for $n_k = 3$ and $i = 2$, the probability that exactly i are operational is shown in Equation 4.9, where C_i is the probability that the ith component is operational.

$$C_1C_2(1 - C_3) + C_1C_3(1 - C_2) + C_2C_3(1 - C_1) \quad (4.9)$$

The above equations are used to calculate the overall PFO, which is used by the decision model to determine if the telescope is near failure and requires servicing or replacement.

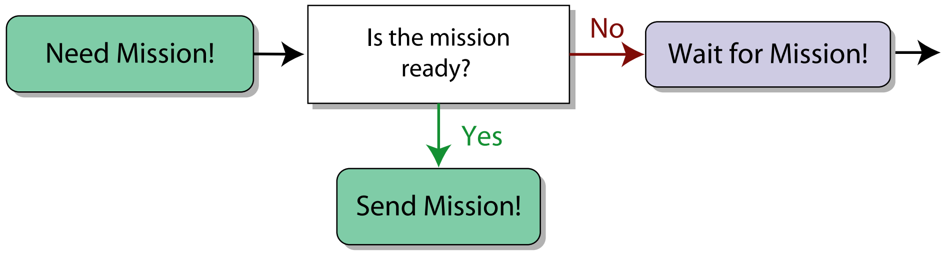


Figure 4-12: Decision Analysis Stage Two

4.5.2 Decision Stage Two: Mission Wait Time

The second stage of the decision analysis, shown in Figure 4-12, is triggered when the simulation has determined that the telescope needs servicing or replacement. The telescope cannot be immediately serviced or replaced because there is a time delay associated with the decision to send a mission - it must wait until the mission is *ready*. In the case of servicing, the servicer spacecraft must be built, tested, and launched. Similarly, in the case of replacement, the telescope itself must be built, which presumably would take longer than building the servicer spacecraft since it will most likely be a more complex system. The latency duration is dependent on the speed at which the servicer spacecraft or telescope could be built. The latencies associated with servicing and replacement missions are represented in the simulation as `serv_latency` and `replace_latency`, respectively. While waiting for the mission to be launched, the telescope operates normally (or remains inoperative if already failed).

4.5.3 Decision Stage Three: Mission Execution

The third stage of the decision analysis, shown in Figure 4-13, examines the events that occur once the servicing or replacement mission is sent. There are three probabilistic events that may occur during a mission, each of which is specified by a parameter in the simulation:

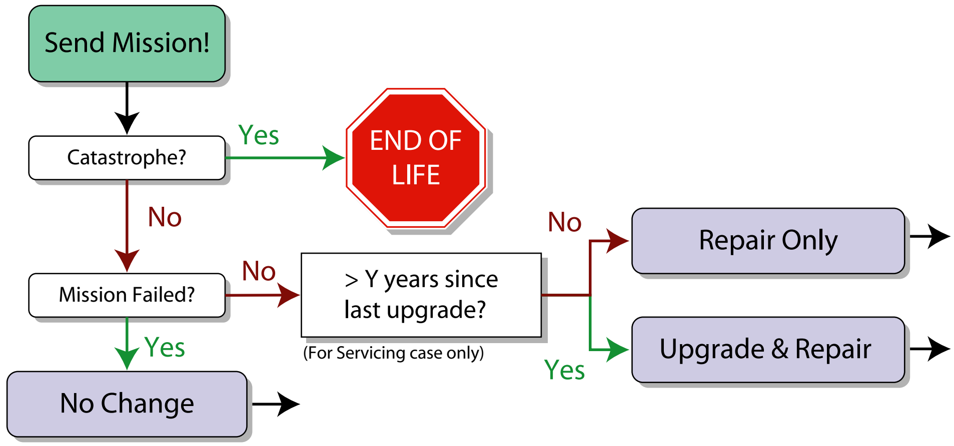


Figure 4-13: Decision Analysis Stage Three

- *Launch Failure*: The replacement telescope fails to be launched, which results in the loss of the replacement. The probability that a launch failure occurs is represented in the simulation as `risk_launch_fail`.
- *Servicing Failure*: The servicing mission fails to upgrade and/or repair the telescope but does not damage the telescope. This event can occur either through a failure during launch or operations. The probability of this event occurring is represented in the simulation as `risk_service_fail`.
- *Catastrophic Failure*: The servicing mission inadvertently disables or destroys the telescope. This will end the telescope program and represents an EOL condition. The probability that a catastrophic failure occurs is represented in the simulation as `risk_cat_fail`.

If none of these events occurs, the mission is a success. In the Replacement case, the second telescope is deployed, without risk to the first telescope[3], and begins producing science data. In the Servicing case, the telescope may be repaired and/or upgraded, and the specific actions that take place during a servicing mission depend on the state of both the instruments and components.

[3]If the initial telescope is still operating, the replacement telescope could be launched to an orbit that does not endanger the initial one.

If an instrument has been upgraded recently, it is unlikely that mission planners would want to replace the expensive science instrument so early in its life for two reasons. First, telescope instruments are expensive, so scientists and program managers would like the instrument to be used for a while before upgrading it with a new one. Second, if the instrument is upgraded rapidly, the discovery efficiency of the replacement will not be significantly higher than that of the original. To account for this issue, the simulation contains a *minimum upgrade time* parameter, which is the minimum amount of time that the instrument must operate before it can be upgraded. If the servicing mission is sent before the minimum upgrade time has elapsed, the instrument will not be upgraded. This parameter is represented in the simulation as `min_upgrade_time`.

The components that are replaced via servicing are those that failed (corrective action) and those that are *close to failure* (preventive action). A component is deemed close to failure if it has been active for a certain amount of time; components that were never active or were active for a short time are deemed healthy enough to remain on the spacecraft. Preventive maintenance is attractive since the marginal cost of replacing additional components while performing servicing is small; however, replacing all components during each servicing mission is both impractical and cost prohibitive. The minimum time that a component must be active in order to be replaced during servicing is represented in the simulation as a parameter called `min_replace_time`.

Once the telescope is replaced or serviced, the telescope state is updated and it returns to the first stage of the decision analysis and the time is incremented. This process continues until an EOL condition is detected, at which time the program ends and the simulation stops.

4.6 Determining Value: Program Comparison

One run of the simulation contains the record of events, both of degradation (failures) and renewal (servicing or replacement), during the telescope program from deployment to end-of-life. For each run, there are two primary metrics that are calculated:

- *Cumulative Science Output*: The total amount of science return that is generated over the life of the telescope program. This is the **net benefit** of the program.

- *Total Lifecycle Cost*: The total amount of money spent on the program from all cost sources (except for the cost of servicing missions in the Servicing case). This represents the **net cost** of the program.

In addition to these general metrics, there are secondary metrics that provide insight into other aspects of the telescope program. Each secondary metric is applicable to one or both of the Replacement or Servicing cases, as noted below.

- *Program Lifetime*: The duration between launch and end-of-life of the telescope program. This is applicable in both Cases.

- *Time Offline*: The total duration during which the telescope operations are suspended while waiting for a servicing or replacement mission. During this time, the telescope is not generating any science data. This is applicable in both Cases.

- *Mission Overlap*: The amount of time that two telescopes are operating simultaneously. This is applicable in the Replacement Case only.

- *Initial Telescope Failure Time*: The time at which the first telescope fails. This is applicable in the Replacement Case only.

- *Initial Servicing Mission Time*: The time at which the first servicing mission is sent to the telescope. This is applicable in the Servicing Case only.

The value of serviceability can be determined using the simulation framework by performing the following steps.

Pre-Simulation: Program Definition

As described throughout this chapter, before the simulation can be executed, the models in the simulation must be initialized with data on the telescope program.

This requires that the telescope itself must be defined (in terms of components and component sets), and the various parameters described in the previous sections must be set. As described in Subsection 3.3.3, the simulation will be executed many times in a Monte-Carlo style method so that the dependence on the particular set of random events is removed.

Step 1: Replacement Case Analysis

The first step is to execute the simulation for the Replacement case, to determine the probability distributions of the primary and secondary metrics for both Replacement subcases. As discussed in Section 3.2, this generates the baseline budget and science return to which the Servicing case is compared.

Step 2: Servicing Case Analysis

Similar to the Replacement case, the second step is to execute the simulation for the Servicing case to calculate the primary and secondary metrics for both Servicing subcases. Remember that the lifecycle cost of the telescope program in the Servicing case does not include the cost of the servicing mission: this is accounted for in Step 3 of the comparison (see below).

Step 3: Program Comparison

The third step will determine the value of servicing by comparing the costs and benefits separately. Comparing the benefits of the two cases will determine the amount of science return in the Servicing case as compared to the baseline Replacement case. This can be expressed as an increase factor. For example, if the Servicing case generates twice as much science return as the baseline Replacement case, then the science return increase is 2.0. Note that if the factor is less than 1.0, then the Servicing case has produced less science return than the Replacement case.

Comparing the costs of the two cases will determine the maximum price of each servicing mission as discussed in Section 3.1. The lifecycle cost of the Replacement case represents the baseline budget for the program. Since the lifecycle cost computed for the Servicing case does not include the cost of servicing, the gap between the baseline budget and the Servicing case represents the *remaining budget* that is available to pay for servicing missions. The remaining budget is the maximum amount that can be spent on servicing missions without exceeding the baseline budget.

This process can be repeated for multiple numbers of servicing missions. This will to provide decision-makers information about how much science return can be gained by performing one, two, three, etc. servicing missions, and the maximum price for each mission. This can inform both design decisions (should the telescope design include serviceability?) and programmatic decisions (should the budget be increased in order to pay for more servicing missions and get more science?) .

Post-Simulation: Sensitivity Analysis

The results of the simulation are dependent on the particular set of parameters that were put into the simulation. Sensitivity analysis involves changing these parameters to determine their effect on the primary and secondary metrics.

4.7 Framework Summary

This chapter described the construction of a framework to analyze the value of serviceability in a telescope design. The framework is implemented by a set of models that were developed in this chapter as well. Some of these models are generally applicable, while others were developed for demonstration purposes only and should be replaced with higher-fidelity models when the framework is used in a real telescope project. These models are based on a set of parameters, which are summarized in Table 4.6.

The next chapter contains a case study which demonstrates how the framework can be used. The parameters will be set with representative data, and the simulation will be executed to determine the value of serviceability for an example telescope program.

Table 4.6: Input Parameters for the Simulation

Program Parameters	
drymass	Telescope Size
max_life	Nominal mission duration
num_serv	Maximum number of servicing missions (Servicing case only)
num_repl	Maximum number of replacement missions (Replacement case only)
	Type of program sub-case (mission at fixed times or on-failure)
Simulation Parameters	
dt	Simulation time step size
Decision Rule Parameters	
min_pfo	Minimum PFO below which a mission is sent
future_rel_yrs	Number of years into future to look forward in PFO calculations
serv_latency	Latency in servicing mission execution
replace_latency	Latency in replacement mission execution
upgrade_latency	Instrument construction lead time
no_service_period	Time before nominal mission end when no servicing is performed
min_upgrade_time	Minimum operating time before an instrument can be upgraded
min_replace_time	Minimum operating time before components can be pre-emptively replaced
Risk Probabilities	
risk_launch_fail	Risk of launch failure (Replacement case only)
risk_service_fail	Risk of servicing mission failure (Servicing case only)
risk_cat_fail	Risk of catastrophic failure (Servicing case only)
Cost Parameters	
learn_curve	Learning effect percentage
fab_fraction	Fraction of total initial cost spent on fabrication
serviceable_inc	Fractional cost increase to incorporate serviceability
cost_replace	Cost of each replacement component
Telescope Model Parameters (for each component set)	
num_comp	Total number of components per set
num_ops	Active number of components per set
num_needed	Minimum number of components required per set
	Hazard rate and reliability parameters

Chapter 5

Framework in Action: An HST-Based Case Study

Now that the simulation framework has been established in Chapter 4, it is ready to be used to analyze the value of incorporating serviceability in a telescope design. This chapter describes a case study of a telescope program, which is a sample execution of the simulation that uses a simplified telescope design and representative numbers for telescope parameters. The telescope and parameter information was derived from HST wherever possible.

The objectives of this case study are twofold. First, it will demonstrate how the simulation framework can be used to determine the value of servicing. Second, it will highlight some general insights about servicing within telescope programs, and how the framework can be used to generate further insights. The case study is executed in four stages, which are the same stages that would be performed if the simulation were used during a real design project. The following sections discuss each of these stages in turn.

1. Define the telescope program to be modelled.
2. Define all simulation parameters.
3. Execute the simulation and analyze the results.
4. Perform a sensitivity analysis on the simulation parameters.

5.1 Defining the System: Telescope Model

As described in Section 4.3, the telescope is modelled as a group of component sets, where each component set is a group of independent, identically distributed components. The component sets must be explicitly specified in the simulation before the analysis can be performed. For a real project, the telescope model would represent all component sets in the design; however, this can render the simulation computationally expensive. For this case study, a simplified model of the telescope will be used, which contains a limited set of components.

If only selected components are included, then those should be the ones that contribute the most to telescope failure. This requires information about the failure rates of telescope components. In the case of HST, this information is provided by the Aerospace Corporation [69] in a report on the reliability of HST midway through its life. The report identified the ten components installed on HST that are *reliability drivers*, which are the components that are most likely to cause HST to fail in the near future. The reliability drivers are described in Table 5.1, where $R(6\text{ yrs})$ is the probability that the component set will be operational in 6 years.

This case study will use four of the top five HST reliability driver components[1]. For each of these components, the report specifies a failure distribution and associated parameters, which is based on design specifications, testing, and on-orbit flight experience. The failure data used in the simulation are summarized in Table 5.2, and they are used in the manner described by Section 4.4.

The payload of the telescope, which in general consists of multiple scientific instruments, is modelled as a single infrared camera in this case study. An infrared camera is selected to ensure that the discovery efficiency metric and the instrument technology data described in Section 4.1 are valid. The camera is assumed to not be subject to random failures. This is a simplifying assumption, but it is reasonable because HST has shown that bus components failures have been more critical than instrument failures. Future work can modify this assumption.

[1]Of the top five components, the Data Management Unit (DMU) is excluded because it is an aggregate of many other components, and the report does not quote a failure rate for the DMU

As well, the overall telescope program modelled in the case study will match the properties of the HST program. There are two parameters that define the telescope program: the nominal mission duration and the size of the telescope. The nominal mission duration is set to 15 years, as was the case for HST. A longer mission duration was selected so that either replacement or servicing must be performed for the telescope to remain operational: it is extremely unlikely that the telescope will last 15 years without intervention. The mass of the telescope is set to 11,000 kg, which is approximately equal to the HST mass of 11,100 kg [1].

Table 5.1: Cost sources for a telescope program (from Wong [69])

Rank	Component	R(6 yrs)
1	Fine Guidance Sensors	0.7025
2	Data Management Unit	0.7377
3	Rate Gyro	0.7998
4	Reaction Wheel Assembly	0.8519
5	Solid State Recorder	0.8642
6	Power Distribution Unit	0.8950
7	Electrical Power/Thermal Control Electronics	0.9118
8	Power Control Unit	0.9208
9	Science Instrument computer	0.9377
10	Solar Array Electronics Control	0.9511

Table 5.2: Failure rate data used in the case study (from Wong [69])

Component	Total	Active	Min	Type	Parameter
Fine Guidance Sensors	3	3	2	Exponential	$\lambda = 8.5335 \times 10^{-6}$
Reaction Wheel Assembly	4	4	3	Exponential	$\lambda = 3.7318 \times 10^{-6}$
Solid State Recorder	2	2	1	Exponential	$\lambda = 8.7468 \times 10^{-6}$
Rate Sensing Unit	6	3	3	Weibull	$\eta = 5.894, \beta = 4.82$

5.2 Setting the Dials: Parameter Specification

The simulation contains many parameters, as described in Chapter 4. Before the simulation can be executed, all of these parameters must be fully specified. In this case study, the parameters will be set using educated guesses, data from the HST program, or historical aerospace industry data. This section describes how each parameter was set, and the parameter settings are summarized in Table 5.3.

5.2.1 Technology Advancement Model

Section 4.1.4 developed an exponential model to represent the discovery efficiency (DE) of instrument technology through time, reproduced here as Equation 5.1:

$$DE = 10e^{pt} \tag{5.1}$$

Setting the rate parameter p requires data on future technology trends. This case study will use a projected trend of infrared CCD technology from [57] as a surrogate for discovery efficiency. This trend represents the progression of detector sensitivity, measured in the number of pixels per chip, as shown in Figure 5-1. Note the logarithmic scale on the y-axis, so the trend is a power law in the form of Equation 5.1 with parameter $p = 0.3218$. The resulting power law of instrument technology that is used in this case study is shown in Figure 5-2.

5.2.2 Decision Model Parameters

PFO Threshold (`min_pfo`)

This parameter sets the minimum threshold of probability of future operations (PFO). A servicing or replacement mission is sent when the telescope PFO drops below this threshold. When NASA planned for HST servicing missions, engineers used a threshold future reliability level of 50% to determine the interval between servicing missions [68]. For this case study, the decision threshold is set to 50% to match the NASA methodology.

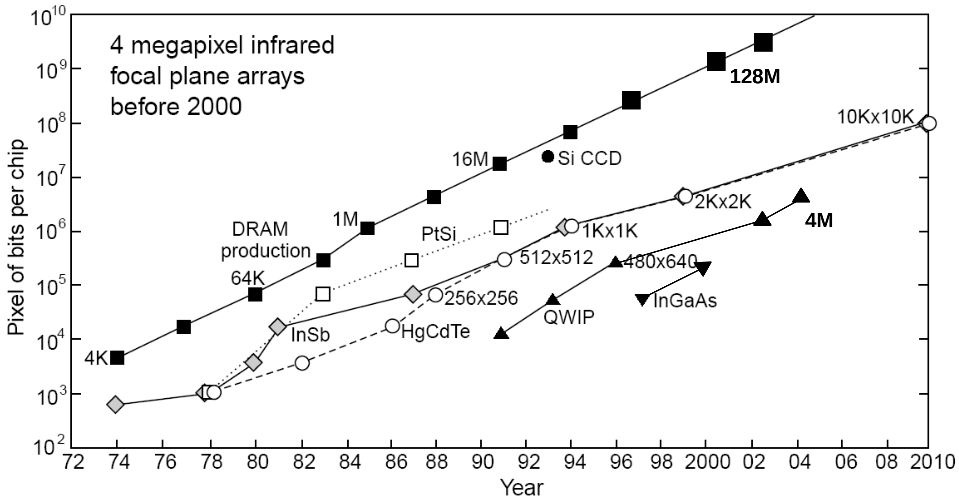


Figure 5-1: Future projections of detector technology (from Xin [57])

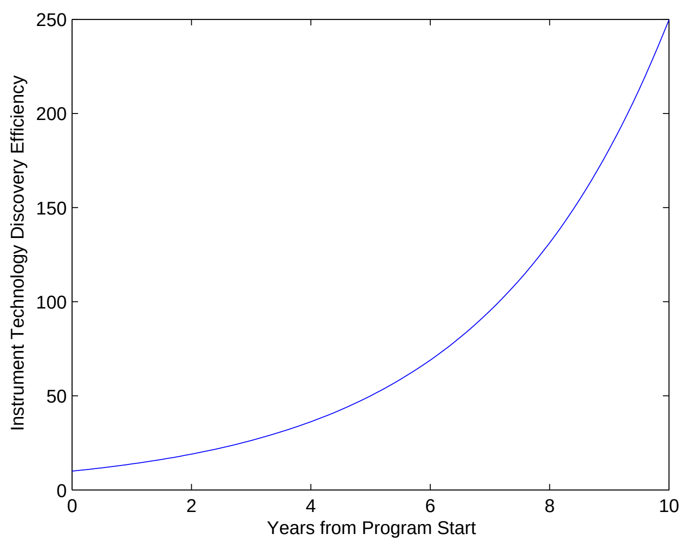


Figure 5-2: Discovery efficiency of state-of-the-art technology

Servicing / Replacement Latency (`serv_latency, repl_latency`)

This parameter specifies the delay between when the decision is made to perform a servicing or replacement mission and when the mission is executed. The parameter depends on the speed at which the servicer spacecraft or replacement telescope could be built. For this case study, the servicing and replacement latencies are set to 1 years and 2 years, respectively, which are broad estimates of turnaround time for spacecraft construction (without development time).

Instrument Construction Lead Time (`upgrade_latency`)

This parameter specifies the amount of time that is required to construct a new instrument that will be installed during a servicing mission. This instrument must be built in advance of the launch, and the duration of this time delay is estimated in this case study to be 1 year.

Time Horizon of Reliability Analysis (`future_rel_yrs`)

This parameter sets how far into the future the reliability of the telescope is calculated. In this case study, the future reliability threshold is set to be comparable to the latencies for servicing and replacement so that the decision lead time roughly matches the lag in execution. A small margin is added to provide additional assurance that the decision will be made well in advance. The time horizon is 1.5 years for the Servicing case and 2.5 years for the Replacement case.

Servicing Blackout Period (`no_service_period`)

This parameter sets the length of time before the end of the nominal mission duration where no servicing will occur. This case study assumes that the telescope will be serviced as needed throughout the entire nominal mission duration, so the servicing blackout period is zero years.

Minimum Upgrade Time (`min_upgrade_time`)

This parameter sets the minimum time that an instrument must operate before it can be upgraded by a servicing mission. In this case study, instruments must operate for at least two years before an upgrade to ensure that it is used sufficiently before being replaced.

Component Replacement Time (`min_replace_time`)

This parameter controls the minimum time components must operate before they are replaced through preventive maintenance; that is, how close they are to failure. This simulation uses components from HST, which were designed with a servicing schedule of one servicing mission every 3.5 years. A reasonable strategy is to replace components pre-emptively when they reach the middle of their expected lives. Therefore, this case study sets the component replacement time to 2 years.

5.2.3 Cost Model Parameters

Fabrication Costs (`fab_fraction`)

This parameter specifies the fraction of the telescope initial cost that is spent on fabrication. To estimate this parameter, one can consider several past missions where a spacecraft designed for one mission was copied for another mission. These mission pairs include:

- Cluster I / Cluster II
- Mars Polar Lander / Phoenix
- Mars Express / Venus Express

By nearly copying the spacecraft design, most of the costs incurred in during system definition of the first mission were not repeated in the second. In addition, there were also savings in the construction and testing phase of the second mission compared to the first [70]. Based on this past experience, an estimated fabrication fraction of 60% is used in this case study.

Learning Effects (`learn_curve`)

This parameter sets the amount of benefits from learning incurred during construction of the second telescope in the Replacement case. Estimates for the learning effect in the aerospace industry range from 85% [71] to 95% [64]. This means the construction cost of the replacement telescope would be 85% to 95% of the cost to construct the first telescope. Since the replacement is constructed at least several years after the first telescope, learning effects would be lower than for normal aerospace projects, because knowledge accumulated during the construction of the first telescope would diminish over time. This case study uses 95%, which represents lower learning effect savings.

Cost of Serviceability (`serviceable_inc`)

This parameter specifies the cost increase incurred to include serviceability in a telescope design. The parameter is difficult to estimate because it is highly design-specific. The single source of information available on this topic was a discussion with a NASA engineer who worked on HST development. He recalled an experience where program managers considered removing serviceability from HST in order to cut costs. The possible savings would have been about \$300M, which was 15% of the program budget of \$2B at the time. As an approximation, the cost increase to incorporate serviceability was set to 1.15 for this case study.

Component Costs (`cost_replace`)

The cost of replacing components on HST ranged from \$1.3M for a Rate Sensing Unit to more than \$12M for a Fine Guidance Sensor [59]. For this case study, the component replacement cost is set at a constant \$5M per component.

5.2.4 Risk Parameters

Launch Failure Risk (`risk_launch_fail`)

This parameter is the probability that a launch vehicle failure occurs during the Replacement case. The US launch vehicle failure rates was 6.5% from 1984 to 1992 [72], and the five-year average launch vehicle failure rate has ranged between 5% to 9% [73], although the failure rates were highest in the 1960's during the experimental stages of US spaceflight [74]. This case study uses a launch vehicle failure probability of 7%, in the mid-range of the historical average.

Servicing Failure Risk (`risk_service_fail`)

This parameter is the probability that a servicing mission fails to upgrade and/or repair the telescope, but does not damage the telescope. A servicing mission can fail either during launch or during the operations phase, so the probability must be at least equal to the launch failure risk. This case study uses an estimated servicing failure risk of an even 10%.

Catastrophic Servicing Failure Risk (`risk_cat_fail`)

This parameter is the probability that the servicing spacecraft disables the telescope, which is estimated in this case study to be 2%.

5.2.5 Parameter Specification Summary

The simulation parameters were set using representative data or best estimates, which are summarized in Table 5.3. Now that the parameters are fully specified, the simulation can be executed to determine the benefits and costs of serviceability.

Table 5.3: Parameter Settings for the Case Study

Program Parameters		
drymass	Telescope Size	11,000 kg
max_life	Nominal mission duration	15 yrs
Decision Rule Parameters		
min_pfo	Minimum PFO below which a mission is sent	50%
future_rel_yrs	Number of years into future to look forward in PFO calculations (Servicing case / Replacement case)	1.5 / 2.5 yrs
serv_latency	Latency in servicing mission execution	1 yr
replace_latency	Latency in replacement mission execution	2 yrs
upgrade_latency	Instrument construction lead time	1 yr
no_service_period	Time before nominal mission end when no servicing is performed	0 yrs
min_upgrade_time	Minimum operating time before an instrument can be upgraded	2 yrs
min_replace_time	Minimum operating time before components can be pre-emptively replaced	1 yr
Risk Probabilities		
risk_launch_fail	Risk of launch failure (Replacement case only)	7%
risk_service_fail	Risk of servicing mission failure (Servicing case only)	10%
risk_cat_fail	Risk of catastrophic failure (Servicing case only)	2%
Cost Parameters		
learn_curve	Learning effect percentage	95%
fab_fraction	Fraction of total initial cost spent on fabrication	60%
serviceable_inc	Fractional cost increase to incorporate serviceability	1.15
cost_replace	Cost of each replacement component	$5M

5.3 Rolling the Dice: Program Analysis

The simulation is now ready to analyze the value of serviceability for the telescope specified in Section 5.1. The analysis consists of executing the simulation for each of the four case / subcase combinations, as described in Section 4.6. This section will show the analysis of a Replacement case with one replacement telescope and a Servicing case with two servicing missions. For the *Fixed-Schedule* subcases, the missions will be equally spaced throughout the 15-year mission duration. The cases and subcases are summarized in Table 5.4. The simulation results will be expressed in the primary and secondary metrics as described in Section 4.6 for each of the four case / subcase combinations.

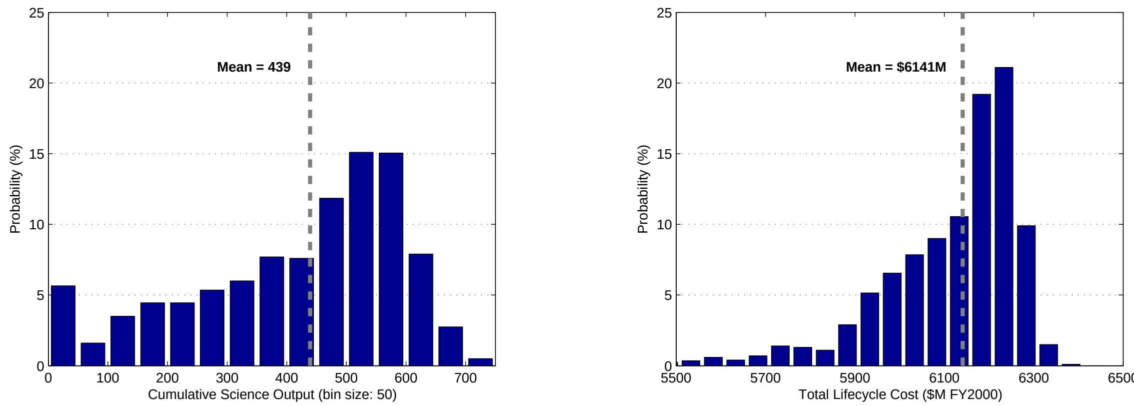
Table 5.4: Cases and subcases to be analyzed

Case	Subcase	Max Missions	Mission Schedule
Replacement	*As-Needed*	1	N/A
Replacement	*Fixed-Schedule*	1	At Year 7.5
Servicing	*As-Needed*	2	N/A
Servicing	*Fixed-Schedule*	2	Every 5 Years

In addition to these programmatic parameters, the simulation itself requires a time step for the decision model; that is, the length of time that each decision and chance node in the decision rule analysis reprsents. This controls how often the telescope program will be evaluated to determine if a mission should be sent. This case study uses a time step of one month (1/12 years).

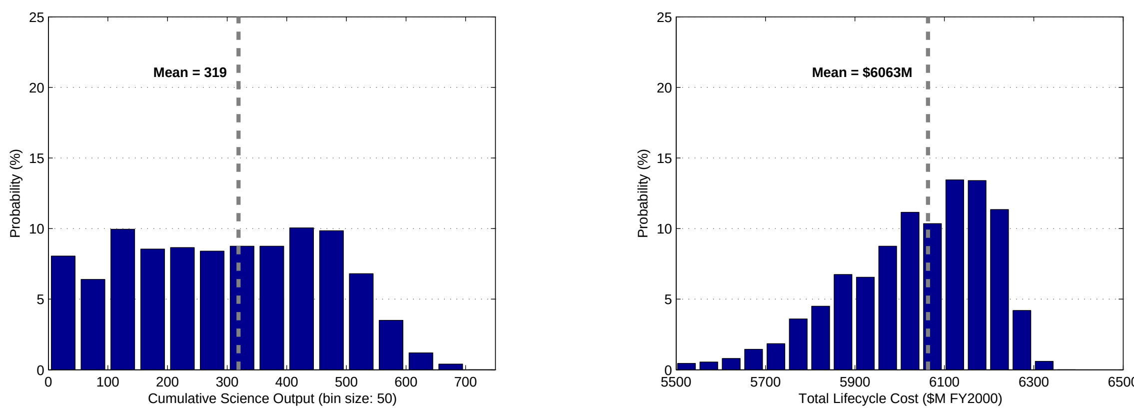
5.3.1 Step 1: Replacement Case Analysis

The primary metrics for the *Fixed-Schedule* replacement subcase are shown in Figure 5-3. The primary metrics for the *As-Needed* replacement subcase are shown in Figure 5-4.



(a) Cumulative Science Output (b) Lifecycle Cost

Figure 5-3: Primary metrics for the *Fixed-Schedule* replacement subcase



(a) Cumulative Science Output (b) Lifecycle Cost

Figure 5-4: Primary metrics for the *As-Needed* replacement subcase

The mean lifecycle cost of each subcase is roughly the same, although the cost distribution for the *Fixed-Schedule* subcase is slightly skewed towards the more expensive side. However, the *Fixed-Schedule* subcase has a much higher science return (mean CSO of 440) than the *As-Needed* subcase (mean CSO of 315). Why is this true? Replacing the telescope at a fixed time of 7.5 years results in better performance because the replacement is launched much later than if the schedule were flexible, so the instruments installed onboard are more advanced. However, this strategy may require the telescope to be offline for a considerable portion of its life. Figure 5-5 shows the distribution of the failure time of the initial telescope.

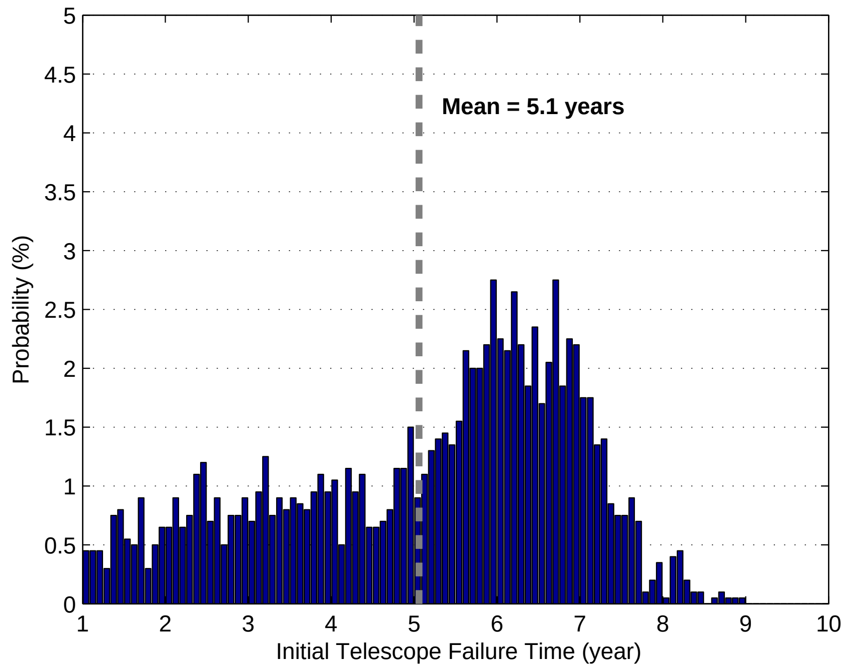
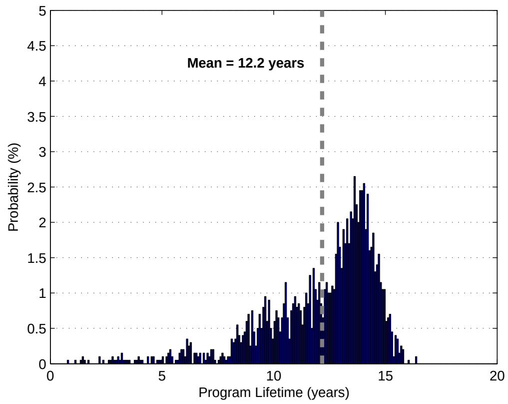
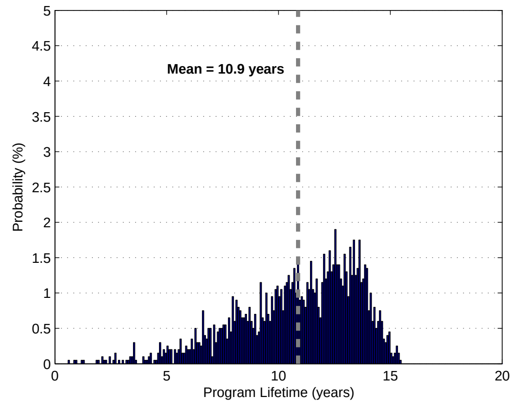


Figure 5-5: Failure time of the initial telescope

Most of the time, the initial telescope fails before 7.5 years, so if the telescope is replaced as needed rather than on a fixed schedule, it will be replaced earlier. Replacing the telescope sooner will result in a less-advanced instrument installed onboard, so the overall science return of the program is reduced. Effectively, this means that it is better to wait to launch a replacement, since by waiting longer, the replacement telescope will receive significantly better instrument technology. As a result, the program (including both telescopes) will have an increased science return over its life.

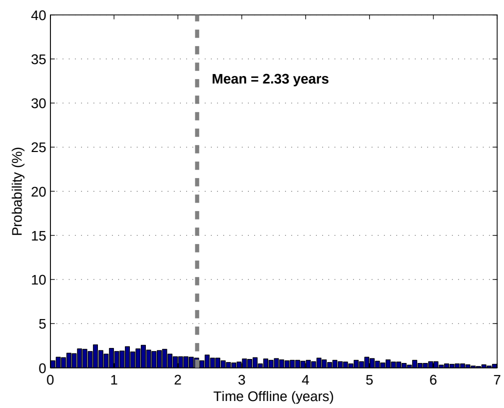


(a) Fixed-Schedule Subcase

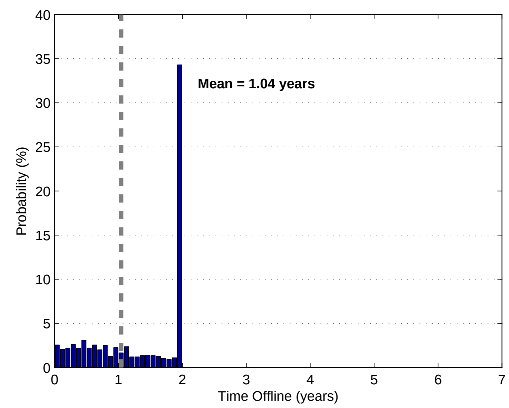


(b) As-Needed Subcase

Figure 5-6: Program Lifetime for both Replacement subcases



(a) Fixed-Schedule Subcase



(b) As-Needed Subcase

Figure 5-7: Offline Time for both Replacement subcases

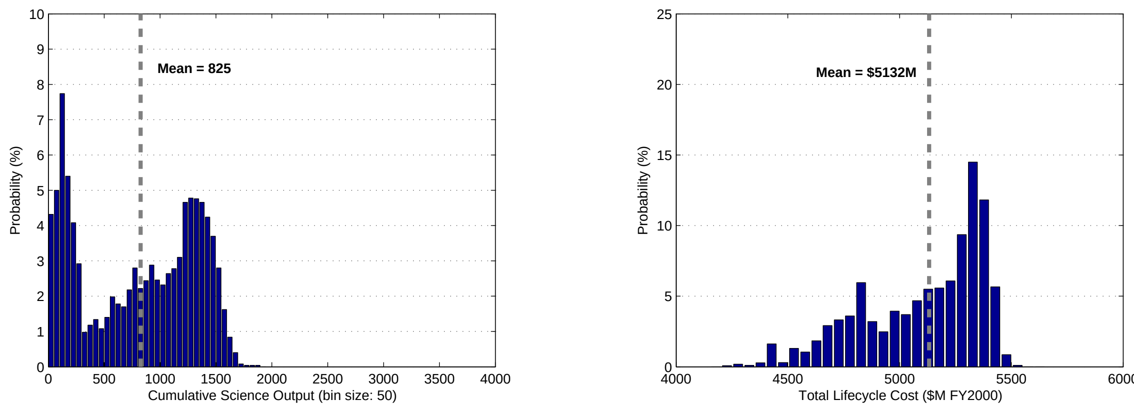
It appears that the *Fixed-Schedule* subcase is preferable under the conditions of this case study; however, an examination of the secondary metrics may illustrate any possible downsides to this strategy. Figure 5-6 shows the program lifetime distributions for both subcases. Again, the *Fixed-Schedule* subcase has a more favourable distribution: the mean is higher, and the distribution is skewed towards a longer program lifetime. However, the fixed schedule strategy has the potential for long periods of time where there is no telescope in operation. Figure 5-7 shows the distributions of offline time for both subcases. In the *As-Needed* subcase, the maximum offline time is 2 years, which corresponds to the telescope construction time. The worst-case scenario is that a telescope fails, and managers immediately make a decision to construct the replacement, which will take 2 years. In contrast, the *Fixed-Schedule* subcase can have offline times that are much longer, because no matter when the first telescope fails, its replacement is sent at a fixed time. For example, if the initial telescope fails after two years, there will be no operational telescope for 5.5 years, until its replacement is launched in year 7.5.

This represents the downside to the *Fixed-Schedule* approach: although the second telescope is launched later and benefits from improved technology, there may be long periods where no science data is being gathered. Indeed, scientists may be willing to sacrifice some science return if it would decrease the likelihood that science operations would be suspended for long durations. The trade between science return and offline time is outside the scope of this simulation, and it must be resolved in discussions between designers and scientists. A major benefit of this simulation framework is that it provides quantitative data to assist this discussion.

This section detailed the results from Step 1 of the simulation. These results will be used as the baseline program against which the Servicing case will be compared to determinine the value of serviceability. The next section discusses the simulation results for the Servicing case.

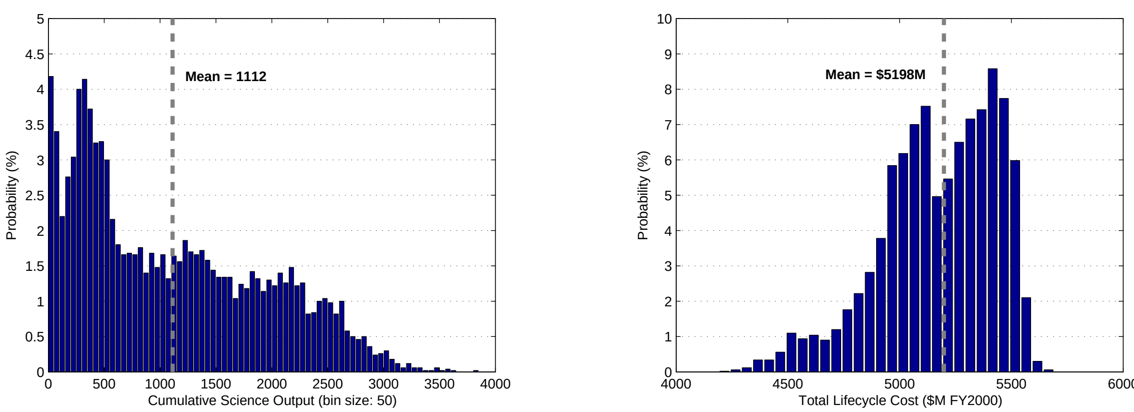
5.3.2 Step 2: Servicing Case Analysis

The primary metrics for the *Fixed-Schedule* servicing subcase are shown in Figure 5-8, and the primary metrics for the *As-Needed* servicing subcase are shown in Figure 5-9.



(a) Cumulative Science Output (b) Lifecycle Cost

Figure 5-8: Primary metrics for the Servicing Case / Fixed-Schedule subcase



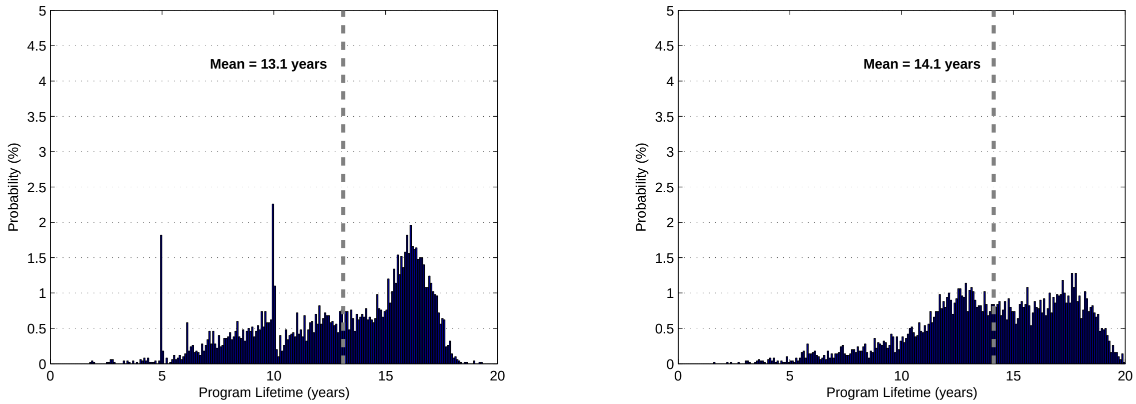
(a) Cumulative Science Output (b) Lifecycle Cost

Figure 5-9: Primary metrics for the Servicing Case / As-Needed subcase

The science return of the telescope program is much higher if the telescope is serviced on failure (in the *As-Needed* subcase) rather than on a fixed schedule of every 5 years. As shown previously in Figure 5-5, there is a strong possibility that the telescope will last past five years without servicing; in fact, it may last up to 8.5 years. If a servicing mission can be held off until later years, then the telescope will be equipped with much more advanced technology than if it were serviced as scheduled 5 years into the program. This is similar to how the *Fixed-Schedule* replacement subcase provided better science return: it was the subcase that delayed the second mission long enough to tap the benefits of more advanced technology.

The secondary metrics for the Servicing case were also computed. Figure 5-10 shows the program lifetime for the two servicing subcases. The *As-Needed* subcase has a higher mean program lifetime than the *Fixed-Schedule* subcase[2]. However, when the telescope is serviced as-needed, there is a higher expected length of time that the telescope will be offline, as shown in Figure 5-11, which is the opposite result than in the Replacement case. The Servicing case analyzed has two missions (once every 5 years), rather than one replacement mission (once every 7.5 years). With the telescope model and components specified earlier, the telescope is likely to fail in about five to six years, so servicing on a fixed schedule every five years seems to prevent telescope from failing and going offline. In contrast, servicing as-needed often means missions are sent once the telescope has failed, so the telescope has a greater chance of being offline longer.

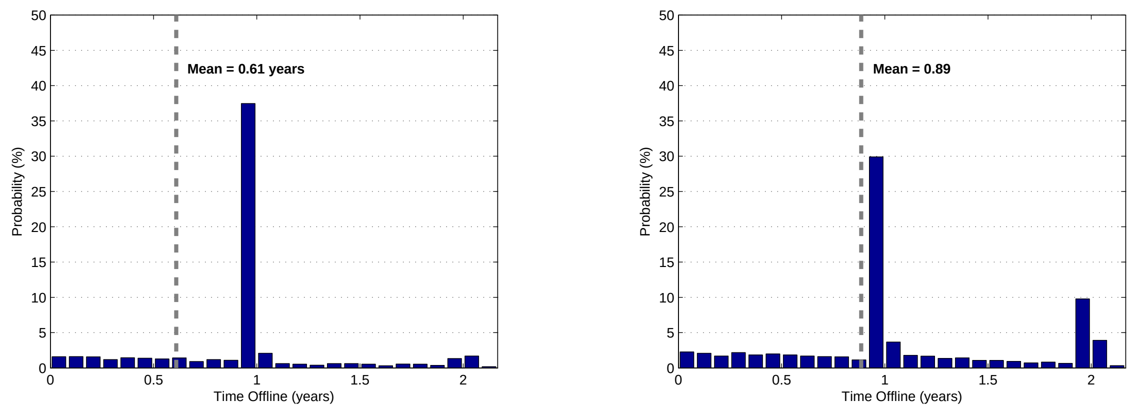
[2]This also accounts for the slightly higher lifecycle cost of the *As-Needed* subcase, since telescopes that operate for longer incur more operations costs



(a) Fixed-Schedule Subcase

(b) As-Needed Subcase

Figure 5-10: End of Life (EOL) for both Servicing subcases



(a) Fixed-Schedule Subcase

(b) As-Needed Subcase

Figure 5-11: Time offline for both Servicing subcases

5.3.3 Step 3: Program Comparison

With the above data for each of the cases and subcases, we can determine the value of servicing as described in Chapter 4 by analyzing the marginal gain in benefits (science return) and the difference in costs under a fixed budget. The Replacement Case is the baseline for comparison, but there are two Replacement subcases from which to choose. If the subcase with better properties is chosen (higher science return, lower cost), the comparison will result in a conservative estimate for the value of servicing. The simulation demonstrated that the two Replacement subcases are relatively equal in lifecycle cost, but the *Fixed-Schedule* subcase has a higher science return. Thus the baseline cost and science return is set to the values for the *Fixed-Schedule* Replacement subcase.

To perform the comparison, we start with the benefits (science return). Table 5.5 shows the science return for each servicing subcase, as a mean plus or minus one standard deviation, compared to the baseline. The results show that science return is, on average, approximately doubled by servicing on a fixed schedule and increased by a factor of 2.5 by servicing as needed.

Table 5.5: Science return of the Servicing subcases compared to baseline

	Baseline	*Fixed-Schedule*	*As-Needed*
Science Return	439 ± 178	826 ± 523	1112 ± 834
Mean Increase	-	1.95	2.53

Next, the program costs between the servicing subcase and baseline must be compared. The total cost of the baseline ($6141M) represents the fixed budget for the program. Table 5.6 shows the budget alongside the lifecycle cost for each Servicing subcase. Remember that these lifecycle costs include all costs **except for the cost of servicing**. The difference between the Servicing lifecycle cost and the baseline budget is the *remaining budget*, which is the amount of the budget that is available to spend on servicing missions, as discussed in Section 4.6. The cost of two servicing missions must be less than or equal to the remaining budget, thus the cost of each servicing mission must be at most half of the remaining budget.

Table 5.6: Total cost of the Servicing subcases compared to baseline

	Baseline	*Fixed-Schedule*	*As-Needed*
Lifecycle Cost (\$M)	6141 ± 153	5132 ± 270	5198 ± 834
Mean Remaining Budget (\$M)	-	1009	943
Max Servicing Cost (\$M)	-	505	472

This maximum servicing cost represents the highest price that the program manager would be willing to pay for each servicing mission. If a contractor states a price that is below or equal to the servicing cost, then servicing can be performed within budget. In this case, the telescope should be designed with the ability to be serviced so it can take advantage of the increased science return that servicing provides. On the other hand, if the quoted mission price is higher than the maximum allowable servicing cost, one of the following two conclusions can be drawn:

1. Performing the two servicing missions cannot be undertaken within budget, so telescope serviceability is not valuable, or

2. The budget should be increased in order to take advantage of servicing.

This analysis can be performed for different maximum numbers of servicing missions to see its effect on both science return and maximum servicing cost. Table 5.7 summarizes the benefits and costs of a telescope program where between one and three servicing missions are sent as needed, with the same baseline for comparison (*Fixed-Schedule* replacement).

The max servicing cost quantities in the last row can be interpreted as before. Consider if a contract states that servicing missions can be completed for $300M. Based on the simulation results, up to two servicing missions can be performed within budget, but three missions cannot because the maximum allowable cost per mission is lower than the quoted price.

Table 5.7: Summary of results for different numbers of *As-Needed* servicing missions

	1 Mission	2 Missions	3 Missions
Science Return	223 ± 128	1112 ± 834	1962 ± 1330
Lifecycle Cost ($M)	4730 ± 203	5198 ± 270	5474 ± 239
Remaining Budget ($M)	1411	943	659
Mean Science Increase	0.51	2.53	3.22
Max Servicing Cost ($M)	1411	472	220

However, performing three servicing missions is very appealing since science return is increased by a factor of 3.22 relative to the baseline. If one wanted to perform three missions, the overall program budget must be increased by $240M (3 × the $80M shortfall per mission). This represents a 4% budget increase over the baseline budget of $6141M.

5.3.4 Program Analysis Summary

The simulation investigated the two telescope program cases with HST-based telescope model and parameter settings in order to determine the value of serviceability. The results showed that the two Servicing subcases were relatively equal in cost, but the *As-Needed* subcase had much higher science return than the *Fixed-Schedule* subcase for two or more servicing missions.

This data can be given to program managers who must decide whether or not to include serviceability in design. With this data, managers would solicit quotes for the price of servicing missions by external contractors. If the quoted price is below the maximum cost, then serviceability can be included within the specified budget. Otherwise, the program managers would know how the budget increase needed to pay for the missions, and the benefit that would be realized if that increase is authorized. The data provided by the simulation provide a more rigorous (and compelling) argument for why serviceability should be incorporated into a design.

5.4 Turning the Knobs: Sensitivity Analysis

The previous section detailed the results of the simulation that used a particular set of values for the parameters described in Section 5.2. Sensitivity analysis can shed some light on the effect of varying these parameters on the results of the simulation, which are referred to in this section as the *nominal* results. This section will detail the results of sensitivity analyses on selected parameters for demonstrative purposes. Results of the sensitivity analysis will be displayed in box plots, the legend for which is shown in Figure 5-12. If the simulation were used during an actual design study, a much larger set of parameters would be examined through sensitivity analysis.

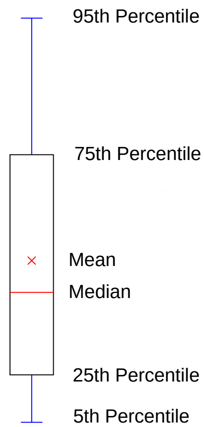


Figure 5-12: Legend for sensitivity analysis boxplots

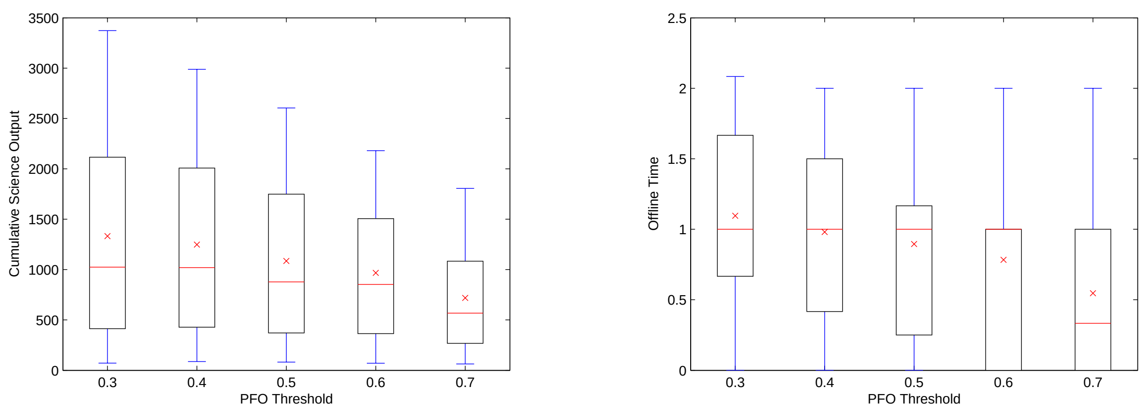
The sensitivity analyses will be performed using the *Fixed-Schedule* replacement subcase as the baseline program, and the *As-Needed* servicing subcase as the nominal servicing program. The nominal servicing program results are summarized in Table 5.8.

Table 5.8: Summary of nominal simulation results

Case/Subcase	# Missions	Science Return	Lifecycle Cost	Max Servicing Cost
As-Needed Servicing	2	1112	\$5198M	\$472M

5.4.1 Sensitivity to PFO Threshold (`min_pfo`)

The PFO threshold controls when servicing or replacement missions are sent. Modifying this parameter changes the reliability threshold below which the program manager will decide to send a mission. Decreasing the parameter causes the manager to wait longer until the telescope is closer to failure, which effectively delays the mission. Conversely, increasing the parameter causes the manager to be more proactive and send missions far in advance of when the telescope is expected to fail. Figure 5-13 shows how the science return and the offline time of the telescope are affected as `min_pfo` ranges from 30% to 70%.



(a) Sensitivity of science return

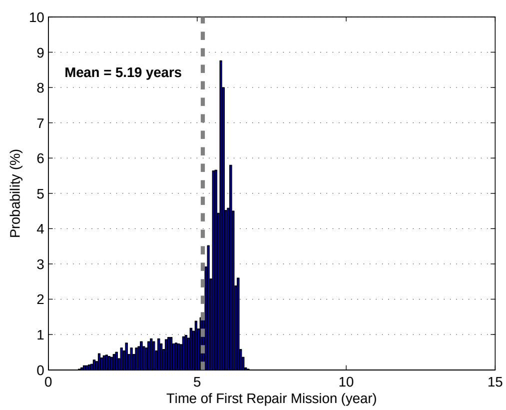
(b) Sensitivity of offline time

Figure 5-13: Sensitivity of results to changes in `min_pfo`

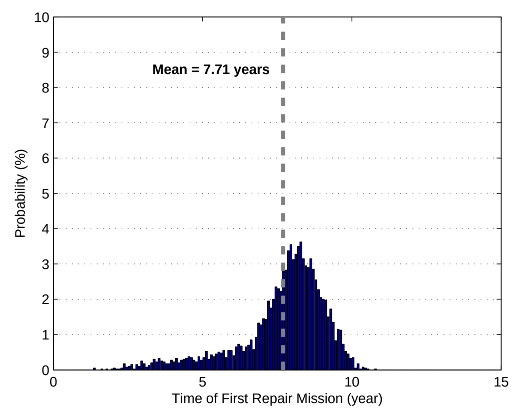
As the parameter decreases, the science return of the telescope increases dramatically (from 719 when `min_pfo` $= 0.7$ to 1332 when `min_pfo` $= 0.3$). Decreasing the parameter pushes the servicing or replacement mission further into the future, so by the time the mission is finally executed, more advanced instrument technology is installed on the telescope. However, by waiting longer to send the mission, it becomes more likely that the telescope will fail before the mission can be executed. Therefore, the expected offline time increases when managers wait longer to make the decision to service or replace.

5.4.2 Sensitivity to Number of Components

One of the main tasks of servicing missions is to replace components that have failed. If the number of backup components onboard is increased, the telescope can operate longer without servicing, so servicing missions can occur later in the life of the telescope. Figure 5-14 shows the distribution of when the first servicing mission is sent for both the nominal telescope design and when the telescope design includes one extra backup component for each component set. Clearly, when additional backup components are installed, the telescope lasts longer.



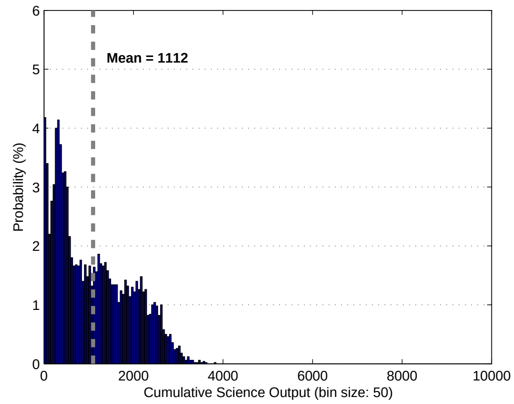
(a) Without extra backups



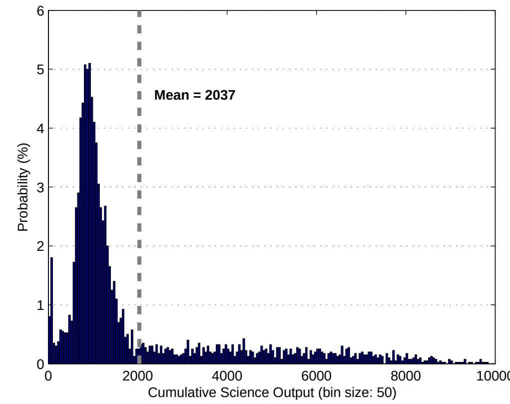
(b) With extra backups

Figure 5-14: *As-Needed* servicing mission times as number of backups is increased

The previous sensitivity analysis demonstrated that delaying the servicing missions increases science return since new instruments that are eventually installed onboard are more advanced. This effect is also apparent when the number of backup components increases. Figure 5-15 shows the distribution of science return for both the nominal and extra-backup designs. The science return is markedly increased when more backups are installed.



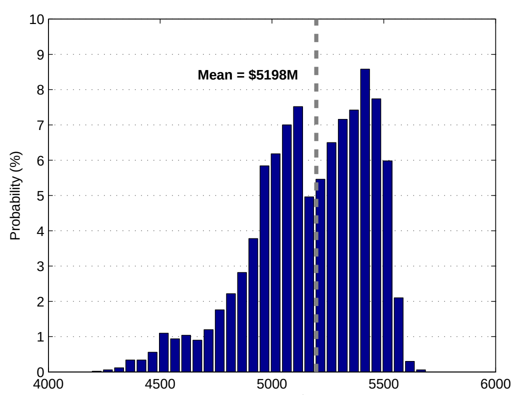
(a) Baseline science return



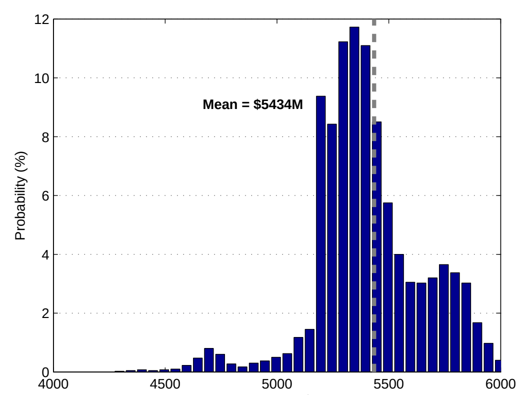
(b) Science return with extra backups

Figure 5-15: Increase in science return when number of backups is increased

Figure 5-16 shows the distribution of lifecycle cost for both the nominal and extra-backup designs. The lifecycle cost of the program marginally increases when extra backups are installed since the telescope life is longer (higher operating expenses) and the cost of additional components. The mean lifecycle cost increased from the baseline of \$5198M to \$5434M when extra backups were included. In practice, it is often difficult to include extra backups due to restrictions on mass, cost, or complexity. However, this sensitivity analysis has demonstrated that it can be worthwhile.



(a) Baseline lifecycle cost

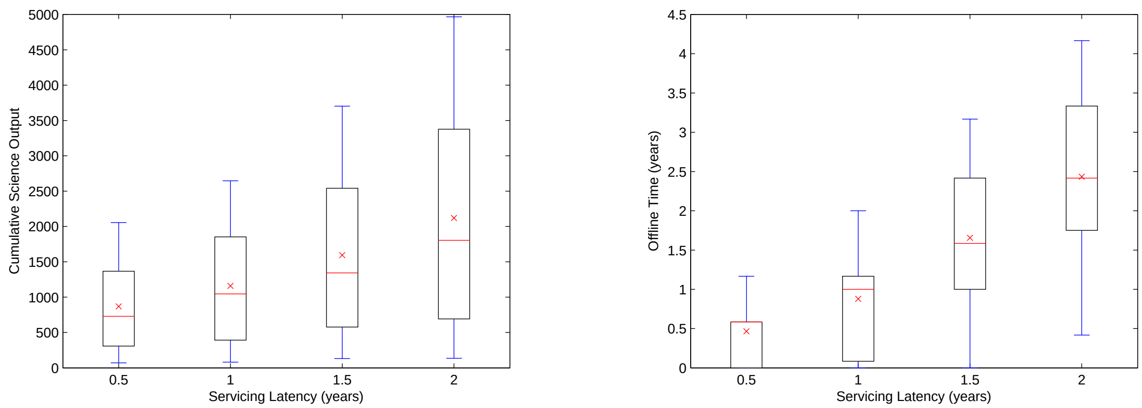


(b) Lifecycle cost with extra backups

Figure 5-16: Increase in lifecycle cost when number of backups is increased

5.4.3 Sensitivity to Servicing Latency (serv_latency)

The servicing latency is the time required to prepare a servicing mission once the decision is made to send one. Modifying this parameter changes the period of time between the decision to service and the execution of the servicing mission. Figure 5-17 shows how the science return and the offline time of the telescope are affected as `serv_latency` ranges from 0.5 years to 2 years.



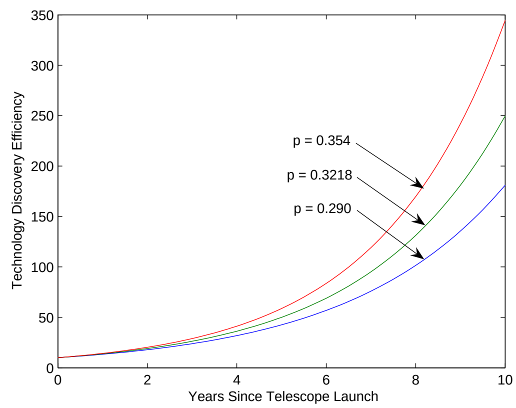
(a) Sensitivity of science return (b) Sensitivity of offline time

Figure 5-17: Sensitivity of results to changes in `serv_latency`

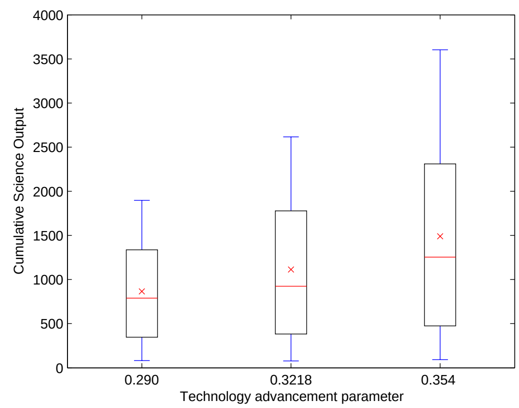
Similar to the `min_pfo` parameter in Section 5.4.1, as the `serv_latency` parameter increases, the science return of the telescope increases as well (from a mean of 870 when `serv_latency` = 0.5 to a mean of 2120 when `serv_latency` = 2.0). With longer servicing latency, the servicing mission requires more time to construct, so by the time it is executed, more advanced technology is installed on the telescope. Note this should be taken as a side-effect of a long construction time: one would not purposefully extend the construction time to gain these benefits. Furthermore, by waiting longer to send the mission, the telescope is more likely to fail before the mission can be executed. Also, a longer latency means that it takes longer to send the mission, so the telescope is more likely to fail in the interim. Thus, the expected offline time also increases.

5.4.4 Sensitivity to Technology Advancement

The discovery efficiency of state-of-the-art instrument technology is controlled by a exponential rate parameter p as described in Section 4.1.4. Increasing this rate parameter will increase the rate of technology advancement. Figure 5-18 shows the effect of increasing and decreasing p by 10% from its nominal value of $p = 0.3218$ in this case study.



(a) Discovery efficiency trends



(b) Sensitivity of science return

Figure 5-18: Sensitivity to changes in technology advancement rate p

Changing the exponential rate parameter had a considerable impact on the instrument discovery efficiency: At year ten, technology developed at a rate of $p = 0.354$ has approximately doubled the discovery efficiency of technology developed at a rate of $p = 0.297$. As technology advances more rapidly, any instrument installed on the telescope will be more advanced as well, which increases the overall science return of the telescope program. As an example, when the parameter was increased by 10%, the science return increased from a mean of 1112 to 1490.

Clearly, serviceability is more valuable when the technology that is installed during servicing advances rapidly. This can both justify including serviceability in a telescope and identify technologies that should be selected as candidates for upgrades via servicing.

5.4.5 Sensitivity to Fabrication Fraction (fab_fraction)

The cost of the replacement telescope is a specified fraction of the cost of the initial telescope. This parameter is critical to the maximum servicing cost because the cost of the Replacement program dictates the baseline budget for any Servicing program. As fab_fraction increases, the replacement telescope becomes more expensive. This increases the baseline budget, which increases the remaining budget available to pay for servicing missions. This in turn increases the maximum price that managers are willing to pay for each servicing mission.

Figure 5-19 shows how the lifecycle cost of the Replacement program changes as fab_fraction varies from 0.4 to 0.8. This represents the baseline budget for different parameter settings. From before, the mean lifecycle cost of the Servicing case (excluding servicing missions) was \$5198M. The difference between this and the budget is the amount remaining that can be paid for servicing.

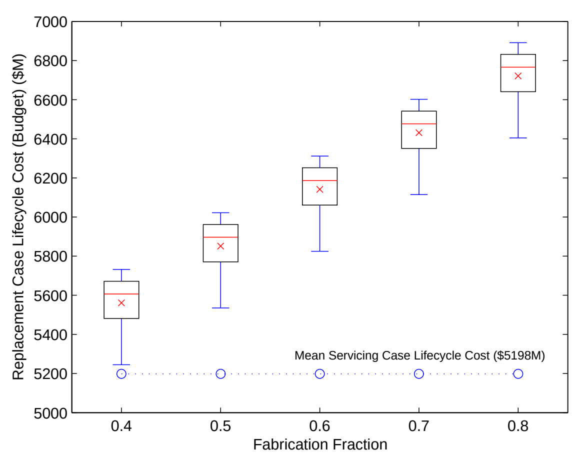


Figure 5-19: Sensitivity of budget to changes in fab_fraction

As the budget increases, the maximum servicing cost increases as well, where the cost is half of the remaining budget (since there are two servicing missions in the nominal case). Figure 5-20 shows the how the maximum servicing cost increases as the fabrication fraction increases.

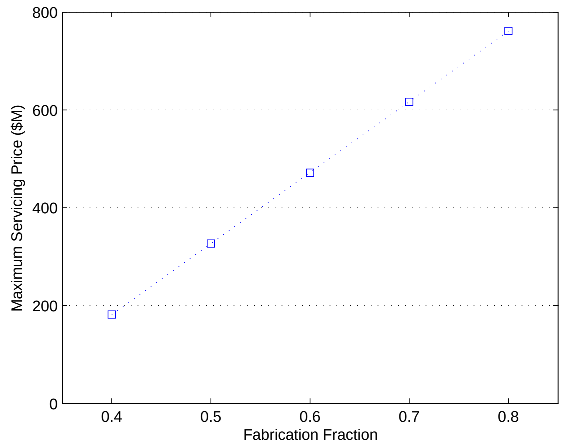


Figure 5-20: Sensitivity of max servicing cost to changes in `fab_fraction`

5.4.6 Sensitivity to Servicing Failure Risk (`serv_risk_fail`)

The servicing failure risk is the probability that the servicing mission will fail to repair and/or upgrade the telescope. Figure 5-21 shows how the science return of the telescope is affected as `serv_risk_fail` ranges from 2% to 20%.

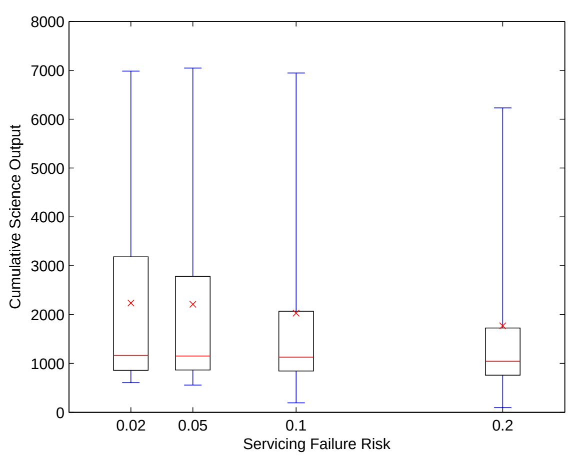
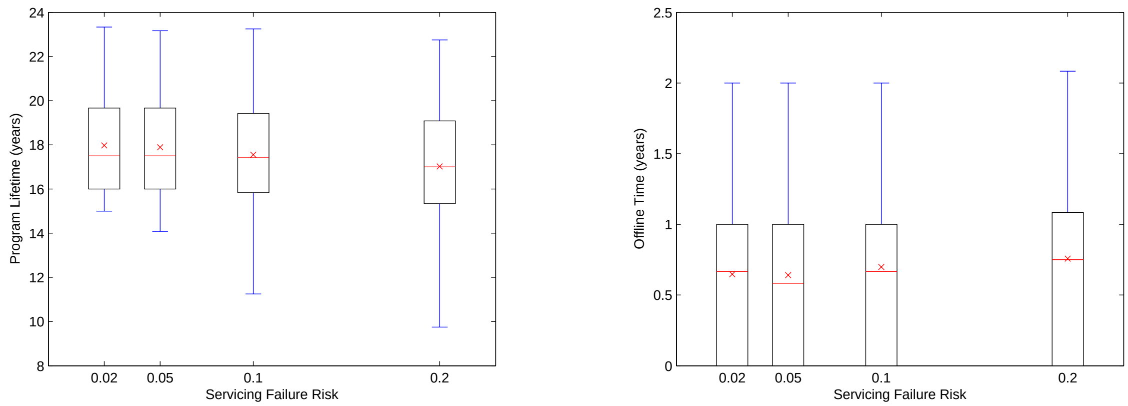


Figure 5-21: Sensitivity of science return to changes in `risk_serv_fail`

The science return is drastically reduced as the servicing failure risk increases because the servicing mission is more likely to fail to upgrade the science instrument.



(a) Sensitivity of program lifetime

(b) Sensitivity of offline time

Figure 5-22: Sensitivity of results to changes in `risk_serv_fail`

If this occurs, the telescope is stuck with the old instrument with a much lower discovery efficiency, so the overall science return of the telescope is lower than if the upgrade was successful.

Figure 5-22 shows the sensitivity of selected secondary metrics to the servicing failure risk. As the servicing failure risk increases, the program lifetime decreases. Without successful servicing missions, components that have failed are not replaced, and eventually so many components fail that the telescope shuts down permanently. The offline time is not significantly affected by the servicing failure risk, because a failure to service will often result in the end of a mission, rather than additional time waiting for a second servicing mission.

5.4.7 Sensitivity to Serviceability Cost (`serviceable_inc`)

The cost of adding serviceability to a telescope is represented as a fractional increase of initial telescope cost. Just as the fabrication fraction parameter discussed above, this parameter dramatically affects the maximum servicing cost. As `serviceable_inc` increases, the serviceable telescope becomes more expensive, and there are less funds available to pay for servicing missions (under a fixed budget), which decreases the maximum price that can be paid.

Figure 5-23 shows how the lifecycle cost of the Servicing program (not including the cost of servicing itself) changes as `serviceable_inc` varies from 1.05 to 1.25. The data points represent the lifecycle cost for each value of `serviceable_inc`. The fixed budget was set at $6141M, which was the lifecycle cost of the *Fixed-Schedule* replacement subcase in the nominal results.

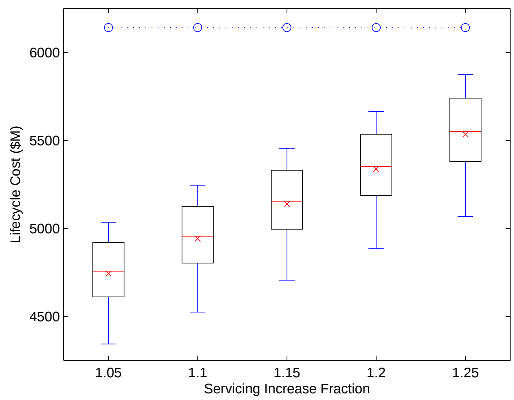


Figure 5-23: Sensitivity of budget to changes in `serviceable_inc`

The difference between the budget and the program lifecycle cost is the remaining budget that can be used to pay for servicing missions. Figure 5-24 shows the how the maximum servicing cost changes as the serviceability cost increases. As before, the maximum servicing cost is half of the remaining budget.

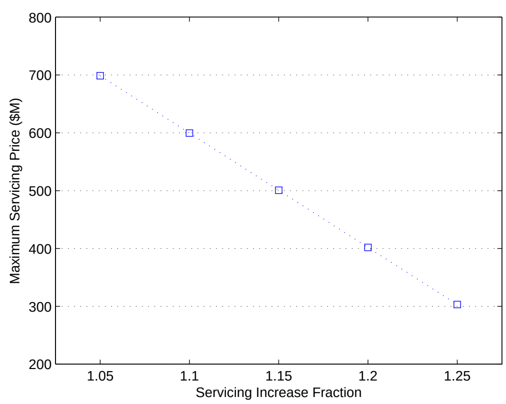


Figure 5-24: Sensitivity of max servicing cost to changes in `serviceable_inc`

5.4.8 Sensitivity Analysis Summary

The sensitivity analysis presented in this section demonstrated the effect of changing parameters from their values in the nominal *As-Needed* Servicing strategy, summarized in Table 5.8. For example, increasing the number of backups or the technology advancement rate increased the science return of the program, which is not surprising. The sensitivity analysis confirmed our intuition and quantified exactly how the parameters affect the results.

In addition, the sensitivity analysis showed effects that were not immediately obvious. For example, increasing the servicing latency also increased the science return, because this pushed servicing missions further into the future and caused more advanced instrument technology to be installed. The analysis also further investigated the effects of parameters on offline time, which was identified as important by the simulation results. Clearly, sensitivity analysis can provide valuable insights into the value of serviceability and give managers a better perspective on the outcome of the simulation.

5.5 Case Study Summary

This chapter has illustrated how the simulation framework can be used to analyze the value of serviceability through the use of a case study. The case study was an execution of the simulation using representative data and a simplified telescope model. If this framework were used in an actual telescope project, the simulation would be populated with more accurate data and a more complete telescope model. As well, a more extensive sensitivity analysis would be conducted.

This case study demonstrated how the value of serviceability can be quantitatively determined using this simulation framework. In Chapter 2, value was defined as benefit at cost. This simulation can compute the benefit (increase in science return) of servicing compared to replacement. Also, the simulation determines the maximum cost of servicing given a fixed budget. A sensitivity analysis can be performed to investigate the effect of simulation parameters on the primary and secondary metrics.

Additionally, this case study provided some important insights. Most significantly, a delay in the servicing or replacement missions, for whatever reason, increases the overall science return of the program, since the instruments installed on the telescope are more advanced. However, this comes at the cost of increasing the time that the telescope is inoperative due to failures. Thus, there is a trade between increasing the telescope science return and minimizing the time offline, which should be addressed by consulting scientists and others who would use the telescope.

Chapter 6

Conclusion

History has shown that although on-orbit servicing is perceived to be valuable, the ability to perform servicing (serviceability) has often been removed from spacecraft designs in the face of budget and scheduling constraints. This is particularly true of space telescope programs. Program managers have often made those decisions without a full, quantitative account of the value of serviceability. This thesis developed a framework that can be used to determine the value of incorporating serviceability into a space telescope. The framework was then implemented using a set of MATLAB functions to demonstrate how it can be used to analyze the value of serviceability. The framework is ready to be used in the future for real telescope design projects, where it can be implemented using company-specific knowledge and numerical codes to provide more accurate and program-specific estimates of value.

6.1 Questions Answered

The goal of the thesis was to develop a framework to analyze the value of serviceability. In order to do so, the framework had to address three specific questions, laid out in Chapter 2. This section will repeat those three questions and discuss how, and to what extent, they were answered throughout this thesis.

Research Question 1

How can the costs and benefits of telescope servicing be compared to determine the value of serviceability?

Answer: A major difficulty in determining the value of serviceability was that the costs and benefits of a space telescope are measured in different units, so they cannot be directly evaluated together in a single metric of value. This issue was resolved by using program comparison. Two separate cases were developed: one where the telescope is replaced on failure (the Replacement case), and one where the telescope is serviced on failure (the Servicing case). The baseline used to compare the two cases is a fixed budget, which was taken as the lifecycle cost of the Replacement case. This allowed the increase in science return from the Replacement case to the Servicing case to be calculated, along with the maximum price that could be paid for servicing missions to achieve this increase in science return.

Research Question 2

How can the incorporation of serviceability in a telescope be justified?

Answer: Telescope serviceability can be justified if the science return of the program is increased compared to the case where a non-serviceable telescope is replaced. This is evaluated under a fixed budget, as discussed above. In order to evaluate this statement, this thesis developed a simulation framework that can be used to determine the expected science return and lifecycle costs (and hence the value) of serviceability. This thesis also presented a case study to demonstrate how the framework can be used in this manner.

Research Question 3

How can the value of flexibility provided by servicing be analyzed?

Answer: Servicing provides the flexibility to make decisions in the future as uncertainty is resolved. This is simulated by using a decision rule model that evaluates when to perform servicing or replacement missions by using a pre-defined set of rules.

The rules are expressed as if/else statements based on a set of parameters that can be tuned to represent the decision criteria of a program manager. As well, a Monte-Carlo analysis is performed to average out the effects of all the uncertain elements in a telescope program. The simulation is executed many times and the results are expressed as probability distributions. These distributions represent the full range of the performance of the telescope program.

The case study has demonstrated that the cost of serviceability is a significant driver of its value. If enabling servicing on a telescope is cost prohibitive, it is unlikely to be valuable, since other alternatives may be more attractive. Other variables that contribute to the value of serviceability include the rate of technology advancement, the amount of backups installed onboard, probabilities of chance events associated with servicing, the thresholds in program decision rules, and the cost of telescope replacement (which is the alternative to servicing).

A major new insight from these analyses is the correlation between increasing science return and increasing offline time when servicing missions are delayed. When missions are pushed further into the future, the science instrument installed on the telescope is more advanced, but this comes at the risk of longer periods where the telescope is offline due to component failures. Scientists may be willing to reduce the expected science return of the telescope in order to protect against this risk. This motivates a trade between increasing the science return of the overall program and ensuring the telescope remains operational (i.e. not offline) throughout its life.

6.2 Future Work

The framework provides a solid foundation for analysis of servicing within telescope programs, but the framework is constructed under several assumptions and modelling decisions. The foundation can be strengthened, and the results more broadly applicable, if further work and research is applied to this framework.

Effect of Component Lifetimes

The failure rate information that was used in the case study was from the components used on the HST. These were designed with the expectation that HST would be serviced every three years. However, if the interval between servicing missions was prolonged, the components would probably have been designed to last longer. This illustrates a trade study that was not incorporated into the framework: the effect of component lifetime on the value of servicing. Realistically, if the servicing schedule is changed, the component lifetime should also be changed accordingly. This was not addressed in the thesis due to a lack of data, as the HST component information was the only data that was found. The effect of component lifetimes is certainly an important avenue to explore.

Trade Between Science Return and Uninterrupted Operations

As discussed in the previous section, there is an important trade between increased science return and less risk of downtime due to failures. To address this trade, one requires the priorities of scientists that would use the telescope data and a method to gauge their tolerance of mission downtime for the benefits of improved science return.

Different Baseline Cases

There were two main strategies (*cases*) that were investigated in this thesis: one where the telescope is replaced (the Replacement case), and one where the telescope is regularly serviced (the Servicing case). The baseline used to determine the value of servicing was the Replacement case, but in doing so it precluded the strategy of designing a single telescope to last for the entire mission duration. This was justified for the case of a long-duration, large telescope, which is impractical to design for such a long life, but there are other cases where neither replacement nor servicing are appropriate, particularly for smaller, shorter duration missions. If other baseline strategies are considered, the framework can be used in more situations.

Generalize to Other Non-Revenue Generating Systems

The framework analyzed the value of serviceability in the specific case of space telescopes. The framework used several techniques to avoid the issue of different units for benefits and costs, to account for decision-making by program managers, and to incorporate the value of flexibility. These techniques are generally applicable to any spacecraft that does not have monetary benefits, which constitutes a large fraction of the current spacecraft fleet in orbit, such as Earth observation satellites, spacecraft that carry science experiments, defence and military systems, and deep space communications arrays.

In each of these cases, the benefits generated by the system are measured differently, but in order to determine the value of serviceability for these programs, issues similar to those for space telescopes must be addressed. It is therefore likely that the framework can be extended for more general non-revenue generating spacecraft.

6.3 Final Thoughts

The Hubble Space Telescope is one of the most productive astronomical instruments in history. It has provided tremendous insights into the universe, expanded the frontiers of scientific knowledge, and captured the imagination of the public unlike any other telescope before. Servicing played an integral part in the success of the HST program, which in turn has renewed interest in the value of on-orbit servicing. It is the author's hope that research into both the *how* and *why* of servicing continues, so that servicing is more widely used in future space programs, and the opportunities that servicing provides can be fully realized.

Appendix A

Simulation Code

The simulation was implemented as a set of MATLAB functions and scripts. The core of the simulation is a set of functions called the *kernel*. The kernel is comprised of two functions that simulate telescope programs through time: `telescopeSim_Replace()` for the Replacement case and `telescopeSim_Service()` for the Servicing case.

Executing these functions will perform one run of the simulation. To perform the Monte-Carlo analysis, these functions must be executed many times, which are done using the `iterateReplace()` and `iterateService()` scripts. Sensitivity analysis was performed using the `sensitivity()` script.

The kernel calls other functions during the simulation to model the performance of the telescope through time. The performance models include models for random events, telescope state updates, and the advancement of technology.

After a simulation has completed, the cost analysis is performed. The lifecycle costs are computed by `costAnalysis_Replace()` for the Replacement case and `costAnalysis_Service()` for the Servicing case. Both of these functions call other models for specific program costs, such as inital costs, instrument costs, etc.

The functions and scripts in the simulation are summarized in Table A.1 and categorized into three groups. The following three sections include the simulation code for these groups.

Table A.1: Summary of MATLAB modules in the simulation

Simulation Kernel	
parameters	Simulation parameter initialization
sensitivity	Sensitivity analysis routine
compInitialize	Script to intialize telescope properties
constInitialize	Script to initialize simulation parameters
iterateReplace	Monte-Carlo sampling for the **replacement** case
telescopeSim_Replace	Decision analysis for **replacement** case
iterateService	Monte-Carlo sampling for the **servicing** case
telescopeSim_Service	Decision analysis for **servicing** case
Performance Models	
computeSetReliability	Computes the future reliability of a component set
createComponent	Creates array to represent the component set
generalFail	General component failure function
hazardGyro	Hazard rate for Weibull-distributed gyroscopes
reliabilityGyro	Reliability calculator for Weibull-distributed gyroscopes
utilityInstrument	Instrument technology discovery efficiency
telescopeRepair	Update telescope state to reflect repair operations
telescopeUpgrade	Update telescope state to reflect upgrade operations
Cost Models	
costAnalysis_Replace	Calculates the lifecycle cost of the Replacement case
costAnalysis_Servicing	Calculates the lifecycle cost of the Servicing case
costComponents	Replacement component cost
costInitial	Initial telescope cost
costLaunch	Launch cost
costOps	Annual operations cost
costScience	Science instrument cost

A.1 Simulation Kernel

```
%PARAMETERS Stores program parameters.

dt = 1/12;
num_serv = 2;
num_repl = 1;

num_iter = 1000;

% select servicing type:
% 'n' = service as needed
% 'y' = service at set times
fixed_schedule_service = 'n';

% select replacement type:
% 'n' = replace when first telescope fails
% 'y' = replace at set times
fixed_schedule_replace = 'n';

% turn off diagnostics
diag_flag = 0;

% turn off sensitivity analysis
% (turned back on by sensitivity.m)
sens_flag = 0;
```

```
%SENSITIVITY Looping script to perform basic sensitivity analysis on one
% or more fixed parameters
%
% Mark Baldesarra
% August 3, 2006

clear
parameters

% turn on sensitivity analysis
sens_flag = 1;

% define parameter to vary for sensitivity analysis
sens_data = 0.4:0.1:0.8;
sens_name = 'fab_fraction';

% set program type to 'replace' or 'service'
program_type = 'replace';

iter_count = 0;
for k=1:length(sens_data)
    iter_count = iter_count + 1;

    % select variable under sensitivity investigation
    sens_param = sens_data(k);

    % specify the program type
    if strcmp(program_type,'replace')
        iterateReplace;
    elseif strcmp(program_type,'service')
        iterateService;
    end

    % extract figures of merit for comparison
    data_science_mean(k) = mean(cum_utility);
    data_science_stdv(k) = std(cum_utility);
    data_science(:,k)    = cum_utility;

    data_offline_mean(k) = mean(offline_time);
    data_offline_stdv(k) = std(offline_time);
    data_offline(:,k)    = offline_time;

    data_EOL_mean(k) = mean(EOL);
    data_EOL_stdv(k) = std(EOL);
    data_EOL(:,k)    = EOL;

    data_cost_mean(k) = mean(total_cost);
    data_cost_stdv(k) = std(total_cost);
    data_cost(:,k)    = total_cost;
end
```

```
%COMPINITIALIZE Script to intialize properties of telescope
%
% Mark Baldesarra
% July 13, 2006

% initialize all constant parameters
constInitialize;

% set operations flags
EOL_flag = 0;                   % flag = 1 when end-of-life occurs
operational_flag = 1;           % flag = 1 when telescope is working
service_now_flag = 0;           % flag = 1 when decision made to service
service_allow_flag = 0;         % flag = 1 when telescope eligible for servicing
sched_flag = 0;                 % flag = 1 when telescope is serviced now

% initialize time counters
time_since_serv = 0;
time_since_upgrade = 0;
serv_wait_time = 0;
offline_time = 0;

% specify properties of telescope component sets
num_needed = [2 3 3 1];     % minimum number of components needed for operations
num_comp   = [3 6 4 2];     % total number of components per set
num_ops    = [3 4 4 2];     % number of components "on" at one time
hazard_ptr = {0.0747, @hazardGyro, 0.0327, 0.0766};
reliability_ptr = {0.0747, @reliabilityGyro, 0.0327, 0.0766};
num_set = length(num_comp);
comp_fails = zeros(1,num_set);

% initialize the output variable arrays
upgrade_times = [];
repair_times = [];
fail_times = [];
repair_set = [0 0 0 0];
utility = [10];     % initialize initial utility to normalized value of 10

% initialize component sets
for i=1:num_set
    comp_set{i} = createComponent(num_comp(i));
end

% start at zero time (at initial operations)
current_period = 0;
current_time = 0;
```

```
%CONSTINITIALIZE Script to intialize properties of telescope
%
% Mark Baldesarra
% July 13, 2006

% set mission lifecycle parameters (time step, max design life)
max_life = 15;
dt = 1/12;
max_period = max_life / dt;
drymass = 11000;

% set program parameters (for fixed case)
fixed_replace_time = 7.5;       % time to perform replacement mission
fixed_serv_times = 3.5;         % time increment between servicing missions

% set decision rule properties
min_pfo = 0.50;                 % minimum PFO below which the telescope is serviced
future_rel_yrs = 1.5;           % reliability analysis time horizon (years)
serv_latency = 1;               % latency in servicing mission (years)
replace_latency = 2;            % latency in replacement mission (years)
upgrade_latency = 1;            % latency in instrument construction (years)

no_service_period = 0;          % servicing blackout period before EOL (years)
min_upgrade_time = 2;           % minimum life before instruments are upgraded (years)
min_replace_time = 2;           % minimum life before components are replaced (years)

% set risk probabilities
risk_launch_fail = 0.07;        % risk of launch failure (replacement case only)
risk_service_fail = 0.10;       % risk of servicing mission failure
risk_cat_fail = 0.02;           % risk of catastrophic mission failure

% set cost parameters
learn_curve = 0.95;             % learning curve (SMAD pg 809)
fab_fraction = 0.6;             % fraction of total RDT&E cost for fabrication
serviceable_inc = 1.15;         % increase in cost to make the telescope serviceable
cost_replace = 5;               % cost to replace each component ($M)
```

```
%ITERATEREPLACE Script for Monte-Carlo simulation for telescope replacement
%
% Mark Baldesarra
% October 3, 2006

constInitialize;

% preallocate output variables
start_time1     = zeros(1,num_iter);
start_time2     = zeros(1,num_iter);
util1           = zeros(1,num_iter);
util2           = zeros(1,num_iter);
offline_time    = zeros(1,num_iter);
cum_utility     = zeros(1,num_iter);
EOL             = zeros(1,num_iter);
overlap         = zeros(1,num_iter);

j = 1;
replace_wait_count = 0;
tic
for i=1:num_iter

    % run simulation
    [comp_fails1 pfo_curve1 fail_time(i,1)] = telescopeSim_Replace(dt);
    [comp_fails2 pfo_curve2 fail_time(i,2)] = telescopeSim_Replace(dt);

    % find period when the first telescope goes below PFO threshold
    for q=1:length(pfo_curve1)
        if pfo_curve1(q) < min_pfo
            threshold_pd = q;
            break
        end
    end

    % find overall utility of first telescope
    util1(i) = utilityInstrument(0) * fail_time(i,1);

    if rand() > risk_launch_fail

        % find launch time of second telescope
        if (fixed_schedule_replace == 'y')
            start_time2(i) = fixed_replace_time;
        else
            start_time2(i) = threshold_pd*dt + replace_latency;
        end

        % find overall utility of second telescope
        util2(i) = utilityInstrument(start_time2(i) - upgrade_latency) * fail_time(i,2);

        % find EOL of telescope program
        EOL(i) = start_time2(i) + fail_time(i,2);

        % calculate secondary metrics
        offline_time(j) = max(start_time2(i) - fail_time(i,1), 0);
```

```
        overlap(j) = max(fail_time(i,1) - start_time2(i), 0);
        j = j+1;

    else
        EOL(i) = fail_time(i,1);
        util2(i) = 0;
    end

    cum_utility(i) = util1(i) + util2(i);

    % Feedback to user during simulation
    if rem(i,num_iter/100) == 0
        clc
        disp(sprintf('Simulation %d%% complete',i/num_iter*100))
    end

end
time = toc;
disp(sprintf('Simulation completed in %2.1g minutes',time/60));

% run cost analysis
total_cost = costAnalysis_Replace(drymass, EOL, fab_fraction, learn_curve, 0);

plot_type = 'replace';
```

```
function [comp_fails pfo_curve fail_time] = telescopeSim_Replace(dt)
%TELESCOPESIM_REPLACE Simulates the performance of a telescope over time without repair
%   This function runs a probabilistic simulation of a telescope and
%   determines when the telescope fails.
%
% INPUTS
%
% OUTPUTS
% comp_fails        vector that denotes which components failed
% pfo_curve         "Probability of Future Operations" curve over time
% fail_time         period at the end of which the system fails
%
% Mark Baldesarra
% July 13, 2006

compInitialize;
EOL_flag = 0;

while ~EOL_flag && (current_period < max_period)

    current_period = current_period + 1;
    current_time = current_time + dt;

    for i=1:num_set
        [comp_set{i} fail_flag(i)] = generalFail(comp_set{i}, num_comp(i),...
            num_ops(i), num_needed(i), dt, hazard_ptr{i});
end

    % calculate reliability of each component set
    for i=1:num_set
        set_reliability(i) = computeSetReliability(comp_set{i}, num_comp(i),...
            num_needed(i), dt, future_rel_yrs, reliability_ptr{i});
    end

    % compute overall system reliability and store in output
    total_rel = prod(set_reliability);
    pfo_curve(current_period) = total_rel;

    % Check if telescope fails
    if (any(fail_flag))
        EOL_flag = 1;
        comp_fails = fail_flag;
        fail_time = current_period*dt;
    end

    if current_period == max_period
        disp('Hey! out of range!')
    end
end
```

```
%ITERATESERVICE Script for Monte-Carlo simulation for telescope servicing
%
% Mark Baldesarra
% July 11, 2006

constInitialize;

% preallocate output variables
[upgrade_times{1:num_iter}]  = deal(0);
[fail_times{1:num_iter}]     = deal(0);
[repair_times{1:num_iter}]   = deal(0);
[repair_set{1:num_iter}]     = deal(0);
[comp_fails{1:num_iter}]     = deal(0);
[utility{1:num_iter}]        = deal(0);
[pfo_curve{1:num_iter}]      = deal(0);

offline_time    = zeros(1,num_iter);
cum_utility     = zeros(1,num_iter);
EOL             = zeros(1,num_iter);
comp_replace    = zeros(1,num_iter);
num_missions    = zeros(1,num_iter);
num_upgrades    = zeros(1,num_iter);
first_repair    = zeros(1,num_iter);

% run Monte-Carlo simulation
for i=1:num_iter
    % input additional parameters if this is a sensitivity analysis
    if sens_flag
        [upgrade_times{i} fail_times{i} offline_time(i) repair_times{i} repair_set{i}...
            comp_fails{i} utility{i} cum_utility(i) EOL(i) pfo_curve{i}] = ...
            telescopeSim_Service(num_serv,dt,fixed_schedule_service,0,sens_name,sens_param);
    else
        [upgrade_times{i} fail_times{i} offline_time(i) repair_times{i} repair_set{i}...
            comp_fails{i} utility{i} cum_utility(i) EOL(i) pfo_curve{i}] = ...
            telescopeSim_Service(num_serv,dt,fixed_schedule_service,0);
    end

    num_missions(i) = length(repair_times{i});
    num_upgrades(i) = length(upgrade_times{i});
    first_repair(i) = repair_times{i}(1);
    comp_replace(i) = sum(repair_set{i});

end

% run cost analysis
total_cost = costAnalysis_Servicing(drymass, serviceable_inc, EOL, num_serv,...
    cost_replace, comp_replace, 0);

plot_type = 'service';
```

```
function [upgrade_times fail_times offline_time repair_times repair_set...
    comp_fails utility cum_utility EOL pfo_curve] = ...
    telescopeSim_Service(num_serv, dt, fixed_flag, diag_flag, varargin)
%TELESCOPESIM_SERVICE Simulates the performance of a serviceable telescope over time.
%   This function runs a probabilistic simulation of a telescope and
%   determines when the telescope is upgraded and repaired, given a maximum
%   number of servicing missions over the design lifetime.
%
% INPUTS
% num_serv          maximum number of servicing missions allowed
% dt                time step (years)
% fixed_flag        set if servicing schedule is fixed ('y' = fixed)
% diag_flag         toggle diagnostic messages (1 = messages on)
%
% VARIABLE INPUTS
% Variables for sensitivity analysis are passed into the function as pairs:
% the first is the name of the variable as a string, and the second is the
% value of the variable.
%
% OUTPUTS
% upgrade_times     vector of times where upgrades were made (years)
% fail_times        vector of times where failures occurred (years)
% offline_time      total time that the telescope is offline (years)
% repair_times      vector of times where repairs were made (years)
% repair_set        vector of components repaired during the mission
% comp_fails        vector that counts which components failed
% utility           vector of utility of telescope instruments through time
% cum_utility       cumulative science output (CSO) of telescope
% EOL               time when telescope operations are terminated (year)
% pfo_curve         "Probability of Future Operations" curve over time
%
% Mark Baldesarra
% July 31, 2006

compInitialize;

% check if varargin has even number of components
if rem(length(varargin),2) ~= 0
    error('Inputs to telescopeSim_Service has odd number of components')
end

% store additional variables passed into function for sensitivity analysis
% (will overwrite parameter that was initialized by compInitialize)
for i=1:length(varargin)/2
    expression = [varargin{2*i-1}, '=', num2str(varargin{2*i})];
    evalc(expression);
end

% NOTE OF CAUTION: Most calculations within the loop are done using periods
% rather than years. The quantities are converted to years prior to output

while ~EOL_flag

    % increment quantities by one period
```

```
time_since_upgrade = time_since_upgrade + dt;
current_period = current_period + 1;
current_time = current_time + dt;

if ops_flag
    % increment operational times of components and determine if any
    % components have failed
    for i=1:num_set
        [comp_set{i} fail_flag(i)] = generalFail(comp_set{i}, num_comp(i),...
            num_ops(i), num_needed(i), dt, hazard_ptr{i});
    end

    % calculate reliability of each component set
    for i=1:num_set
        set_reliability(i) = computeSetReliability(comp_set{i}, num_comp(i),...
            num_needed(i), dt, future_rel_yrs, reliability_ptr{i});
    end
end

% compute overall system reliability and store in output
total_rel = prod(set_reliability);
pfo_curve(current_period) = total_rel;

% *** DECISION ANALYSIS STARTS HERE ***

% check if the telescope is allowed to be serviced
if length(repair_times) >= num_serv
    service_allow_flag = 0;
elseif current_time >= max_life
    service_allow_flag = 0;
else
    service_allow_flag = 1;
end

% for the fixed servicing schedule case, check if the telescope is
% scheduled to be serviced in the current period
if strcmp(fixed_flag,'y')
    servicing_periods = round(max_period*(1:num_serv)./(num_serv+1));
    if (any(current_period == servicing_periods))
        sched_flag = 1;
        service_now_flag = 1;
    else
        sched_flag = 0;
    end
end

% check if failure occurred and store current period if yes
if any(fail_flag)
    fail_times = [fail_times current_time];
    ops_flag = 0;
end

% if the telescope fails and the telescope is not to be serviced,
% signal termination of operations (EOL)
```

```
if (any(fail_flag) && ~service_allow_flag)
    EOL_flag = 1;
    EOL = current_period * dt;
    break

% perform servicing if you are required to service now AND the
% servicing spacecraft is ready
elseif service_now_flag && (serv_wait_time >= serv_latency || sched_flag)

    % DIAGNOSTIC TEXT
    if diag_flag
        disp(sprintf('Servicing now: time %2.2f',current_time))
    end

    % store current period as a repair time and add the components that
    % have failed to the "comp_fails" cumulative vector
    repair_times = [repair_times current_time];
    comp_fails = comp_fails + fail_flag;

    % check if a catastrophic failure occurs, in which case signal EOL
    if (rand() < risk_cat_fail)

        % DIAGNOSTIC TEXT
        if diag_flag
            disp(sprintf('WARNING: Catastrophic failure: time %2.2f',current_time))
        end

        EOL_flag = 1;
        EOL = current_period * dt;
        break
    end

    % check if the repair mission fails, thus the components are not
    % modified and the utility of the spacecraft is zero
    if (rand() < risk_service_fail)

        % DIAGNOSTIC TEXT
        if diag_flag
            disp(sprintf('WARNING: Servicing failure: time %2.2f',current_time))
        end

        % maintain same utility as last period
        utility = [utility utility(length(utility))];
        serv_wait_time = 0;
        service_now_flag = 1;

    else
        % repair components as needed
        [comp_set repair_set] = telescopeRepair(comp_set, repair_set, num_set,...
            num_comp, min_replace_time, diag_flag);

        % set flags to signify operational status & no need to service
        ops_flag = 1;
```

```
            service_now_flag = 0;

            % reset the servicing waiting time counter
            serv_wait_time = 0;

            % The telescope will be upgraded if it has been at least
            % "min_upgrade_time" since the last upgrade
            if (time_since_upgrade >= min_upgrade_time)
                % perform upgrade operations and reset upgrade timer
                [upgrade_times utility] = telescopeUpgrade(upgrade_times, utility,...
                    current_time, dt, upgrade_latency);
                time_since_upgrade = 0;
            else
                utility = [utility utility(length(utility))];
            end

        end

        % DIAGNOSTIC TEXT
        if diag_flag
            disp(sprintf('---------------------------------------'))
            disp(sprintf(' '))
        end

    % Set servicing flag if either there was a failure OR the telescope is
    % unreliable OR the telescope is inoperative
    elseif service_allow_flag && (any(fail_flag) || any(total_rel < min_pfo) || ~ops_flag)
        service_now_flag = 1;
        serv_wait_time = serv_wait_time + dt;
        utility = [utility utility(length(utility))];

    else
        ops_flag = 1;
        utility = [utility utility(length(utility))];
    end

    % if the telescope is inoperative, set the curret utility of the
    % telescope (the last entry in the "utility" vector") to zero.
    if ~ops_flag
        utility(length(utility)) = 0;
        offline_time = offline_time + dt;
    end

    % ERROR CHECKING STEP: stop on "infinite loop"
    if current_time == 3*max_life
        error('Program stopped: infinite loop detected')
    end
end

% compute cumulative utility over telescope life (Riemann sum)
cum_utility   = sum(utility) * dt;
```

A.2 Performance Models

```
function total_rel = ...
    computeSetReliability(comp_set, num_comp, num_needed, dt, num_years, rel_fcn)
%COMPUTESETRELIABILITY Determines the overall reliability of a component set
%   Computes the reliability of a set of components where
%   determines when the telescope fails.
%
% INPUTS
% comp_set          matrix describing the component set
% num_comp          number of components in the set
% num_needed        number of working components needed for operations
% num_years         years to look forward into the future for reliability
% dt                length of period expressed in years
% rel_fcn           function that computes the reliability of the component
%
% OUTPUTS
% total_rel         overall current reliability of component set
%
% VERIFICATION
% Tested using the standard "R out of N" combinatorial formula (applicable
% only for identical elements) and hand calculations.
%
% Mark Baldesarra
% July 24, 2006

% Step 1: Generate all possible combinations of operational components
% (N choose K, where N is # of components and K is # required to operate)

N = [];
for i=num_needed:num_comp
    % add rows for (num_comp choose i) combinations
    N = [N; nchoosek(1:num_comp,i) zeros(nchoosek(num_comp,i),num_comp-i)];
end

% Step 2: Generate a matrix comp_status where 1's represent failed
% components, and 0's represent working components

[row col] = size(N);
comp_status = ones(row,col);

for i=1:row
    for j=1:col
        if N(i,j)
```

```
            comp_status(i,N(i,j)) = 0;
        end
    end
end

% Step 3: Determine reliability vector "comp_reliability" by evaluating the
% reliability of each component in the set using the "rel_fcn"

comp_reliability = [];
for i=1:num_comp

    % if the component has failed, make reliability zero
    if comp_set(i,1) == 0
        comp_reliability = [comp_reliability 0];
    else

        if isa(rel_fcn, 'function_handle')
            reliability = rel_fcn(comp_set(i,2), dt, num_years);
        else
            reliability = exp(-rel_fcn*num_years);
        end

        comp_reliability = [comp_reliability reliability];
    end
end

% Step 4: Compute overall reliability of component set by adding the
% probabilities of all possible combinations of failed/working components
% that lead to a set that is still operational

total_rel = 0;
for i=1:row

    % The probability of the combination is equal to the product of the
    % reliability / failure of each component, where the status of each
    % component is stored in row i of matrix "comp_status".

    % For example, if the reliabilities are [0.8 0.9 0.7] and the row of matrix
    % "comp_status" is [1 0 0], probability of this combo is (1-0.8)*(0.9)*(0.7)

    prob_comb = prod(abs(comp_status(i,:)-comp_reliability));
    total_rel = total_rel + prob_comb;
end
```

```
function comp_set = createComponent(num_comp)
%CREATECOMPONENT Creates array to represent the component set
%   A set of components is represented as a (num_comp x 2) array, where
%   each row is one component in the set.
%
%   * The first column is either a 1 (available) or 0 (failed/offline).
%   * The second column stores the number of periods that the component has
%     been operational. While a component is held in reserve, the number of
%     periods is held at 0.
%
%   Mark Baldesarra
%   July 11, 2006

comp_set = [ones(num_comp,1) zeros(num_comp,1)];
```

```
function [comp_set fail_flag] = ...
    generalFail(comp_set, num_comp, num_ops, num_needed, dt, fail_fcn)
%GENERALFAIL Generalized component failure function
%   For a set of num_comp components, of which num_needed are operational
%   at one time, computes which components are operating, which fail and
%   which remain online.
%
% INPUTS
% comp_set      (num_comp x 2) array that represents the component set
% num_comp      Total number of components in the set
% num_ops       Number of components operational at one time
% num_needed    Minimum number of working components for the set to be operational
% dt            Length (in years) of one period
% fail_fcn      Either the failure rate (per year) or pointer to hazard rate function
%
% OUTPUTS
% comp_set      Updated component set array
% fail_flag     If the component set has "failed", set equal to 1
%
% Mark Baldesarra
% July 11, 2006

% "online_count" tracks how many components have been set as online for the
% period in question
online_count = 0;
fail_flag = 0;

% iterate over all components
for i=1:num_comp

    % check if current component (row) is operational and must be online
    if (comp_set(i,1) && online_count < num_ops)

        % increment online time for the component and the online count
        comp_set(i,2) = comp_set(i,2) + dt;
        online_count = online_count + 1;

        if isa(fail_fcn, 'function_handle')
            fail_prob = fail_fcn(comp_set(i,2), dt);
        else
            fail_prob = 1-exp(-fail_fcn*dt);
        end
```

```
        % if the component fails, set to offline
        if rand() <= fail_prob
            comp_set(i,1) = 0;
        end

    end
end

% check if there are insufficient online components
if sum(comp_set(:,1)) < num_needed
    fail_flag = 1;
end
```

```
function h = hazardGyro(t, dt)
%HAZARDGYRO Outputs the gyro failure rate given the current period.
%
% INPUTS
% t              current time of operation for the component (years)
% dt             length of one period (years)
%
% OUTPUTS
% h              probability of failure in next period (time t < T < t+dt)
%
% Mark Baldesarra
% July 11, 2006

eta  = 5.89;
beta = 4.82;

% Rate of Failures in time period "pd"
fail_rate = (beta/eta).*(t./eta).^(beta-1);

% Probability of Failure in one period
h = 1-exp(-fail_rate*dt);
```

```
function pfo = reliabilityGyro(t, dt, num_years)
%RELIABILITYGYRO Calculate future reliability of a gyroscope
% Determines the probability of a gyroscope that is operational at period
% "pd" to be operational "num_years" from that period
%
% INPUTS
% t             current age of gyroscope (years)
% dt            period length (years)
% num_years     future time horizon (years)
%
% OUTPUTS
% pfo           probability of working "num_years" years from period "pd"
%
% Mark Baldesarra
% August 22, 2006

eta  = 5.89;
beta = 4.82;

% probability of failure within t years (from Weibull CDF)
F1 = exp(-(t./eta).^beta);

% probability of failure within t+2 years (from Weibull CDF)
F2 = exp(-((t+num_years)./eta).^beta);

pfo = F2./F1;
```

```
function [util] = utilityInstrument(t)
%UTILITYINSTRUMENT Summary of this function goes here
%
% INPUTS
% t         time
%
% OUTPUTS
% util      utility of instrument technology at time "t"
%
% Mark Baldesarra
% October 2, 2006

p = 0.3218;

util = 10*exp(p * t);
```

```
function [comp_set repair_set] = ...
    telescopeRepair(comp_set, repair_set, num_set, num_comp, min_replace_time, diag_flag)
%TELESCOPEREPAIR Perform repair operations on telescope parameters.
%   Updates the "comp_set" and "repair_set" variables with the new values
%   given when the telescope is repaired
%
% INPUTS
% comp_set          cell array with all component sets
% repair_set        vector with record of all repairs made to telescope
% num_set           number of component sets
% num_comp          array with number of components in each set
% min_replace_time  minimum life of components replaced during servicing
% diag_flag         toggles diagnostic text on/off
%
% OUTPUTS
% comp_set      updated "comp_set" cell array (with latest values)
% repair_set    updated "repair_set" vector (with latest values)
%
% Mark Baldesarra
% July 18, 2006

for i=1:num_set
    for j=1:num_comp(i)

        % replace if failed or operating for over specified # periods
        if (comp_set{i}(j,1) == 0) || (comp_set{i}(j,2) > min_replace_time)

            % record which components were repaired
            repair_set(i) = repair_set(i) + 1;

            % repair components (return to available status & new)
            comp_set{i}(j,1) = 1;
            comp_set{i}(j,2) = 0;
        end
    end
end

if diag_flag
    disp(sprintf('After servicing:'));
    for i=1:num_set
        disp(comp_set{i}(:,:)');
    end
end
```

```
function [upgrade_times utility] = ...
    telescopeUpgrade(upgrade_times, utility, t, dt, upgrade_latency)
%TELESCOPEUPGRADE Perform upgrade operations on telescope parameters.
%   Updates the "upgrade_times" and "utility" variables with the new values
%   given when the telescope is serviced in "current_period"
%
% INPUTS
% upgrade_times     vector with the periods when upgrades took place
% utility           vector with the utilities of previous telescope instruments
% t                 the current time in the analysis
% dt                length of one time period (years)
% upgrade_latency   time before servicing that the utility of telescope is set
%
% OUTPUTS
% upgrade_times     updated "upgrade_times" vector (with latest value)
% utility           updated "utility" vector (with latest value)
%
% Mark Baldesarra
% July 18, 2006

% compute utility of new instrument (evaluated at "upgrade_latency" years
% before current time "t")
new_util = utilityInstrument((t - upgrade_latency));

% update utility and upgrade_times variables with new values
upgrade_times = [upgrade_times t];
utility = [utility new_util];
```

A.3 Cost Models

```
function [total_cost, varargout] = ...
    costAnalysis_Replace(drymass, lifetime, fab_fraction, learn_curve, plot_flag)
%COSTANALYSIS_REPLACE Find total lifecycle cost of a two-telescope program.
%
% INPUTS
% drymass           mass of telescope (kg)
% lifetime          total lifetime of telescope (years)
% fab_fraction      fabrication fraction of initial cost
% learn_curve       fraction of init cost that the 2nd iteration will cost
% plot_flag         flag = 1 to plot the components of cost
%
% OUTPUTS
% total_cost        total lifecycle cost of the replacement program
%
% VARIABLE OUTPUTS
% Will output the following when requested (in order):
% [cost_init1 cost_init2 cost_launch cost_ops cost_sci]
%
% Mark Baldesarra
% August 22, 2006

% RDT&E / Fabrication cost for telescope 1
cost_init1 = costInitial(drymass);

% RDT&E / Fabrication cost for telescope 2
% Learning curve equation from SMAD pg 809
cost_init2 = costInitial(drymass) * fab_fraction * (2^(1-log(1/learn_curve)/log(2))-1);

% Launch of telescope cost
cost_launch = costLaunch(drymass);

% Operations cost
cost_ops = costOps(cost_init1) * lifetime;

% Science Instruments cost
cost_sci = costScience(drymass) * 2;

% calculate total cost and maximum servicing price
total_cost = cost_init1 + cost_init2 + cost_launch*2 + cost_ops + cost_sci;

% plot components of total cost (if requested)
if plot_flag == 1
```

```
    figure
    area(drymass',[cost_init1; cost_init2; cost_launch*2; cost_ops; cost_sci]')
    title('Lifecycle Cost of Telescope Replacement Program')
    xlabel('Telescope Dry Mass (kg)')
    ylabel('Total Lifecycle Cost ($M FY2004)')
    legend('Initial Cost + Telescope 1','Cost of Telescope 2','Launch Costs',...
        'Operations Costs','Science Instruments',2)
end

% output cost breakdown data (if requested)
if (nargout > 1)
    varargout{1} = cost_init1;
    varargout{2} = cost_init2;
    varargout{3} = cost_launch;
    varargout{4} = cost_ops;
    varargout{5} = cost_sci;
end
```

```
function total_cost = costAnalysis_Servicing(drymass, serviceable_inc, EOL,...
    num_serv, cost_replace, comp_replace, plot_flag, varargin)
%COSTANALYSIS_SERVICING Find maximum servicing price.
%
% INPUTS
% drymass           mass of telescope (kg)
% serviceable_inc   amount that RDT&E and Fab is incremented to allow for
%                   servicing (i.e. a 10% increase => variable = 1.10
% EOL               total lifetime of telescope (years)
% num_serv          total number of servicing missions
% cost_replace      vector of component costs, OR single cost
% comp_replace      cell array with components that were replaced, OR
%                   single aggregate number of components
% plot_flag         flag = 1 to plot components of total budget
%
% VARIABLE INPUTS
% budget            budgetary constraint (for plotting only)
%
% OUTPUTS
% total_cost        total lifecycle cost of the servicing program
%
% Mark Baldesarra
% August 22, 2006

if plot_flag == 1
    budget = varargin{1};
end

% RDT&E / Fabrication cost
cost_init = costInitial(drymass) * serviceable_inc;

% Launch cost
cost_launch = costLaunch(drymass);

% Operations cost
cost_ops = costOps(cost_init) * EOL;

% Science Instruments cost
cost_sci = costScience(drymass) * (num_serv+1);

% Components cost
if iscell(comp_replace)
    cost_repairs = costComponents(cost_replace, comp_replace);
```

```
else
    cost_repairs = cost_replace * comp_replace;
end

% calculate total cost and maximum servicing price
total_cost = cost_init + cost_launch + cost_ops + cost_sci + cost_repairs;

% plot components of total cost (if requested)
if plot_flag
    figure
    hold on
    area(drymass',[cost_init; cost_repairs*ones(1,length(drymass));...
        cost_launch; cost_ops; cost_sci;]')
    plot(drymass,budget)
    hold off
    title('Lifecycle Cost of Telescope Servicing Program')
    xlabel('Telescope Mass (kg)')
    ylabel('Total Lifecycle Cost ($M FY2004)')
    legend('Initial Cost', 'Component Replacement', 'Launch Costs',...
        'Operations Costs', 'Science Instruments', 'BUDGET',2)
end
```

```
function cost = costComponents(cost_replace, comp_replace)
%COSTSCIENCE Determine cost of telescope instruments
%
% INPUTS
% cost_replace  Vector of component costs ($M, FY2004)
% comp_replace  Cell array of cells arrays that indicate which components
%               were replaced during each servicing mission.
%
% OUTPUTS
% cost          Cost of telescope instrument ($M, FY2004)
%
% Mark Baldesarra
% August 24, 2006

cost = 0;

for i=1:length(comp_replace)
    for j=1:length(comp_replace{i})
        num_comp_replace = sum(comp_replace{i}{j});
        cost = cost + cost_replace(i) * num_comp_replace;
    end
end
```

```
function cost = costInitial(drymass)
%COSTINITIAL Determine initial cost of telescope
%   Calculates the RDT&E and fabrication cost of the first copy of a
%   telescope design. Based on the Advanced Missions Cost Model (AMCM) by
%   NASA JSC (http://www1.jsc.nasa.gov/bu2/AMCM.html)
%
% INPUTS
% drymass   Mass of telescope (kg)
%
% OUTPUTS
% cost      Total initial telescope cost ($M, FY2004)
%
% Mark Baldesarra
% August 21, 2006

a=0.000504839;
b=0.594183076;
c=0.653947922;
d=76.99939424;
e=1.68051e-52;
f=-0.355322218;
g=1.554982942;

B = 1;
D = 1;
Q = 1;
S = 2.17;

IOC = 2004;
inf91 = 1.414;

% change to weight in pounds
W = drymass*2.2;

% compute total cost (express in $M)
cost = a * power(Q,b) * power(W,c) * power(d,S) * power(e,(1/(IOC-1900))) *...
    power(B,f) * power(g,D) * inf91;
```

```
function cost = costLaunch(drymass)
%COSTLAUNCH Computes launch cost for a given spacecraft mass
%
% INPUTS
% drymass   total spacecraft mass launched to LEO (185-200km)
%
% OUTPUTS
% cost      total launch cost ($M, FY2004)
%
% Mark Baldesarra
% August 17, 2006

% Falcon 1 LV Family
if (drymass >= 0) && (drymass < 668)
    cost(i) = 6;

% *** Placeholder 1 ***
elseif (drymass >= 668) && (drymass < 1220)
    cost(i) = 0.02536*drymass - 10.94;

% Taurus LV Family
elseif (drymass >= 1220) && (drymass < 1900)
    cost(i) = -0.0000279*drymass^2 + 0.1039*drymass  - 64.7;

% Delta II LV Family
elseif (drymass >= 1900) && (drymass < 5139)
    cost(i) = 0.0044*drymass + 27.381;

% Linear interpolation
elseif (drymass >= 5139) && (drymass < 8600)
    cost(i) = 0.02398*drymass - 73.24;

% Delta IV M+ LV Family
elseif (drymass >= 8600) && (drymass < 11700)
    cost(i) = 0.0016*drymass + 119.13;

% Atlas V LV Family
elseif (drymass >= 11700) && (drymass < 20050)
    cost(i) = 1.332e-6*drymass^2 - 0.02827*drymass + 283.21;
else cost(i) = NaN;
end

cost = cost .* 0.908; % deflate cost to FY2000 dollars
```

```
function cost = costOps(cost_init)
%COSTOPS Determine yearly operations cost of a telescope
%   Based on the Mission Operations Cost Model (MOCM) by NASA JSC
%   (http://www1.jsc.nasa.gov/bu2/MOCM.html)
%
% INPUTS
% cost_init Initial cost of telescope ($M, FY2004)
%
% OUTPUTS
% cost      Yearly telescope operations cost ($M, FY2004)
%
% Mark Baldesarra
% August 21, 2006

% set parameters (from MOCM)
INF87= 1.689;
a = 0.047;
b = 0.878;
c = cost_init / INF87;

% compute operations cost (express in $M)
cost = a * power(c,b) * INF87;
```

```
function cost = costScience(drymass)
%COSTSCIENCE Determine cost of telescope instruments
%
% INPUTS
% drymass   Dry mass of telescope (kg)
%
% OUTPUTS
% cost      Cost of telescope instrument ($M, FY2004)
%
% Mark Baldesarra
% August 24, 2006

% Assume directly proportional to mass (known $100M cost for 11mt telescope)
cost = 100*(drymass/11000);
```

Bibliography

[1] "Hubble Space Telescope Servicing Mission 3B Media Guide," Tech. Rep. NP-2001-12-412-GSFC, NASA Goddard Space Flight Center, 2001.

[2] SPACE.com, "NASA's Mission to Service Hubble in 2008 Will Cost $900 Million," Web site, 2007, http://www.space.com/news/061013_hubble_cost.html.

[3] "Costs for Hubble Servicing Mission and Implementation of Safety Recommendations Not Yet Definitive," Tech. Rep. GAO-05-34, United States Government Accountability Office, 2004.

[4] Lillie, C. F., "On-Orbit Servicing for Future Space Observatories," *Space 2005*, Long Beach, CA, 30 Aug - 1 Sep 2005, AIAA 2005-6609.

[5] de Weck, O. L., de Neufville, R., and Chaize, M., "Staged Deployment of Communications Satellite Constellations in Low Earth Orbit," *Journal of Aerospace Computing, Information, and Communication*, Vol. 1, No. 3, 2004, pp. 119–136.

[6] Park, C. S., *Contemporary Engineering Economics*, Prentice-Hall, Upper Saddle River, NJ, 2002.

[7] Lester, D. and Lillie, C. F., "Servicing the Single Aperture Far Infrared (SAFIR) Telescope from a Lunar Exploration Enabled Gateway," *Space Resources Roundtable VII*, League City, TX, 25-28 Oct 2005.

[8] Murman, E., *Lean Enterprise Value: Insights from MIT's Lean Aerospace Initiative*, Palgrave, New York, NY, 2002.

[9] Rouse, W. B. and Boff, K. R., "Strategies for Value: Quality, Productivity, and Innovation in R&D/Technology Organizations," *Systems Engineering*, Vol. 4, No. 2, 2001, pp. 87–106.

[10] Crawley, E. F., "System Architecture: IAP Lecture 2," Course notes.

[11] de Neufville, R., "16.861 Engineering Systems Analysis for Design," Course notes.

[12] Trigegoris, L., *Real Options: Managerial Flexibility and Strategy in Resource Allocation*, MIT Press, Cambridge, MA, 1996.

[13] Saleh, J. H., *Weaving Time Into System Architecture: New Perspectives On Flexibility, Spacecraft Design Lifetime, and On-Orbit Servicing*, Ph.D. thesis, Massachusetts Institute of Technology, 2002.

[14] Nilchiani, R., Hastings, D. E., and Joppin, C., "Calculations of Flexibility in Space Systems," Summary report.

[15] Space Studies Board, *Extending the Effective Lifetimes of Earth Observing Research Missions*, National Research Council of the National Academies, Washington, DC, 2005.

[16] Joppin, C. and Hastings, D. E., "Evaluation of the Value of the Flexibility Offered by On-orbit Servicing - Case of Satellite Upgrade," *Space 2003 Conference and Exhibit*, 23-25 Sep 2003, AIAA 2003-6366.

[17] Joppin, C. and Hastings, D. E., "Upgrade and Repair of a Scientific Mission Using On-Orbit Servicing Based on the Hubble Space Telescope Example," *Space 2004 Conference and Exhibit*, San Diego, CA, United States, 28-30 Sep 2004, AIAA 2004-6052.

[18] NASA, "A Brief History of the Hubble Space Telescope," Web site, 2007, http://history.nasa.gov/hubble/index.html.

[19] Doxsey, R. E., "15 Years of Hubble Space Telescope Science Operations," *SpaceOps 2006*, AIAA, 19-23 June 2006, AIAA 2006-5936.

[20] "NASA Facts: Servicing History and Long-Term Plans," Tech. Rep. NF 195, NASA Goddard Space Flight Center, 1993.

[21] NASA, "The Hubble Program - Servicing Missions - Introduction," Web site, 2006, http://hubble.nasa.gov/missions/intro.php.

[22] HubbleSite, "Team Hubble - Servicing Missions," Web site, 2006, http://hubblesite.org/the_telescope/team_hubble/servicing_missions.php.

[23] Waltz, D. M., *On-orbit servicing of space systems*, Krieger Publishing Co., Melbourne, FL, 1993.

[24] NASA, "The Hubble Program - Technology - Optical Assembly," Web site, 2006, http://hubble.nasa.gov/technology/optics.php.

[25] van der Marel, R. P., "Best Gyroscope Usage to Maximize the HST Mission Lifetime," Tech. Rep. ACS 2004-03, Space Telescope Science Institute, 2004.

[26] Stocker, R. B., "Corrective Optics Space Telescope Axial Replacement (COSTAR)," *Space Programs and Technologies Conference and Exhibit*, Huntsville, AL, 21-23 Sep 1993, AIAA 1993-4206.

[27] Dedalis, R. P. and Mitchell, P. L., "Servicing The Hubble: Risk Mitigation, Lessons Learned, And Rewards In Completing Hubble Space Telescope Servicing Missions," *Space 2004 Conference and Exhibit*, San Diego, CA, 28-30 Sep 2004, AIAA 2004-5831.

[28] Johns Hopkins University, "ACS: The Advanced Camera for Surveys," Web site, 2007, http://acs.pha.jhu.edu/.

[29] NASA, "The Hubble Program - Technology - Instruments," Web site, 2006, http://hubble.nasa.gov/technology/instruments.php.

[30] Kasai, T., Oda, M., and Suzuki, T., "Results of the ETS-7 Mission - Rendezvous docking and space robotics experiments," *iSAIRAS'99; Proceedings of the Fifth International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Noordwijk, Netherlands, 1-3 June 1999.

[31] Yoshida, K., Hashizume, K., Nenchev, D. N., Inaba, N., and Oda, M., "Control of a space manipulator for autonomous target capture - ETS-VII flight experiments and analysis," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, 14-17 Aug 2000, AIAA 2000-4376.

[32] SpaceDaily, "ETS-7 Orbital Rendezvous and Robotic Mission," Web site, 1997, http://www.spacer.com/spacenet/text/ets7-b.html.

[33] "DART Demonstrator To Test Future Autonomous Rendezvous Technologies in Orbit," Tech. Rep. FS-2004-08-113-MSFC, NASA Marshall Space Flight Center, 2004.

[34] Orbital Corporation, "DART Rendezvous Vehicle," Web site, 2006, http://www.orbital.com/AdvancedSpace/DART/.

[35] Iannotta, B., "SUMO wrestles satellites into new orbits," *Aerospace America*, Feb 2006, pp. 26.

[36] Naval Research Laboratory, "SUMO," Web site, 2006, http://projects.nrl.navy.mil/sumo/home.php.

[37] Reynerson, C. M., "Spacecraft servicing - First order model for feasibility and cost effectiveness," *Space 2001 Conference and Exposition*, Albuquerque, NM, Aug 2001, AIAA 2001-4732.

[38] Amato, M. J., Benford, D. J., Moseley, S. H., and Roman, J. A., "Science Promise and Conceptual Mission Design Study for SAFIR - the Single Aperture Far Infrared Observatory," Tech. Rep. NRA-03-OSS-01-VM, University of Texas at Austin, 2003.

[39] Moe, R., Mather, J., Lester, D., Ward, B., and Derkowski, B., "Application of In-space Capabilities to a Large Infrared Telescope for Astronomy-SAFIR," *1st Space Exploration Conference: Continuing the Voyage of Discovery*, Orlando, FL, 30 Jan-1 Feb 2005, AIAA 2005-2686.

[40] Lester, D., Friedman, E., and Lillie, C. F., "Strategies for servicing the Single Aperture Far IR (SAFIR) telescope," *UV/Optical/IR Space Technologies: Innovative Technologies and Concepts II*, San Diego, CA, United States, 31 July-1 Aug 2005, SPIE-5899.

[41] Long, A. and Hastings, D. E., "The Effect of Uncertainties in Architectural Design Elements on Potential Non-Commercial On-Orbit Satellite Servicing Markets," *Space 2005*, Long Beach, CA, 30 Aug - 1 Sep 2005, AIAA 2005-6699.

[42] Saleh, J. H., Torres-Padilla, J.-P., Hastings, D. E., and Newman, D. J., "To Reduce or to Extend a Spacecraft Design Lifetime?" *Journal of Spacecraft and Rockets*, Vol. 43, No. 1, 2006, pp. 207–217.

[43] Saleh, J. H., Lamassoure, E. S., and Hastings, D. E., "Space systems flexibility provided by on-orbit servicing, part I," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, 2002, pp. 551–560.

[44] Lamassoure, E. S., Saleh, J. H., and Hastings, D. E., "Space systems flexibility provided by on-orbit servicing, part II," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, 2002, pp. 561–570.

[45] Saleh, J. H., Lamassoure, E. S., Hastings, D. E., and Newman, D. J., "Flexibility and the value of on-orbit servicing - New customer-centric perspective," *Journal of Spacecraft and Rockets*, Vol. 40, No. 2, 2003, pp. 279–291.

[46] NASA Marshall Space Flight Center, "NASA/Marshall Solar Physics: The Solar Maximum Mission," Web site, 2007, http://solarscience.msfc.nasa.gov/SMM.shtml.

[47] Beichman, C. A., Coulter, D. R., Lindensmith, C. A., and Lawson, P. R., "Summary Report on Architecture Studies for the Terrestrial Planet Finder," Tech. Rep. JPL 02-011, Jet Propulsion Laboratory, 2002.

[48] de Neufville, R., *Applied Systems Analysis: Engineering Planning and Technology Management*, McGraw-Hill Publishing Company, New York, NY, 1990.

[49] Apostolakis, G. E., "Probability and its Applications to Quality Control, Risk and Reliability Assessment," Course Notes.

[50] Rice, J. A., *Mathematical Statistics and Data Analysis*, Duxbury Press, Belmont, CA, 1999.

[51] Wertz, J., *Reliability and Productivity Modeling for the Optimization of Separated Spacecraft Interferometers*, Master's thesis, Massachusetts Institute of Technology, 2002.

[52] Meylan, G., Madrid, J. P., and Macchetto, D., "Hubble Space Telescope Science Metrics," *Publications of the Astronomical Society of the Pacific*, Vol. 116, 2004, pp. 790–796.

[53] Joppin, C., *On-orbit servicing for satellite upgrades*, Master's thesis, Massachusetts Institute of Technology, 2004.

[54] Canada-France-Hawaii Telescope, "WIRCam Throughput," Web site, 2007, http://www.cfht.hawaii.edu/Instruments/Imaging/WIRCam/WIRCamThroughput.html.

[55] Space Telescope Science Institute, "Wide Field Camera 3," Web site, 2006, http://www.stsci.edu/hst/wfc3.

[56] Bodegom, E., "Charge-coupled devices," Web site, 2007, http://physics.pdx.edu/ d4eb/ccd/index.htm.

[57] Xin, Z.-J., "Overview of Infrared Detectors: Status and Trends," Presentation.

[58] NASA Johnson Space Center, "Advanced Missions Cost Model," Web site, 2006, http://cost.jsc.nasa.gov/AMCM.html.

[59] "Hubble Space Telescope Servicing Mission 3A: Cost to Taxpayers," Tech. Rep. FS-1999-10-032-GSFC, NASA Goddard Space Flight Center, 1999.

[60] Isakowitz, S. J., *International reference guide to space launch systems*, American Institute of Aeronautics and Astronautics, Reston, VA, 2004.

[61] NASA Johnson Space Center, "International Launch Vehicle Data," Web site, 2006, http://cost.jsc.nasa.gov/ELV_INTL.html.

[62] NASA Johnson Space Center, "NASA Inflation Calculator," Web site, 2007, http://cost.jsc.nasa.gov/inflation/nasa/inflateNASA.html.

[63] NASA Johnson Space Center, "Mission Operations Cost Model," Web site, 2006, http://cost.jsc.nasa.gov/MOCM.html.

[64] Larson, W. J. and Wertz, J. R., *Space Mission Analysis and Design*, Space Technology Library, Microcosm Press, Segundo, CA, 3rd ed., 1999.

[65] Leidecker, H. and Thomas, W., "Notes on the reliability of HST gyros," Tech. rep., NASA Goddard Space Flight Center, 2002.

[66] National Institute of Standards and Technology, "Assessing Product Reliability," Web site, 2006, http://www.itl.nist.gov/div898/handbook/apr/apr.htm.

[67] Rausand, M. and Hyland, A., *System Reliability Theory: Models, Statistical Methods, and Applications*, John Wiley & Sons, Inc, Hoboken, NJ, 2004.

[68] Space Studies Board, *Assessment of Options for Extending the Life of the Hubble Space Telescope*, National Research Council of the National Academies, Washington, DC, 2005.

[69] Wong, H., "Hubble Space Telescope Reliability Assessment, July 2002 Model," Tech. Rep. TOR-2003(2154)-2352, The Aerospace Corporation, 2002.

[70] Lamamy, J., *Methods and Tools for the Formulation, Evaluation and Optimization of Rover Mission Concepts*, Ph.D. thesis, Massachusetts Institute of Technology, 2007.

[71] NASA Johnson Space Center, "Learning Curve Calculator," Web site, 2007, http://cost.jsc.nasa.gov/learn.html.

[72] Chang, I. S., "Investigation of space launch vehicle catastrophic failures," *Journal of Spacecraft and Rockets*, Vol. 33, No. 2, 1996, pp. 198–205.

[73] Allen, B. D., "Historical reliability of U.S. launch vehicles," *AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, 37th*, Salt Lake City, UT, 8-11 July 2001, AIAA 2001-3874.

[74] The Aerospace Corporation, "Space Launch Vehicle Reliability," Web site, 2007, http://www.aero.org/publications/crosslink/winter2001/03.html.