Fall 12-2020

Using Interacting Multiple Model Filters to Indicate Program Risk

Amy Sunshine Smith-Carroll
Old Dominion University, asmithcarroll@verizon.net

USING INTERACTING MULTIPLE MODEL FILTERS TO INDICATE PROGRAM RISK

by

Amy Sunshine Smith-Carroll
B.S. May 1997, Mississippi University for Women
M.S. May 1999, Mississippi State University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ENGINEERING MANAGEMENT

OLD DOMINION UNIVERSITY
December 2020

Approved by:

Andres Sousa-Poza (Director)

Charles Keating (Member)

Adrian Gheorghe (Member)

Victor Gehman (Member)

ABSTRACT

USING INTERACTING MULTIPLE MODEL FILTERS TO INDICATE PROGRAM RISK

Amy Sunshine Smith-Carroll
Old Dominion University, 2020
Director: Dr. Andres Sousa-Poza

Technology development has increased exponentially. Program managers are pushed to accelerate development. There are many resources available to program managers that enable acceleration, such as: additional resources in the form of funding, people and technology. There are also negative impacts to acceleration, such as: inclusion, inexperience program managers, and communication. This research seeks to identify the limit to which a program or project can be accelerated before the program manager begins to accept an unacceptable amount of pre-determined risk.

This research will utilize estimation algorithms used by sensor systems to estimate the current and future state of objects in space. The most common estimation algorithm used is the Kalman filter developed by Kalman (Bar-Shalom, Rong Li, & Kirubarajan, 2001). This research will examine the use of two Kalman filters in for the form of an Interacting Multiple Model (IMM) in order to predict the future state of the program. Traditional multiple model filters use Bayesian technique to adaptively switch between different motion models implemented in the filter structure (USA Patent No. 7030809, 2005). These logic designs rely upon a predefined Markov Switching Matrix (MSM). If the future state approaches a predetermined acceptable level of risk, the MSM will indicate to the program manager that the project has potentially reached a level of unacceptable risk.

This thesis is dedicated to my Mom and Dad. I love you.

ACKNOWLEDGMENTS

There are many people who have contributed to the successful completion of this dissertation. I extend many, many thanks to my committee members for their patience and hours of guidance on my research and editing of this manuscript. The untiring efforts of my major advisor deserves special recognition. I would also like to extend a heartfelt thanks to my mentors, Dr. Hugh McCabe, Dr. Victor Gehman and John E. Gray. These gentleman have forgotten more than I will ever learn. To my family, Brett, Colby and Cameron, I thank you for your patience, support and understanding. I love you all. Finally, to my Mom and Dad, thank you for always being there for me and pushing me to keep going.

NOMENCLATURE

AI	Artificial Intelligence
CA	Constant Acceleration
CV	Constant Velocity
CT	Constant Turn
IMM	Interacting Mulitple Model
MCO	Mar's Climate Orbiter
MM	Multiple Model
MOA	Memorandum of Agreement
MOU	Memorandum of Understanding
MSM	Markov Switching Matrix
SME	Subject Matter Expert
TTO	Technology Partnering Office
t	time
x	Schedule
y	Cost
z	Performance

TABLE OF CONTENTS

Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

CHAPTER 1

INTRODUCTION

Technology development has increased exponentially such that normal acquisition processes are unable to keep pace. Often times during development, new technology, such as sensors, are released by industry and unable to be incorporated into ongoing program development. In order to pace this technology, program managers are pushed to accelerate development.

In an effort to match the pace of technology, program managers are asked to accelerate development and to also be agile. There are numerous methods to accelerated program development and there are negative impacts. The purpose of this research is to evaluate the modified or redefined use of estimation techniques for target tracking to estimate schedule, cost and performance with a predefined risk tolerance.

1.1 PURPOSE

This research will utilize estimation algorithms used in sensor systems to estimate the current and future state of objects in space to estimate future program cost and schedule. The most common estimation algorithm used is the Kalman filter developed by Kalman (Bar-Shalom, Rong Li, & Kirubarajan, 2001). This research will examine the use of two Kalman filters in for the form of an Interacting Multiple Model (IMM) to predict the future state of the program. Traditional multiple model filters use Bayesian technique to adaptively switch between different motion models implemented in the filter structure (USA Patent No. 7030809, 2005). These logic designs rely upon a predefined Markov Switching Matrix (MSM). In this research, the MSM values will be used to represent the amount of risk that a program manager is willing to accept.

1.2 PROBLEM

The problem that will be used in this research is an actual program planned schedule to deliver equipment to sites in a two-year timeframe. The program manager has been asked to accelerate delivery and has provided the planned schedule and cost presented in Section 3. Performance will not be evaluated since the program is focused on delivery and no acceptance testing has occurred to estimate performance at this point.

The model to be used consists of two state models representing planned and actual data for cost, schedule, performance and time. Performance will be set to zero based on the current state of the program. Therefore, the program planned will be represented in the following form:

$$xp_k = \begin{bmatrix} s \\ c \\ p \\ t \end{bmatrix}. \tag{1-1}$$

Program actual data will be represented in the following form:

$$xa_k = \begin{bmatrix} s \\ c \\ p \\ t \end{bmatrix}. \tag{1-2}$$

The program model is dynamically changing over time and reacting to program change(s) (i.e. attempts to accelerate). Planned versus actual program schedule and cost will be evaluated to determine if attempt to accelerate has reached an unacceptable predetermined risk tolerance. The program manager will be able to use this information to determine if methods of acceleration implemented are successful.

1.3 DISSERTATION OVERVIEW

This dissertation is organized as follows:

- Chapter 1 addresses the outline of the research to include the Theoretical Formulations, Purpose, Problem and Model Design.
- Chapter 2 provides an overview of the literature focusing on methods for acceleration of programs, difference between programs and projects.
- Chapter 3 provides a summary of the methods of acceleration and potential negative impacts to acceleration.
- Chapter 4 provides the outline of the Kalman filter.
- Chapter 5 provides the outline for an Interacting Multiple Model.
- Chapter 6 provides details on the research methodology.
- Chapter 7 provides an analysis of results.

CHAPTER 2

BACKGROUND OF THE STUDY

What can be done in order to accelerate a program schedule from its current state and what is the associated risk? It is much harder to accelerate a program that is currently executing than it is for a new start program. New start programs have the advantage of developing a common understanding of the customers' needs/requirement, and a program plan and of, forming teams, and program management approach. Executing programs may begin with an understanding of customer need, but as program evolved customer needs, team membership, and program plans might change due to a variety of uncontrollable events. The following section will provide an overview of the literature available on program acceleration, both methods and potential consequences. Additionally, the following section will provide an overview of estimation theory applied to program management.

2.1 LITERATURE REVIEW

By reviewing the literature associated with accelerated program performance, the common understanding between what consists of a program versus a project as defined by Mumms and Bjeirmi (Munns & Bjeirmi, 1996) is a great place to start. A project is defined as a series of activities to meet an objective while a program or program management is the process of controlling project activities. This work will focus on the program level and evaluate the high-level cost and schedule associated with project activities. Much research has gone into the evaluation of methods to accelerate programs. Nicoletti and Nicolo identified activities that can be performed concurrently and to what extent (Nicoletti & Nicolo, 1998). Roemer et al. evaluated the tradeoff between activity crashing and overlapping in order to accelerate program deliver (Roemer, Ahmadi, & Wang, 2004). Effective communication always positively impacts program acceleration as described by Keyton (Keyton, 2002). Additional resources in the form of personnel and effective group formation can also aid in program acceleration (Wheelan, 2009). Negative impacts can also be associated with acceleration such as those associated with the Mars Climate Orbiter project failure described by Sauser et al. (Sauser, Reilly, & Shenhar, 2009).

This research assumes that one or more of the recommended methods of acceleration has been determined and implemented. The use of the Kalman filter to forecast program schedule, cost and performance has been demonstrated by the research conducted by Bondugula from Texas A&M University (Bondugula, 2009). Additionally, Byung utilized two probabilistic models in the form of a Kalman filter and Bayesian adaptive forecasting method to predict performance estimation (Byung, 2007). This work expands on the work by Bondugula and Byung by evaluating the IMM described in Section 4 to estimate program schedule and cost.

2.2 DIFFERENCES BETWEEN PROJECT AND PROGRAM MANAGEMENT

First, it is important to define the differences between project and program management. Much research has been done to define and explain the differences between a project and the program management associated with it by Munns and Bjeirmi (Munns & Bjeirmi, 1996). A project can be defined as an effort to meet a "specific objective which involves a series of activities and tasks which consume resources (Munns & Bjeirmi, 1996)" or "a complex, non-routine, one-time effort limited by time, budget, resources and performance specifications design to meet customer needs (Attarzadeh, 2008)".

Project success, which is long term, is based on goal, user satisfaction, usability or perceived value. Some reference project success as the golden triangle of time, budget and quality. These factors contribute to project success include clear objective; understandable/concise requirements; customer involvement; and workforce with subject matter expertise, proper planning, and organizational support. Of these factors contributing to success, Attarzadeh and Ow suggest the most important are customer involvement, organizational support, understandable/concise requirements and proper planning. Profitability and competitive advantage are also factors, they are not prevalent within government laboratories but are extremely important to our industry partners. Of these, the most important factor is customer involvement. Without a clear understanding of customer needs and intended use, a project can end up being irrelevant.

"Program management can be defined as the process of controlling the achievement of the project objective" (Munns & Bjeirmi, 1996) or "a set of tools, techniques, and knowledge that, when applied, helps to achieve the three main constraints of scope, cost and time (Attarzadeh, 2008)". Program management can also be defined as "A group of related projects managed in a coordinated way to obtain benefits and control not available from management them individually. Programs may contain elements of work outside the scope of the discrete projects in the program (Weaver, 2010)."

Program management success, which is short-term, is based on resources, organizational support, commitment, and clearly defined tasking that achieves project goals and schedule. Program management can be considered a subset of overall project execution but is not the only factor influencing project success. Many organizations use program management to achieve project goals. Program management becomes the mechanism we can use to accelerate a project but is not the only mechanism. Organization, financial and schedule factors influence the ability to accelerate a project.

Given these definitions, the discussion of accelerating project delivery can be distinct, given the definition of program management. It is understood within the community that program management is critical to project success but not the only factor influencing project success. This is presented to make clear that project and program management success are not mutually exclusive. Munns and Bjeirmi present three factors that cause confusion between project and program management. First is time frame. Project time frame is much longer in that it is not realized initially upon project completion but upon user evaluation. Second is the establishment of clear objectives. Program management success is defined by budget, schedule and quality criteria established at project initiation. Profitability is a project objective, yet budget is the primary program management objective. Many times, objective capability is lost due to budget and time constraints. Lastly, ease of measurement is a factor. Budget and schedule can be measured, but project relevance is qualitative and cannot be clearly measured.

What, then, can be done to accelerate a project? Effective program management techniques offer a means to plan and control a projects development. Brooks' cautions program managers that: "1. Large differences exist between high and low end performers, 2. Development team composition may make all the difference, 3. You must have a written plan, 4. Written specifications are necessary, 5. Vertical division of labor will result in radically simplified communication and improved conceptual integrity, 6. Change is inevitable making change management and planning imperative (Verner, Overmeyer, & McCain, 1999)". Program management in order to be effective must negate all of these cautions presented by Brooks.

Good program management techniques include accurate cost estimation, resource scheduling, communication, user coordination and risk acceptance.

CHAPTER 3

METHODS AND IMPACTS OF ACCELERATING PROGRAMS

Once a program has started, in general, the total life-cycle cost estimate has been generated to accommodate for the resources (manpower, equipment and facilities) required to complete the execution of the project. How do we accelerate a program without funding adjustments? This is a hard problem to solve. Programs that may be more expensive in the near-term may pay for themselves in customer utility and total lifetime cost of the program. There are times when the cost of not accelerating a program may be considered.

3.1 OVERVIEW OF METHODS TO ACCELERATE PROGRAMS

Many have studied methods to accelerate programs such as additional of resources in the form of funding or personnel, resource scheduling, incorporation of new technology, increased communication, clear definition of requirements, acceptance of risk and removing barriers. The following sections will provide additional details on each of these methods.

3.1.1 APPLYING ADDITIONAL RESOURCES

In theory, acceleration equates to shifting everything to the left (Firesmith, 2015). This would include funding. Cost estimation considers full/part time employees required to complete each task within a project, program management support, software development tools, hardware, office/lab space, and test facilities such as ranges. This is always an estimate. It is common for program managers to add an additional 20% to the cost estimate to cover unknowns. Increasing funding allows for more resources in the form of manpower, equipment or facilities to be brought to bear. It is thought that more people and more equipment provide acceleration in delivery. That may provide some benefit within an organization with an appropriately cross-trained workforce. Many organizations have a hierarchical architecture that consists of many layers of management. Hierarchical architectures make sense for work that is linear in nature. There are many challenges with hierarchical architectures such as communication flow, which occurs from top down. Top down communication means “innovation stagnates, engagement suffers, and collaboration is virtually non-existent (Morgan, Forbes, 2015).”

Flat organizations possess more of a streamline processes with, less organizational overhead and management. Less organizational overhead and management structure leads to quicker decision-making processes, which has the potential to save time and money. Additionally, organizational implementation of standardized processes such as consistent documentation and repeatable processes are elements of a good organization that allow for proper configuration control resulting in a better program. Flat organizations present their own challenges in that employees who have been there longer tend to be viewed as senior, cliques’ form that can cause communication and collaboration challenges (Morgan, Forbes, 2015).

3.1.2 RESOURCE SCHEDULING

Using resource scheduling as a technique to speed up project development is usually one of the first techniques implemented. The resources include additional labor, work -hours, equipment, and facilities. Individuals with specific subject matter expertise relevant to the project can bring a wealth of knowledge and expertise to bear on a problem which will in -turn speed up development. Increasing the availability of equipment and laboratory/test facilities also provides an opportunity to complete project activities rapidly.

3.1.3 INCORPORATION OF MATURE TECHNOLOGY

A great way to accelerate development of a project is to take advantage of mature technology. Mature technology can be leveraged both within the project itself and as a contributor the project. For example, CAD tools may be used to generate drawings in which 3-D printers are able to print parts versus actual machining of parts. 3-D printing of parts has the potential to cut cost and schedule demands dramatically. Another example involves the use of a mature technology within a project. Recent work from a university on set-based design has provided the government the ability to develop Program Objective Memorandum (POM) more effectively and efficiently by reducing the number of work hours needed to iterate through the various combinations within the solution space (Singer, Doerry, & Buckley, 2009).

3.1.4 INCREASED COMMUNICATION

Increased communication between organization, program managers, customers, and individual team members can accelerate development. Communication between the customer, program managers and the individual team members, is critical to delivering a project that meets the customer's intended use. Organizational communication of strategic intent provides focus for program managers and individual team members. Organizational goals differ from those of the individual project in that organizations tend to focus on return on investment, customer satisfaction, and development of quality product; therefore, organizational support and commitment is critical to accelerating a project. A common belief is that co-location increases communication and thus accelerates development by reducing the number of meetings, phone calls, and reviews.

3.1.5 CLEAR DEFINITION OF THE REQUIREMENT

The requirement via customer input is critical in defining the project. The lack of customer input accounted for fifty percent of failed projects according to a study by Verner (Verner, Overmeyer, & McCain, 1999). Early and often customer input generates confidence in that the project is going to deliver needed capability. Requirements must be clear, concise and attainable. Projects not properly planned, possessing vague requirements, or having no clear deliverable are at a higher risk of project failure over those that do possess these characteristics.

3.1.6 ACCEPTANCE OF RISK

Organizations accelerate projects by accepting more scheduled risk in areas such as certification, testing, verification, and validation. There are many programs in which such risks would not be acceptable but feasible. There are times when risk can be waived with minimal impact. Consider a manned air platform. Manned air platforms have very specific requirements for testing and certification based on the dangers associated with loss of life. Currently, unmanned air platforms must follow the same requirements as manned air platforms. Since there is no risk related to loss of life with an unmanned air platform, those requirements for testing and certification could be waived, which would save the program both time and money.

3.1.7 REMOVING BARRIERS

Finally, removing the barriers to acceleration is critical within project development. Many policies and processes apply to one project but not to another. For example, unmanned systems should not have to go through the same flight test and evaluation as a manned system. Many safety components just do not apply to an unmanned system. By removing these unnecessary testing requirements, an unmanned systems project can be accelerated while providing the additional benefit of saving cost.

Industry developed proprietary technology presents a challenge in that any modifications required to update that technology in the future require the technology owner to make the modification. This affects both cost and schedule in that proprietary technology assumes modification at additional cost and schedule increase. By developing projects with open-systems standards, reliance on the original developer to make modifications is reduced, hence potentially reducing cost and time to completion.

In the 1970s, universities focused more on the process of discovery and less on the process of transition of technology to industry. It was not until 1980, when Congress passed the Bayh-Dole Act, that universities shifted focus to the transition of Science and Technology (S&T) to Industry and Government organizations (Pub. L. 96-517, 1980). To address many of the disadvantages associated with transition, many universities established Technology Transition Offices (TTOs). These TTOs provided resources for external partnerships and innovation opportunities. The significance of this act lies in the fact that before the Bayh-Dole Act, federal research funding contracts and grants obligated inventors to assign inventions they made by using federal funding to the federal government. After enactment, universities, small businesses, or non-profit institutions are permitted to claim ownership of an invention. The purpose of a TTO is to establish agreements between academia, industry and the government to foster exchange of information and protect that information. These agreements can be in the form of Memorandum of Agreements (MOAs), Memorandum of Understanding (MOUs), and Collaborative Research and Development Agreements (CRADAs).

3.2 NEGATIVE IMPACTS TO ACCELERATING

There are many positives associated with accelerating projects, but there are also many negatives. As always, delivery of a product to the customer faster as anticipated can always positively impact acceleration. The negatives tend to have the most effects on an organization's lead times and product, such as longer than expected experimentation timelines and additional rework due to unexpected failures.

3.2.1 INCLUSION

F. P. Brooks' book, *The Mythical Man-Month*, discusses the idea that addition of manpower does not accelerate until a time lag has passed wherein training yields additional productivity. Brooks gives three explanations as to why the "Brooks Law" (Verner, Overmeyer, & McCain, 1999) is applicable. With the addition on new manpower comes the need for additional training required to bring those additional employees up to speed in order to be a productive member of the team. If the additional manpower is in the form of employee/s with subject matter expertise (SME) specific to the project, then employee contribution occurs quickly. This does not consider inclusion. Inclusion takes additional time, as the employee must become a trusted contributor to the team. SMEs tend to dominate and are sometimes slow to be adopted into the team. If the additional manpower is not a SME but a junior employee, inclusion may happen faster since the junior employee tends to listen and learn versus dominate. Contribution to the project takes longer since the junior employee must be trained and mentored in order to be brought up to speed on project development.

3.2.2 DIVISION OF WORK

There are times when division of work/tasking changes. Some work may not be able to be split for others to support. There are instances in which a task cannot be performed in parallel or divided among team members. One example can be demonstrated by the failure of NASA's Mars Climate Orbiter (MCO) in which metric units were used in coding of the ground software. Since the program was innovative in that there was much uncertainty and complexity involved in its development, management decided to reuse as many components as possible from previous and ongoing programs. This allowed for the reduction in time, cost and uncertainty. Integration remained an issue. Employees working on the integration of the navigation system also worked on another project. This led to confusion amongst the engineers working on the MCO and ultimately led to its failure. This failure could have been avoided had the group members or subject matter experts communicated amongst themselves on the different projects. Additionally, due to the acceleration, many of the critical informal and formal reviews were ignored (Sauser, Reilly, & Shenhar, 2009).

3.2.3 COMMUNICATION

There is an increase in communication that must occur due to the addition of manpower. Accelerating a project, more times than not, decreases communication in that decisions are made very rapidly with little input from developers or customers. In order to accelerate, an organization must streamline the decision process in order to remove barriers. This can also be observed from the failure of NASA's MCO (Sauser, Reilly, & Shenhar, 2009). Lack of communication created confusion and frustration amongst the team members. Subject matter expertise was often ignored (Report to the Presidential Commission on the Space Shuttle

Challenger Accident, 1986). There was little knowledge regarding the actual innovations being added to the program and the integration challenges that were going to occur due to reuse of components.

Communication with customers is critical to achieving a product with the desired capability. The failure of NASA's MCO demonstrated how rapid development and lack of customer communication are critical to mission success (Sauser, Reilly, & Shenhar, 2009). If the customer's needs are not fully established at the initiation of the project, then chances are the product will not meet those needs. Often, capability within a project is sacrificed in order to accelerate. Continued communication allows for the potential to develop a plan to deliver limited capability in order to accelerate delivery.

3.2.4 SOCIAL FACTORS

There are also social factors related to the addition of manpower. Tuckman suggests additions to the team cause the storming, norming and conforming cycle to repeat itself (Tuckerman, 1993). This takes time for the team to become a cohesive productive team again. Keyton suggests that the most effective teams are comprised of at least "three or more members that interact with each other to perform a number of tasks and achieve a set of common goals" (Keyton, 2002). Team composition and size have been the source of many studies. Wheelan asserts that groups with approximately eight members are the most productive (Wheelan, 2009).

Verner and his colleagues studied twenty large software projects twenty-five years after the publication of Brooks' Law and deduced that many of Brook's Laws still hold (Verner, Overmeyer, & McCain, 1999). Manpower retention and addition play a large role in the timely completion of projects. Still, there are other effects presented by the addition of manpower, which include low morale and inconsistent continuity of staff either via reassignment or via turnover. Hsai, Hsu and Kung attempted to revisit Brooks' Law. They deduced that time is a critical factor in adding manpower. Just the addition of manpower alone makes the project costlier but does not always make the project late. If tasks are done sequentially, then the additional manpower will not speed up development of a project. However, what if the additional manpower was brought in early in the development process with experience and tasks could be conducted in parallel versus sequentially. The optimal project timeline for bringing on additional manpower immediately is one-third and halfway through the project timeline (Hsai, Hsu, & Kung). Increasing daily work schedules without augmenting the workforce with additional resources in the form of people can cause employee burnout and decrease in employee morale. Crawford suggests that with acceleration comes mistakes as employees are tasked with simplification or elimination of tasks. Additionally, employees tend to "ignore, postpose or mishandle" uncertainty based on aggressive development schedules (Swink, 2003).

3.2.5 TEST AND EVALUATION

Many times, in order to accelerate an effort, test and evaluation is ignored. Many assume that component reuse eliminates the need for continued testing and evaluation but that is an incorrect assumption in many cases. Component reuse can accelerate a project in terms of individual components but does not address integration issues that come from design of a new system. Often, component reuse requires additional design, development and testing in order to ensure proper functionality exists. In the case of MCO, this lack of integration, verification and validation was due to cost constraints and resulted in the ultimate failure of the project (Sauser, Reilly, & Shenhar, 2009). Among other hurtles to acceleration are requirements for certification that could be waived which would save the program both cost and schedule. Proprietary technology presents a challenge in that modifications required to update that technology in the future require the technology owner to make the modification. This affects both cost and schedule in that proprietary technology assumes modification at additional cost and schedule increase.

3.2.6 INEXPERIENCED PROGRAM MANAGERS

Inexperienced program managers with little to no experience in project planning, timeline development, project integration, communication of priorities and tasks present problems to projects attempting to accelerate or even follow a normal timeline. Inexperienced program managers lack the control necessary to accelerate and tend to micro-manage their workforce in order to meet schedule deadlines. Pitch, Lock and De Meyer's (Pitch, Loch, & De Meyer, 2002) coping strategy model identifies two coping strategies: learning and instructionalist. The learning strategy is based on the team's response to variation and the program manager's ability to plan for variability in target execution.

This approach leads to the increase in testing and implementation of training of engineers to address future uncertainty (Sauser, Reilly, & Shenhar, 2009). The instructionalist strategy maintains that the project has little uncertainty and tends to follow the incremental design approach with no true modularity based on previous efforts. Crawford suggests that program managers are "less able to predict or control the effects of aggressive time goals on various steps of a highly complex project (Crawford, 2004)".

3.3 SO WHY ACCELERATE?

The pace of technology development is exponential (Kurzweil, 2001). Current acquisition processes do not always allow for this rapid leap in technology. In order to pace competitors and satisfy stakeholder needs, accelerated deployment of new technology should be considered. In today's fiscally constrained environment, seeking new ways to increase innovation allows us to keep up with global trends. Incorporation of academic or industrial developed S&T into government-led programs is designed to leverage knowledge from all sources, accelerate development, and reduce cost and risk associated with development. Some advantages include reduced internal investment and reduced long-term development cycle.

Karagozoglu and Brown associate acceleration of a project to workforce motivation such that the workforce has a "sensor of priority such that they give greater attention to the project activities and make more efficacious use of project resources (Karagozolgu.N. & Brown, 1993)". Acceleration also plays a role in the quality of personnel supporting the project, thus increasing the importance of project leadership and management approach.

With the development of new technology, the ability to share information across organizations, programs, projects, and so on enables the workforce to rapidly acquire and assess information with little delay. The need for multiple meetings and/or reviews has diminished with the availability to communicate information quickly via email, share-drive, drop box, and a multitude of other mechanisms. With the ease of access to data or information, informed decisions can be made rapidly with very little impact to project schedule. People, availability of information availability and development of new technology enable projects to deliver needed solutions into the hands of the customer as rapidly as possible.

CHAPTER 4

MULTIPLE MODEL FILTER

Tracking filters used to filter out the noise associated with a sensor are widely available with several different types to choose from. Two such classes of filters are the single model filter and the multiple model (MM) filter. Some examples of a single model tracking filter are alpha-beta, alpha-beta-gamma, and Kalman filters. When two or more of these types of single model tracking filters are run in parallel, a multiple model filter is formed. A multiple model filter provides improved performance for tracking maneuvering objects over a single model filter. MM filters consist of two or more filters that combine their estimates in some fashion to achieve an improved estimate.

Work performed by Bondugula established the fact that Kalman filters have the ability to forecast program schedule, cost and performance (Bondugula, 2009). The equations governing the Kalman filter can be found in Appendix A. When utilizing the Kalman filter, the user must make assumptions regarding the dynamic motion associated with the program.

For example, a sensor system assumes a linear motion for objects in space moving in a straight line at a constant velocity (CV). A single Kalman filter utilizing a CV motion model will present a lag when acceleration occurs. Another model that can be used is an acceleration model that accounts for exponential rate of change. The constant turn model would be useful when an object maneuver consists of approximately constant speed and turn rate. Other models such as polynomial, Singer acceleration, and mean-adaptive acceleration can be found in Li et al. (Li, 2003).

Better estimation allows for an increase in the number of maneuvering and non-maneuvering objects that can be tracked and a reduction in reaction time (or lag). Multiple Model tracking filters improve tracking of both maneuvering and non-maneuvering objects. What distinguishes a superior Multiple Model tracking filter design from a poor filter design is the speed with which the switching logic detects and then responds to an object maneuver by reshuffling the weights to match the new object dynamic configuration. Most Multiple Model (MM) filter designs incorporate a Markov Switching Matrix (MSM) as part of their switching logic. This matrix, whose values are selected in a generally ad hoc manner, has a significant impact on the response time of the switching logic to a sudden object maneuver.

No "optimum" method exists for selecting values with which to populate this matrix. A set of values that may provide a "good" tracking performance against a specific object type may not yield a "good" performance against a different object type. Since one cannot know in advance what object type is going to be encountered in each scenario, the filter designer is faced with a design dilemma. Despite this, the MM filter structure has won wide acceptance within the academic tracking community (USA Patent No. 7030809, 2005).

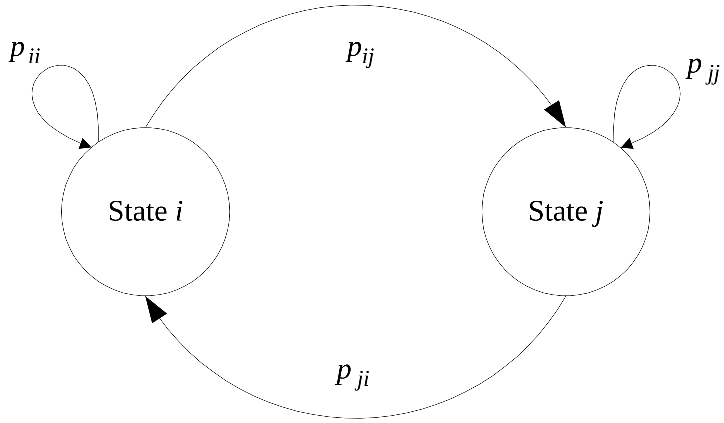
Multiple Model (MM) filters often use a Markov Switching Matrix (MSM) in their switching logic design, as does the Interacting Multiple Model (IMM). A MSM is an NxN matrix, where N is the number of models in the filter bank that consists of switching probabilities, $p_{ij}\ for\ i, j = 1 \ldots N$. The probabilities, p_{ij}, in the MSM have the following properties:

- The diagonal element, p_{ii} , represents the conditional probability that the system state will remain in state *i* after the next transition.

- The off diagonal elements p_{ij} represent the conditional probability that the system will transition to state *j* after the next transition given that it is currently in state *i*.
- The sum of elements across each row of the MSM must be unity.

These properties are highlighted in **Figure 4-1**.

$$p_{ij} = \Pr\{\text{System will switch to state } j \text{ given that it is currently in state } i\}$$



$$\Pi \equiv \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N1} & p_{N1} & \cdots & p_{NN} \end{bmatrix}$$

N = Number of models

Figure 4-1. Switching Probabilities

There is no optimum way to pick the values for the MSM. Different values chosen for the MSM will produce different results. Although it is generally agreed that the diagonal elements in the MSM should be "close" to unity, there are no other constraints that can be used to pick the elements for the MSM in some "optimum" manner. [sun1] No one set of values will work well for all trajectories; what works well for one scenario may work poorly for another. Even small changes in p_{ij} can affect the results of the filter. Each designer must choose their own values in some ad hoc manner. When a set of values has been selected, numerous computer simulations are run and the results compared with runs from other combinations of values.

The MSM has a significant impact on how rapidly the switching mechanism detects and then responds to a rapid change in program schedule or cost. What distinguishes a superior filter design from a poor filter design is the speed with which the switching logic detects and then responds to the schedule or cost change by reshuffling the weights to match the new schedule or cost estimate. MM filters have won wide acceptance within the target tracking community and system developers in other fields. Blair presented the interacting multiple bias model filter system for tracking. This system incorporated Markovian switching coefficients for its logic (USA Patent No. 5325098, 1994). In a later patent, multiple Kalman filters feed a model probability update circuit (USA Patent No. 5214433, 1993). The Markov model transition or switching probability function values provide the probability of jumping or changing from models at time K-1 to model t at time K. The values of the model transition probabilities determined as part of the overall system design are analogously to the choice of values for the initial values of the predetermined model parameters.

In general, the Kalman filter is an optimal state estimator for single mode systems provided that an exact motion model for the object dynamics is available. The IMM algorithm was designed to allow increased accuracy while tracking a maneuvering object. The IMM algorithm allows two or more single mode system filters to run in parallel (i.e. CV motion models). The IMM will be described in more detail in the following sections.

4.1 CONSTANT VELOCITY (CV) MODEL

The state vector for the Constant Velocity (CV) Model is given by:

$$x_k = \begin{bmatrix} s \\ \dot{s} \\ c \\ \dot{c} \\ p \\ \dot{p} \end{bmatrix}. \tag{4-1}$$

The state equation that describes the CV model is given by

$$X_{k+1} = \emptyset_k X_k + w_k \tag{4-2}$$

and the measurement equation is given by

$$Z_k = HX_k + v_k. \tag{4-3}$$

The state transition matrix for the CV model is defined as

$$\emptyset_k = \begin{pmatrix} A & B & B \\ B & A & B \\ B & B & A \end{pmatrix} \tag{4-4}$$

where

$$A = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \tag{4-5}$$

T = time interval and

$$B = 0_{2x2}. \tag{4-6}$$

The (6x1) process noise vector, w_k, has a block diagonal covariance matrix given by Q_{CV}.

H is defined as the standard measurement matrix,

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{4-7}$$

The CV models are initialized using the initial measurements Z_1 and Z_2, and are stored in X_1 in the following manner:

$$Z_1 = \begin{bmatrix} s_1 \\ c_1 \\ p_1 \end{bmatrix} \; and \; Z_2 = \begin{bmatrix} s_2 \\ c_2 \\ p_2 \end{bmatrix} \tag{4-8}$$

$$X_1 = \begin{bmatrix} s_2 \\ \dfrac{s_2 - s_1}{T} \\ c_2 \\ \dfrac{c_2 - c_1}{T} \\ p_2 \\ \dfrac{p_2 - p}{T} \end{bmatrix} \tag{4-9}$$

and the error covariance is initialized to the following

$$P_1 = diag\,[A \quad A], where\ A = \begin{bmatrix} .0625 & .0625 \\ .0625 & .1250 \end{bmatrix}. \tag{4-10}$$

4.2 INTERACTING MULTIPLE MODEL

In general, the Kalman filter is an optimal state estimator for single model systems provided that an exact motion model for the object dynamics is available. Many have tried to broaden the Kalman filter to provide optimal state estimates for multiple model systems. The IMM algorithm was designed to allow increased accuracy while tracking a maneuvering object. The IMM algorithm allows two or more filters to run in parallel. Typically, constant velocity (CV), constant acceleration (CA) and constant turning rate (CTR) filters are used in conjunction with an IMM algorithm. The IMM algorithm using two models is shown in **Figure 4-2**. In this study, two Kalman filters are employed, using two CV motion models. One CV motion model employs a low process noise, and the second employs a high process noise. This change in process noise will allow the IMM to estimate the future state of the program with less lag or error during a rapid change or maneuver.

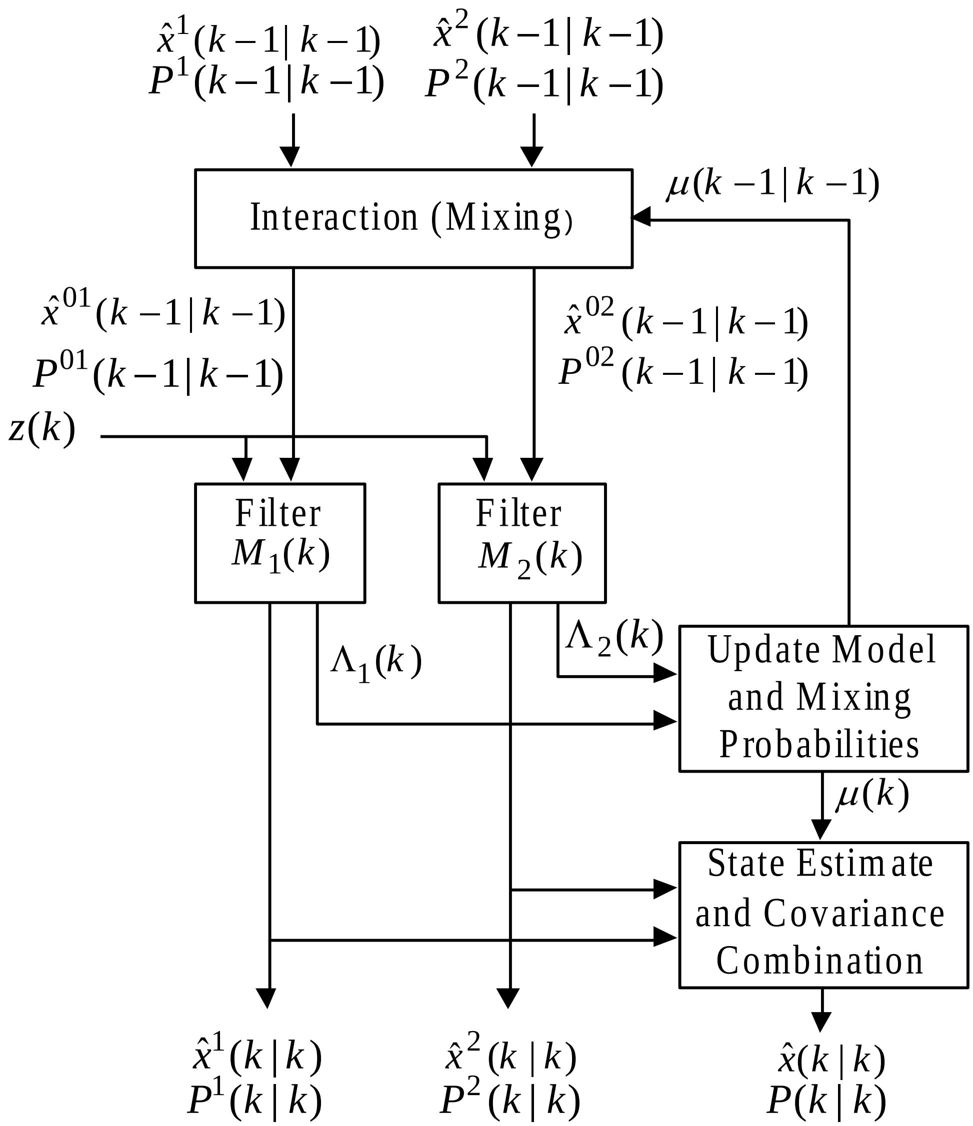


Figure 4-2. Interacting Multiple Model

The equations defining the IMM algorithm for tracking with N dynamic motion models are outlined in the following five steps:

Step 1: Mixing of State Estimates

The filtering process starts with prior state estimates $X^{j}_{k-1|k-1}$ state error covariances $P^{j}_{k-1|k-1}$, and the associated probabilities μ^{j}_{k-1} for each model. The mixed state estimate for M^{j}_{k}, $X^{0j}_{k-1|k-1}$, is computed as

$$X^{0j}_{k-1|k-1} = \sum_{i=1}^{N} X^{i}_{k-1|k-1}\mu^{i|j}_{k-1|k-1} \tag{4-11}$$

where

$$\mu^{i|j}_{k-1|k-1} = \frac{1}{c_j} p_{ij}\mu^{i}_{k-1}\, with\; \overline{c_j} = \textstyle\sum_{i=1}^{N} p_{ij}\mu^{i}_{k-1} \tag{4-12}$$

and p_{ij} is the probability of switching to mode j given that the system is in mode i. Note that the probabilities, p_{ij}, are what constitute the elements of the MSM, Π. In this study, the MSM used in the IMM is comprised of the following values,

$$\Pi = \begin{bmatrix} .95 & .05 \\ .05 & .95 \end{bmatrix}. \tag{4-13}$$

The mixed covariance for M^{j}_{k}, $P^{0j}_{k-1|k-1}$ is computed as

$$P^{0j}_{k-1|k-1} = \textstyle\sum_{i=1}^{N} \mu^{i|j}_{k-1|k-1} \left(P^{i}_{k-1|k-1} + \left(X^{i}_{k-1|k-1} - X^{0j}_{k-1|k-1}\right)\left(X^{i}_{k-1|k-1} - X^{0j}_{k-1|k-1}\right)^{T}\right). \tag{4-14}$$

Step 2: Model-Conditioned Updates

The conventional Kalman filter equations provide the model-conditioned updates.

Step 3: Model Likelihood Computations

The likelihood function for model M_k^j , Λ_k^j is computed with $\bar{Z}_k^j$, $S_k^j$and the assumption of Gaussian statistics. It is given by

$$\Lambda_k^j = \frac{1}{\sqrt{|2\pi S_k^j|}} exp\left[-0.5\left(\bar{Z}_k^j\right)^T\left(S_k^j\right)^{-1}\bar{Z}_k^j\right] \quad (4\text{-}15)$$

A positive lower bound of 10^{-6} is imposed on Λ_k^j to provide numerical stability in the computer program.

Step 4: Model Probabilities Update

The model probabilities, μ_k^j, are updated as

$$\mu_k^j = \frac{1}{c}\Lambda_k^j \; with \; c = \sum_{i-1}^{N}\Lambda_k^j c_i. \quad (4\text{-}16)$$

Step 5: Combination of State Estimates

The state estimate and error covariance for the IMM algorithm output, $X_{k|k}$ and $P_{k|k}$, respectively, are obtained from a probabilistic sum of the individual filter outputs and are given by

$$X_{k|k} = \sum_{i=1}^{N} X_{k|k}^i \mu_k^i \quad (4\text{-}17)$$

$$P_{k|k} = \sum_{i=1}^{N} \mu_k^i \left(P_{k|k}^i + \left(X_{k|k}^i - X_{k|k}\right)\left(X_{k|k}^i - X_{k|k}\right)^T\right). \quad (4\text{-}18)$$

CHAPTER 5

RESEARCH METHODOLOGY

A simulation research design will be used to develop insights about the behavior of cost, schedule and performance on program acceleration. This research seeks to identify the limit to which a program or project can be accelerated before the program manager begins to accept an unacceptable amount of pre-determined risk. A deduction process will be used to build the model, while an induction process will be used to analyze the results. As a positivist/empiricist, this research will seek to understand real world processes such that controls can be put in place to understand risk associated with acceleration. The primary difference in this research and research identified in Section 4.0 is the use of the Interacting Multiple Model (IMM) to predict future schedule and cost and the Markov Switching Matrix (MSM) to evaluate predetermined risk threshold. An assumption about risk tolerance will be made. Risk tolerance values will consist of 5, 10, 15, 20, 25 and 50 percent. The IMM will be used to estimate or forecast cost, schedule, and program performance.

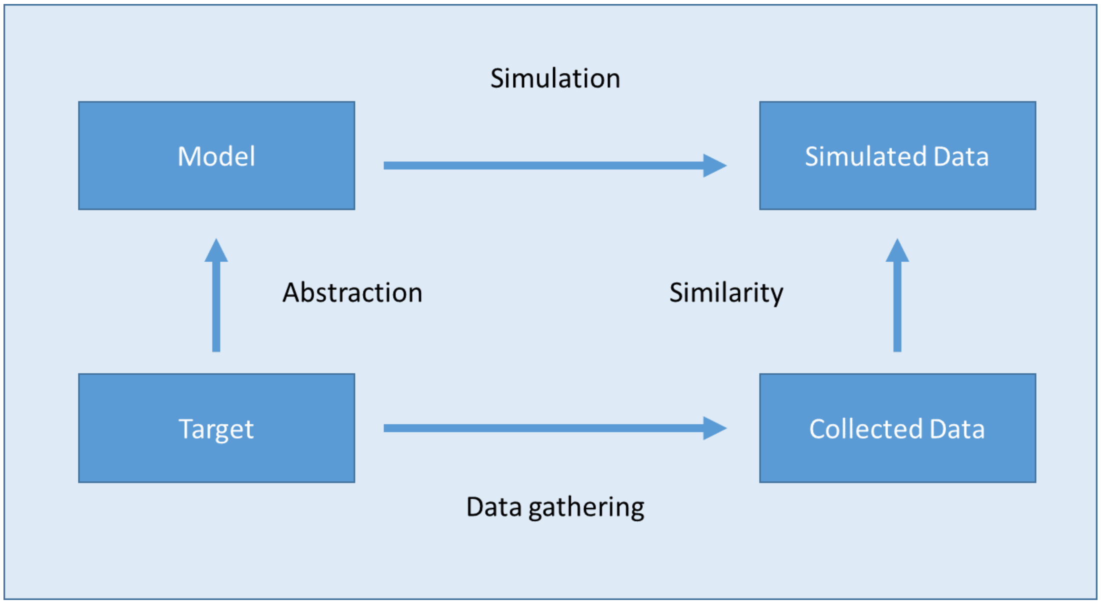


Figure 5-1. Logic of Statistical Modeling as a Method

Figure 5-1 articulates the simulation research design that will be used in the quantitative study. A description of the research methodology will be discussed below.

"Simulation means driving a model of a system with suitable inputs and observing the corresponding outputs". (Bratley, Fox, & Schrage, 1987) As described in **Figure 5-1**, researchers develop a model on presumed processes. The model might exist in the form of a computer program or statistical equation. The model is run, and its behavior is measured. The simulated data can be used for explanation or prediction (Gilbert, Chapter 2: Simulation as a Method, 2005).

Axelrod describes seven different types of simulation in his work (Axelrod, 2005). Among these are prediction, training, entertainment, education, proof, history and theory discovery. Of these seven types, prediction is the simulation type aligned to this research. Prediction is based on a model composed of structure and rules that govern that structure and produce an output (Dooley, 2002). By comparing the output of different structures and governing rules, researchers can infer what might happen in a real situation. Validity of the result is based on the validity of the model. This is a common approach for large organizations, and it is very difficult to model large scale change and understand its implications. Researchers look to predict what will result based on change in order to make recommendations on the value of the change.

Simulation in which a validated model can be used to assess the performance of a task is referred to as performance simulation (Dooley, 2002). This can be used for efforts such as diagnosis and decision-making. Uncertainty and randomness are evident within any organization and are inherent in any system. Simulation allows researchers to take into account uncertainty in the decision-making process by using Monte Carlo simulation. Monte Carlo simulation consists of hundreds or thousands of trials in which each trial samples from the distribution of the variable specified. The composite answer is the aggregate and is described by a distribution of possible outcomes (Dooley, 2002).

Dooley from Arizona State University argues that “computer simulation is growing in popularity as a methodological approach for organizational researchers (Dooley, 2002).” He goes on to argue that simulation-based research allows researchers to investigate the future and ask “what if” questions. Typically, research focuses on historical perspectives, gathering data based on historical events to address questions such as what happened and why. Dooley presents three main schools of simulation practice (see **Table 5-1**).

Simulation Type	Description
Discrete Event Simulation	Modeling of an organization over time according to the availability of resources and event triggers
System Dynamics	Identifying the key “state” variables that define the behavior of the systems and then relate those variables to one another through coupled, differential equations
Agent-based Simulation	Involving agents that attempt to maximize their utility functions by interacting with other agents and resources; behavior is determined by embedded schema that is both interpretive and action-oriented in nature

Table 5-1. Three Schools of Simulation Practice

Discrete event simulation models are best used when the organization can be adequately characterized by variables and corresponding states (Dooley, 2002). It is not appropriate when state variables interact with one another and change continuously. Discrete event simulations describe systems that are discrete, stochastic and dynamic (Law & Kelton, 1982). Law and Kelton characterize discrete event simulation using **Figure 5-2** below.

- Entities: Objects that comprise the system
- System State: state variables that describe a system at a given moment
- Simulation Clock: denoting the passage of simulated time
- Event list: list specifying the events to occur in the future and time at which they will occur
- Statistical Counters: for collecting data during the simulation run, to record history, to be analyzed later
- Initialization Routine: some means to prepare the model for an experimental run
- Timing Routine: subroutine that manages the event list
- Event Routine: subroutine for each different type of event
- Report Generator: reports the aggregate results as obtains from the statistical counters
- Main Program: program that coordinates activity between all the various other elements of the simulation system

Figure 5-2. Discrete Event Simulation

System dynamics simulation or continuous simulation is best used when there are many inter-related variables in question. System dynamics is considered a “top-down” approach in which extensive knowledge about the system and system interactions are required. This approach became popular in the 1950 and later in the 1960 in works by Jay Forrester (Forrester, 1961) and P. Senge (Senge, 1990) along with cybernetics and the desire to use systems theory in the social domain. Forrester defines systems dynamics as "the study of the information-feedback characteristics of industrial activity to show how organization structure, amplification, and time delays interact to influence the success of enterprises" (Forrester, 1961). It treats the interaction between the flows of information, money, orders, materials, personnel and capital equipment in a company, and industry or a national economy. It is a quantitative and experimental approach for relating organizational structure and corporate policy to industrial growth and stability (Forrester, 1961).

In systems dynamic simulations, the variables need not be specific entities or states. Variables do not necessarily have to be consistent in the way they are chosen. Once variables are defined, relationships must be defined to characterize the relationship between variables. State variables are often referred to as sinks, and relationships between sinks are often referred to as flows. Flows are defined as the first derivative of the state variable, hence defining the rate of change between one state variables on another.

In 1997, Sterman et al. documented their work on organizational improvement on the Analog Devices Company (Sterman, Repenning, & Kofman, 1997). The company, Analog Devices, was going through a total quality management change process. Sterman presented the first case representing what happened to Analog Device and its successful waste reduction program in manufacturing. The success in reducing waste in the manufacture of products led to excess capacity for the company. This forced the company to lay off workers and eventually do away with the waste reduction program.

Agent-based simulation is best used when the system is modelled as a collection of agents operated via schema in which they interpret the world and interactions with others. Agent-based simulations focus on learning and adaptation. This is a "bottom-up" approach in which the variables/agents and their connectivity or interactions are known without knowledge of larger scale aggregate behavior. Agent-based simulation stems from artificial intelligence (AI).

System dynamics and discrete event simulation differ from agent-based simulation in that agent-based simulation focuses on the collective behavior of an organization versus independent variables. Hence behavior is produced by parallel and simultaneous actions of many variables versus one variable. These types of systems are referred to as self-organizing (Dooley, 2002). Self-organizing systems can lead to emergent behavior that has not yet been witnessed.

There are two issues one must consider when developing agent-based simulation. The first is the fact that it is difficult to evaluate structural and behavior changes to agent-based models due to underlying emergent behavior of variables. The second issue is that the researcher must decide whether to favor model complexity or model validity. By model complexity, it is meant that as the model become more complex, it is less understandable and likewise, less valid (Dooley, 2002).

Simulation research is in its infancy compared to most other research methods. Computers were not invented until the 1940s/1950s, and access to computers for research purposes did not occur until the late 1960s. Simulation research has its roots in organizational research. In the 1960s, Cyert and March simulated firm behavior (Harrison, The Concept of Simulation in Organizational Research).

Some of the first computer simulation was performed under the Manhattan project. Gilbert and Troitzsch also forged the path forward for simulation research in social science (Gilbert & Troitzsch, Simulation for the social scientist, 2005). Gilbert and Troitzsch argue that simulation provides value as a tool for formalizing theory in social sciences. Computer simulation provides an advantage over traditional math models for research interested in processes and mechanics rather than association between variables (Gilbert & Troitzsch, Simulation for the social scientist, 2005).

Kevin Dooley summarized the three simulation methods in his book Companion to Organizations published in 2002 (Dooley, 2002). Dooley concludes that simulation enables researchers to look to the future versus evaluating the past. Simulation also gives the researcher the opportunity to make improvements to performance in a laboratory environment. Of the three simulation methods, discrete event simulation is the most common and the organization is best represented as a machine with uncertainty in the form of random variables. System dynamics is best used for specific purposes versus generic problems. System dynamic models that are more abstract in nature rarely provide value to the researcher. Agent-based models are best used to answer questions organizational researchers have. This field is in its infancy and a learning curve exists. Dooley suggests that attention must be paid to alignment of theory and model; testing of code; validation of model and results; rigorous experimental design; and appropriate and rigorous statistical analysis (Dooley, 2002).

Rose, Spinks and Canhoto describe the strengths and weaknesses of simulation research in their book. The key strength is "its ability to support investigation of phenomena that are hard to research by more conventional means". (Rose, Spinks, & Canhoto, Management Research: Applying the Principles, 2015) According to Davis "the ability to show outcomes of interacting processes over time or interaction of processes where empirical data is limited". (Davis, Eisenhardt, & Bingham, 2007) An example of is correlation studies. Some of the challenges with simulation research consist of model misrepresentation, errors in developing the computer program, and generalization.

What is proposed in this research is use of discrete event simulation. The basic simulation research steps outlined by Gilbert and Troitzsch will be followed and are outlined in **Figure 5-3** (Gilbert & Troitzsch, Simulation for the social scientist, 2005)).

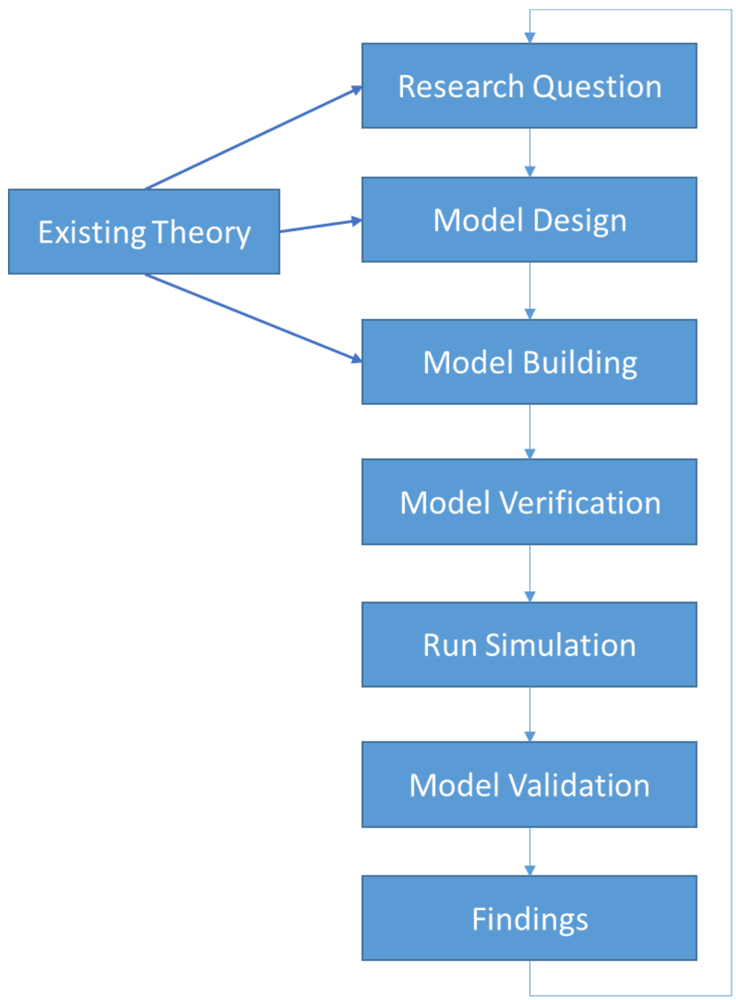


Figure 5-3. Steps in a Simulation-Based Study

5.1 RESEARCH QUESTION

This research seeks to answer the questions:

a) Can the IMM predict future program cost, schedule and performance?

b) Can the Markov Switching Matrix (MSM) within an Interacting Multiple Model (IMM) predict program risk using upper and lower bounds (see **Figure 5-4**)?

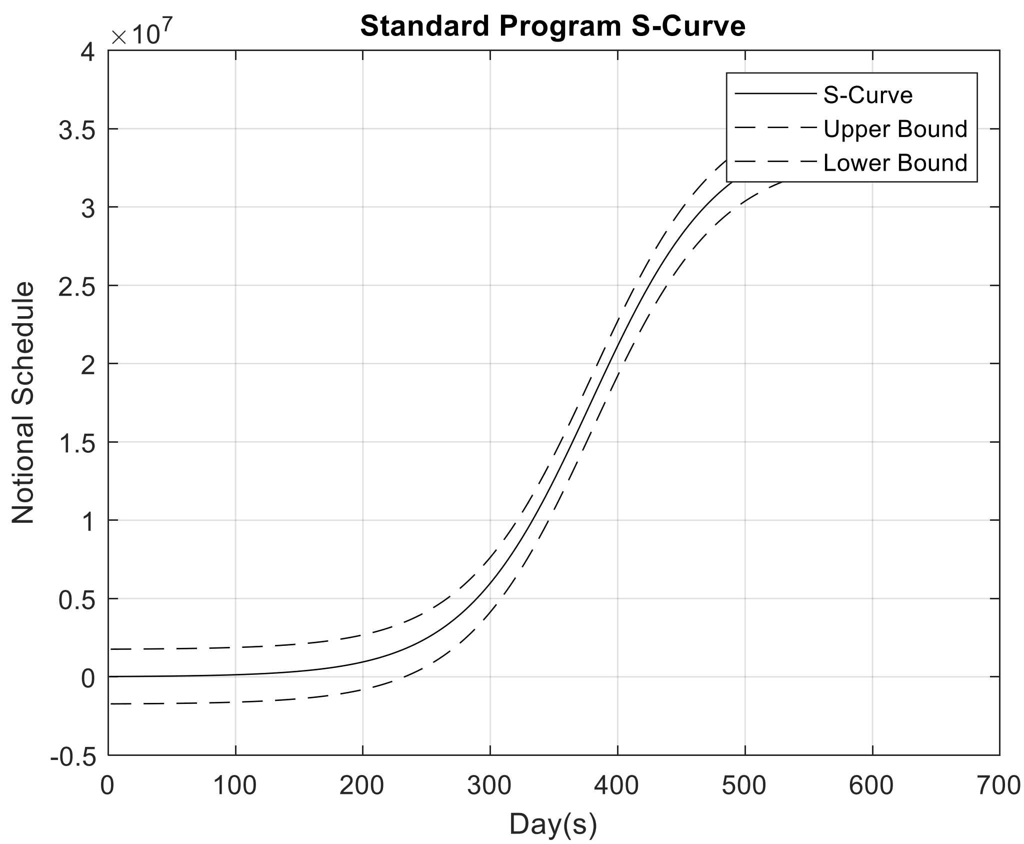


Figure 5-4. Risk Tolerance

In this research proposal, various acceleration parameters and their potential negative impacts have been outlined. It will be assumed that the program manager has determined a method of acceleration and the amount of acceptable risk using a range of risk values (5%, 10%, 15%, 20%, 25% and 50%). The MSM will be used to represent the predetermined risk. Various risk values will be evaluated to determine whether the CV models switch when an unacceptable amount of risk has been reached.

5.2 MODEL DESIGN

The model design will be based on a Monte Carlo simulation model. MATLAB is a software tool developed by MathWorks© for iterative analysis and design processes. This is a desktop software tool used by many scientists and engineers to run Monte Carlo simulations or simulations that require multiple iterations. Kalman filter process noise will be used as a mechanism for inserting randomness into the simulation. This randomness will be representative of the acceleration methods chosen by the program manager. An upper and lower bound will be hypothesized based on the risk assumptions in which acceleration parameters and negative impacts cause the project to assume too much risk (see **Figure 5-4**).

5.2.1 BASIC ASSUMPTIONS

The object is defined as the program model as represented in the "real world". The object model to be used is a program development model and program development lessons learned.

The project estimation model will be developed based on the assumed method of project acceleration implemented by the program manager. The model that will be used for this research will be the Interacting Multiple Model (IMM) (Bar-Shalom, Rong Li, & Kirubarajan, 2001). The IMM allows for predicting the future state of the program given the current estimated state and sensitivity analysis of acceleration parameters to development model Figures of Merit (FOMs). This sensitivity analysis will be the focus of future work.

The planned program schedule is provided in **Figure 5-5**. The planned program schedule shows the number of systems to be delivered over two years. The program manager has been asked to accelerate delivery. In order to accelerate delivery, the program manager has utilized one or more of the acceleration methods identified in 3.1. Performance of the program will be set to zero and will not be evaluated since the program is focused on delivery and no acceptance testing has occurred to estimate performance at this point.

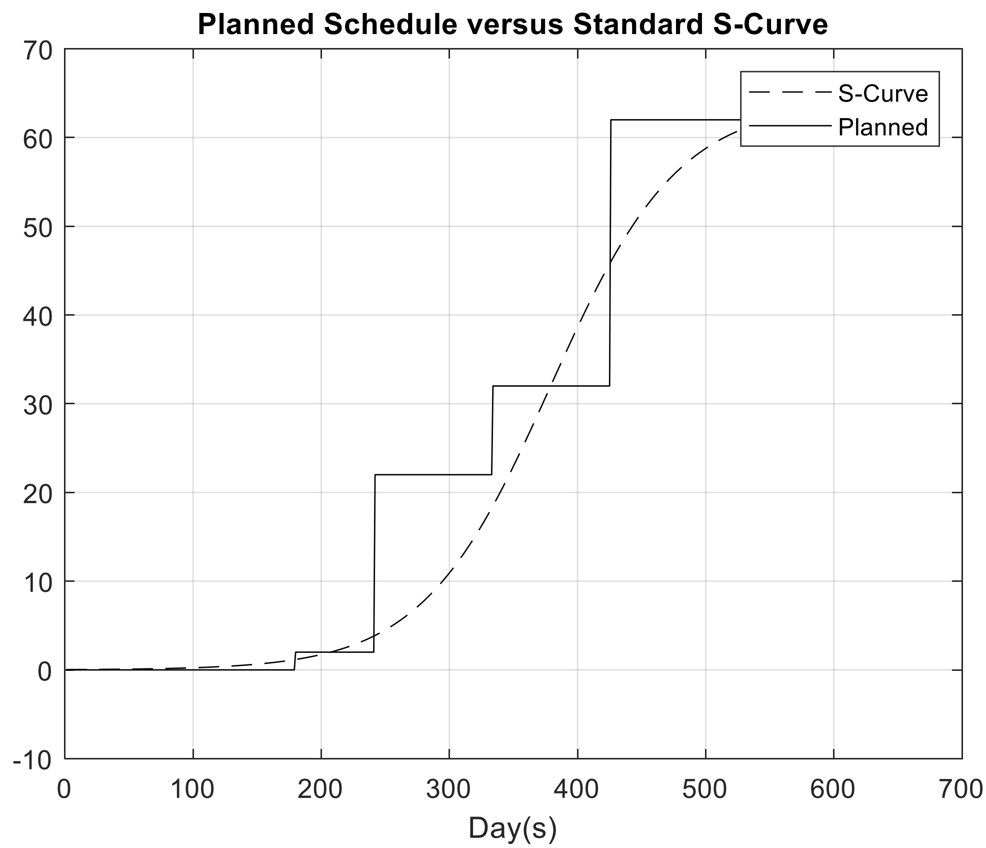


Figure 5-5. Planned Schedule versus Standard Program S-Curve

The planned program cost is depicted in **Figure 5-6**. It is expected that acceleration will impact cost in a similar manner. This may or may not be the case and will be a topic for future research. The cost can be assumed to follow a linear motion model as seen in the figure below. Cost increase occurs initially between days 50 – 280 and then reaches a more stable state.

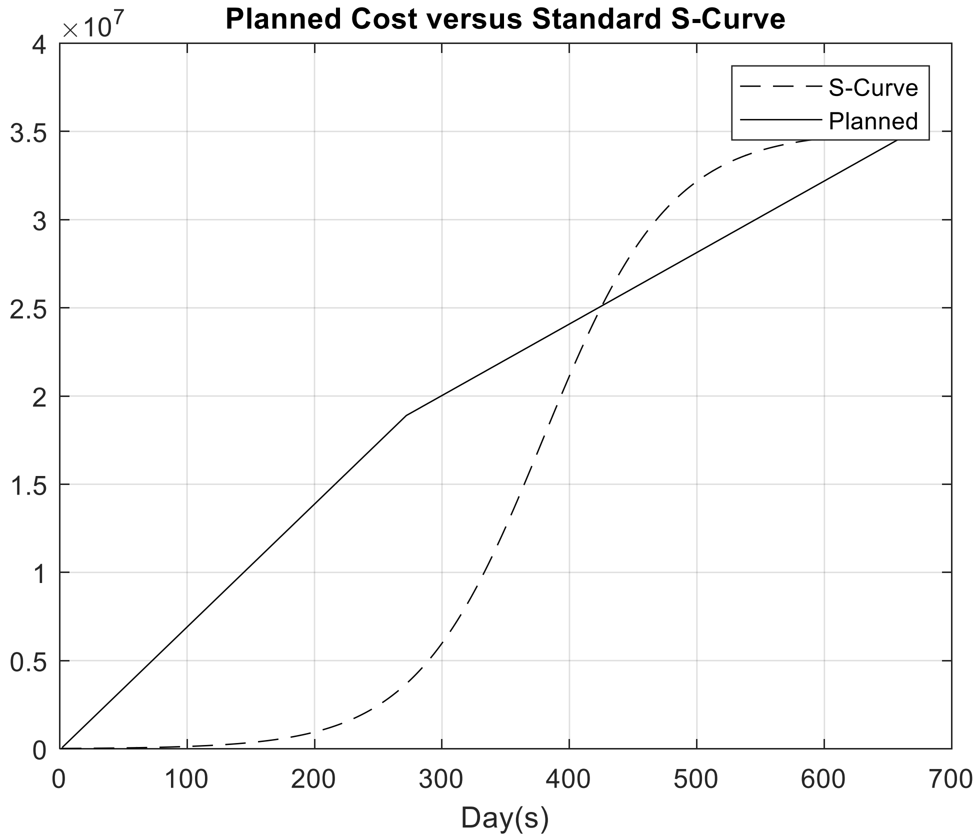


Figure 5-6. Planned Cost versus Standard Program S-Curve

5.3 MODEL BUILDING

In order to build the model, the program manager will define an acceptable risk tolerance. Two CV filters will be generated with differing process noise factors. One CV filter will have a low process noise value which will provide a relatively high margin of error for the program future state. The second CV filter will have a high process noise value. The high process noise value will enable the CV filter to estimate the current and future state of the program with a relatively lower margin of error enabling the second filter to identify rapid changes to the planned cost and schedule. For this research, the Process Noise values for each filter will be as follows:

$$R_{CV1} = .05 \; and \; R_{CV2} = 2.0 \,. \tag{5-1}$$

Risk tolerance will be incorporated into the Markov Switching Matrix (MSM) and have the following form:

$$\Pi = \begin{bmatrix} .95 & .05 \\ .1 & .9 \end{bmatrix}. \tag{5-2}$$

Therefore, the program is expected to execute according to the planned cost and schedule 95% of the time, and the first CV will provide the best estimate of current and predicted state. There is a 5% chance that the program will accelerate or decelerate according to the planned cost and schedule. Should this happen, the second CV will become the primary filter and continue to provide an estimate of cost and schedule. Continuing this line of thought, should the second CV filter reach its accepted risk level, the first filter will take over as primary and provide estimated cost and schedule. The error between planned and estimated program schedule and cost will also be computed to verify and validate filter performance.

5.4 MODEL VERIFICATION

In order to verify that the simulation model is working correctly, the S-curve will be utilized. A Monte Carlo simulation will be executed containing the S-curve as the program schedule and cost. The IMM will be utilized to estimate the current and future state of the S-curve. It is expected that the IMM will estimate the S-curve almost exactly in order to verify accurate representation of the program.

In order to verify that the IMM is modeled correctly, the CV motion model Kalman filters are provided in **Figure 5-7** and **Figure 5-8**. It can be seen in **Figure 5-7** that the CV motion model with low process noise is providing an estimate of the program schedule with a large margin of error, as defined.

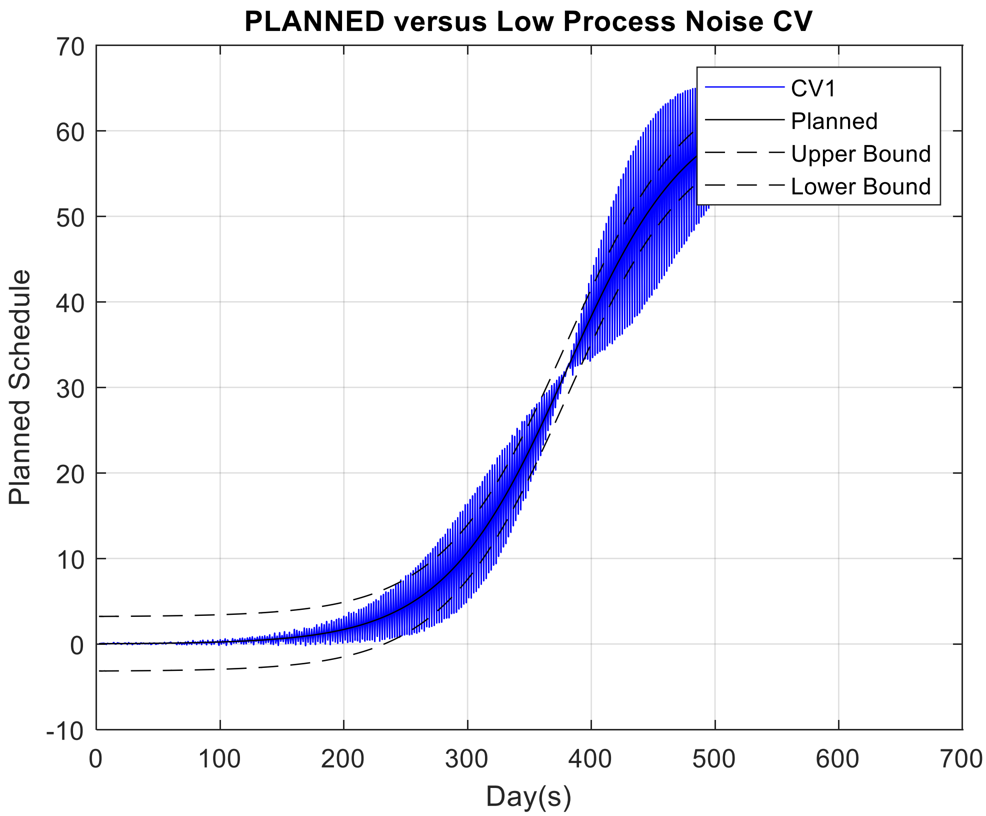


Figure 5-7. Constant Velocity (Low Process Noise Filter) estimate of S-Curve

It can be seen in **Figure 5-8** that the CV motion model with high process noise is providing an estimate of the program schedule with a low margin of error, as defined.

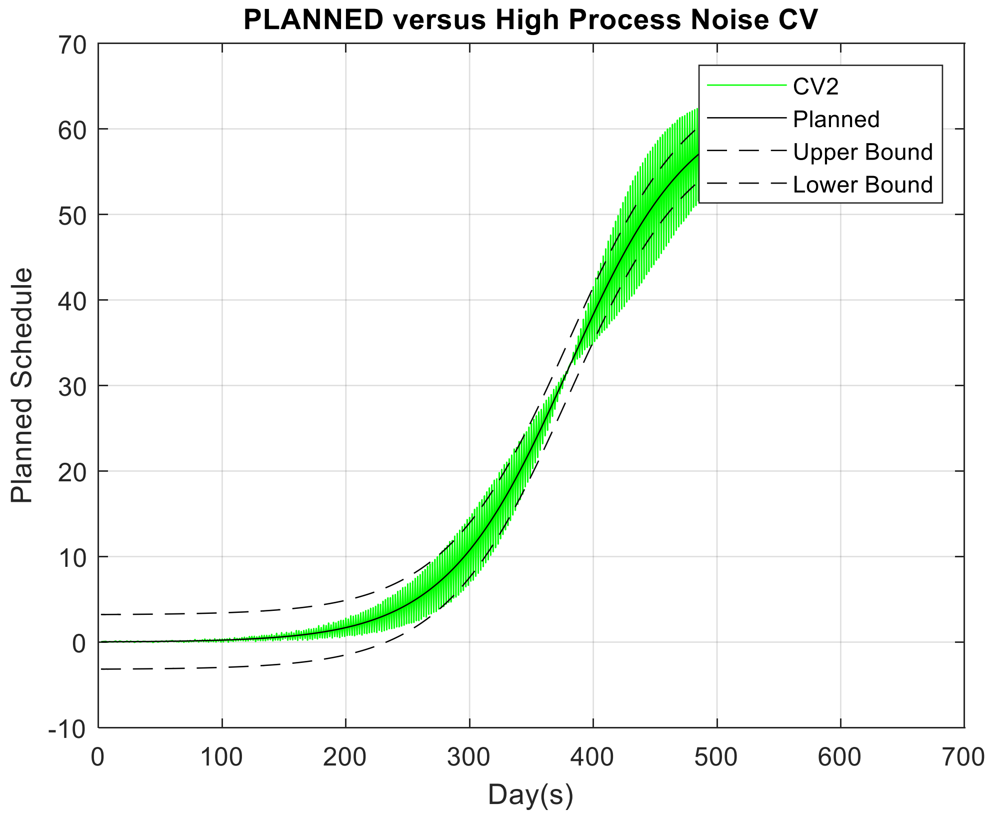


Figure 5-8. Constant Velocity (High Process Noise Filter) estimate of S-Curve

Finally, when both filters are run in parallel as part of the IMM, the estimate schedule takes into account a mixture of both the CV1 and CV2. **Figure 5-9** provides the estimated schedule produced by the IMM.

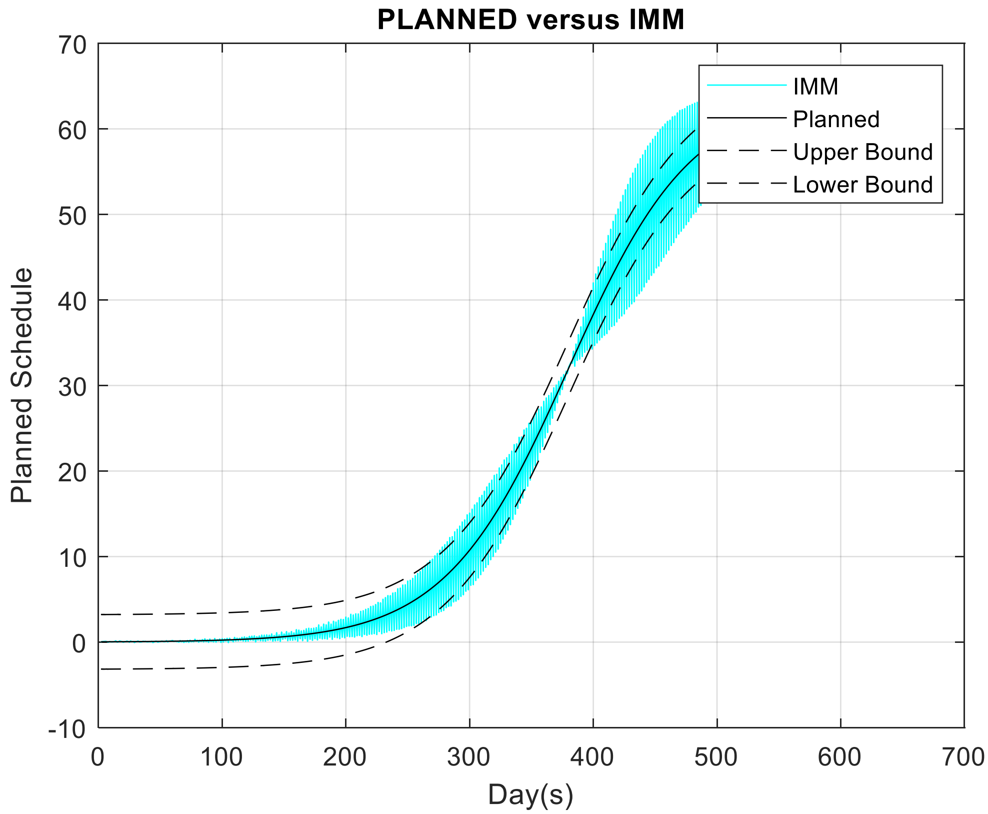


Figure 5-9. Interacting Multiple Model estimate of S-Curve

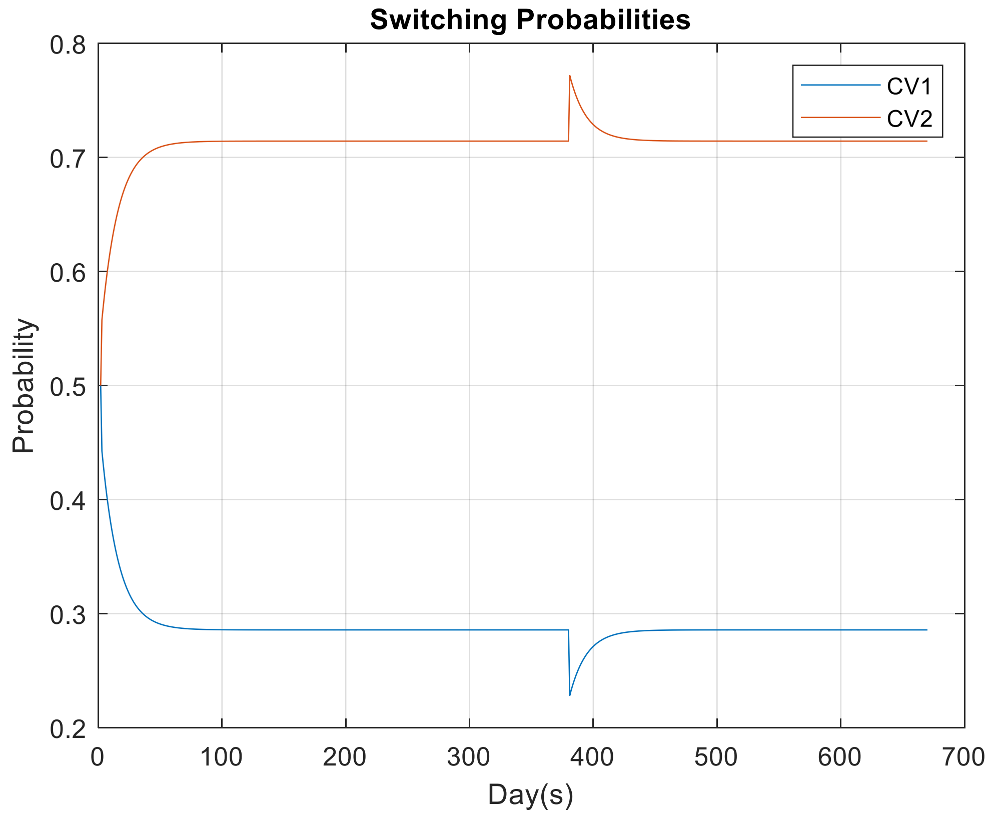


Figure 5-10. IMM Markov Switching Matrix

5.5 RUN THE SIMULATION

One hundred Monte Carlo simulations will be run under different conditions in order to vary the model parameters and initial conditions. The model parameters (i.e. process noise) will be used to simulate the acceleration parameters such as insertion of mature technology or definition of clear requirements. This will be set as previously described risk values within the MSM and will be modified in separate simulations.

5.6 MODEL VALIDATION

Model validation is the process of confirming that the model is an accurate representation of the object. Validation can be ascertained by comparing the output of the simulation with data collected from the object (i.e. development project model and lessons learned) (Gilbert, Chapter 2: Simulation as a Method, 2005). There are caveats to consider according to Gilbert:

- both model and object processes are likely to be stochastic,
- simulation may be path-dependent,
- model may not reproduce all aspects of object model, and
- model could be incorrect.

Validity in quantitative research is also improved by using the appropriate statistical analysis of the data, design of research tools, sample selection and sample size. Validity should be viewed as a continuum such that it can always be improved but will never be 100% valid (Meshguides, n.d.). Validity must be considered through all stages of research. Validity is affected by the design of the instrument to be used for data collection, researcher biases, effectiveness and accuracy in representation of instrument on data collection; therefore, these should all be considered when drawing conclusions.

It is important to first define inductive and deductive reasoning. Inductive reasoning is when the premise provides reasons to support some evidence of the truth of the conclusion. Deductive reasoning when the premise provides a guarantee of the truth of the conclusion (Copi, Cohen, & Flage, 2006). For an inductive argument, the premises are so strong and true that the conclusion is unlikely to be false. For a deductive argument to be valid, one of the following must be true: either the premise is true, or the conclusion provides such strong support for the premise that the premise has to be true. If a valid argument has premises that are true, the argument is said to be sound. An inductive argument can be affected by acquiring new premises where a deductive argument cannot (Deductive and Inductive Arguments, n.d.).

Validity, as it applies to this research, will be addressed through the selection of the appropriate program/object model representation of the actual program development model. It has been verified in Section 5.2 that the program model representation of the S-curve is an accurate representation of the program schedule and cost. The S-curve has been identified as the most reliable representation of a project's status progress and performance (Gibbs M. N., 2000). Many program managers use S-curves to evaluate a projects performance, cash flow forecast, schedule range of possibilities and quality output comparison.

The program development model will be evaluated at five points in time. These points in time are meant to coincide with the acceleration parameters and negative impacts being studied. By using the actual program development model, the establishment of bounds of acceleration are hypothesized to be based on stability attributes of reliability. Validity of the bounds to acceleration are hypothesized to be based on content validity. The S-curve has been validated as the conceptual model for the actual project development model. The validity of acceleration and negative impact parameters are hypothesized to be based on construct validity as well. As this research is based on simulation, it is important that as the simulation model is developed, it is continuously verified and validated (i.e. validation is focused on the process of proving something is valid) against the actual development model or real system and the S-curve or conceptual model to ensure the simulation model is correct (see **Figure 5-11**).

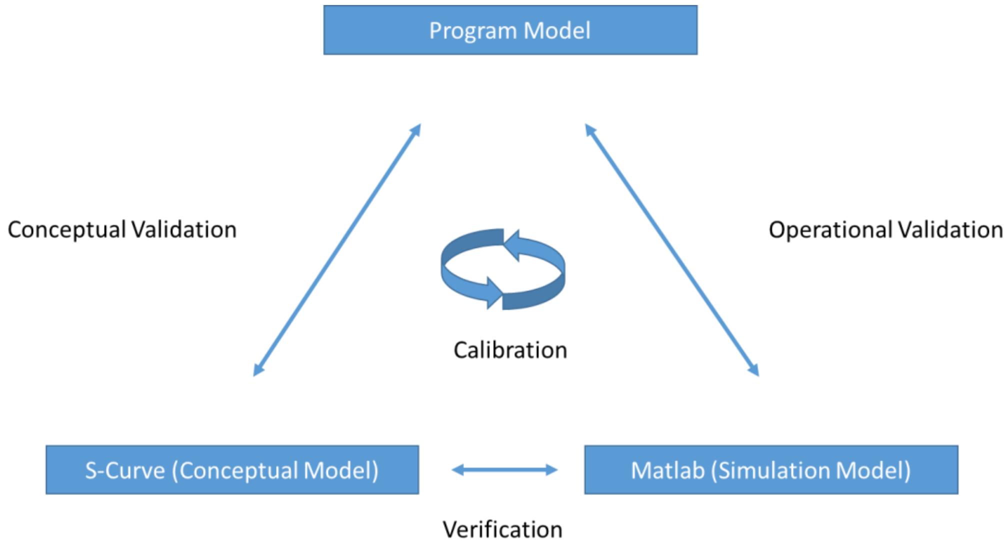


Figure 5-11. Validation and Verification in Simulation (Ulgen)

Section 5.4 verifies that the simulation model accurately predicts the future state of the program schedule and performance. With both the conceptual validation of the program model and the S-curve complete and the operational validation of the program model and simulation model complete, the model is now validated.

5.7 FINDINGS AND CONCLUSIONS

Sensitivity analysis will be performed to evaluate the effect of varying the model assumptions. Selected acceleration parameters and negative impacts will be assumed to have been chosen and implemented by the program manager, a priori. The Monte Carlo simulation will be executed in MATLAB and the results will be analyzed to determine the model's sensitivity to changes in the risk tolerance values of the MSM.

The output of the IMM will be compared to the planned program schedule and cost. The cumulative schedule and cost error (i.e. the difference between the program planned and the IMM estimate at each time, t) will be used to measure the differences between the values predicted by the model and the values planned. Conclusions will be summarized and presented in the dissertation. It is likely that recommendations will be made on which parameters, as part of the program, should be recorded as part of the program development model that currently are not. Instead, many of these parameters are captured in lessons learned upon program completion. The methodology, identified in **Figure 5-3**, will be utilized for this simulation-based research.

CHAPTER 6

RESULTS

The results in this chapter will be presented as follows. First, an overview of the Monte Carlo simulation will be presented as well as the program's planned versus actual schedule and cost. The program manager's risk tolerance levels will be defined and implemented within Sections 6.2 – 6.7 will provide the results. The results will consist of the planned program schedule and cost versus the Interacting Multiple Model (IMM) estimate of program schedule and cost, the IMM Markov Switching Matrix (MSM), the error associated with the IMM estimate versus the planned program schedule and cost.

6.1 MONTE CARLO SIMULATION OVERVIEW

The simulation will be executed suing the MathWorks© tool MATLAB. The program's planned and actual cost and schedule will be converted from Microsoft Excel into a .mat data file that can be uploaded into MATLAB. The IMM will be coded in MATLAB using two CV motion models in the form of two Kalman filters. The process noise for each CV filter will be defined such that one CV contains a low process noise value and the other CV contains a high process noise value. The MSM will account for the random variable in the simulation and will represent the program manager's approved risk tolerance. The values will range from 5% to 50%.

Recall that the planned program schedule shows the number of systems to be delivered over two years. After the first 180 days, the program manager is asked to accelerate delivery. In order to accelerate delivery, the program manager has utilized one or more of the acceleration methods identified in Section 3.1.

Figure 6-1 shows the actual deliveries based on the program managers attempt to accelerate. The program initially remains on schedule until approximately day 240. At this point, the program remains at steady state without acceleration. This lends itself to questioning whether the program has reached an undesirable risk level.

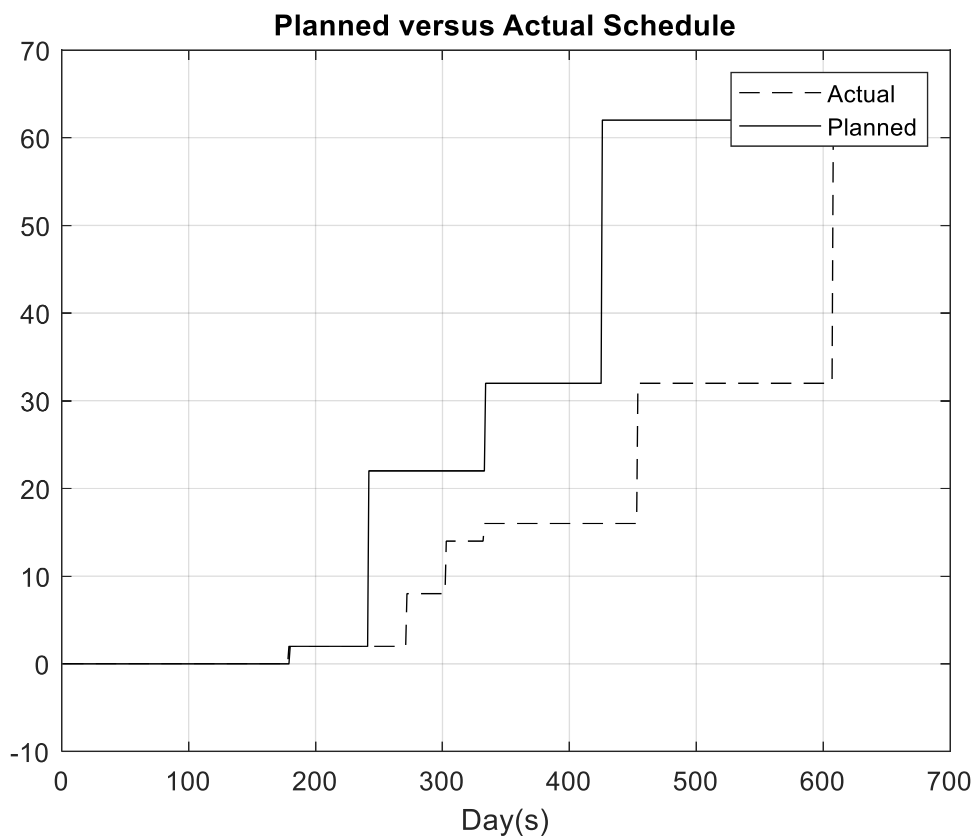


Figure 6-1. Planned versus Actual Program Schedule

Figure 6-2 provides the planned program cost versus the actual program cost. There is the tendency to expect increase cost associated with acceleration. It can be seen that the planned cost is maintained.

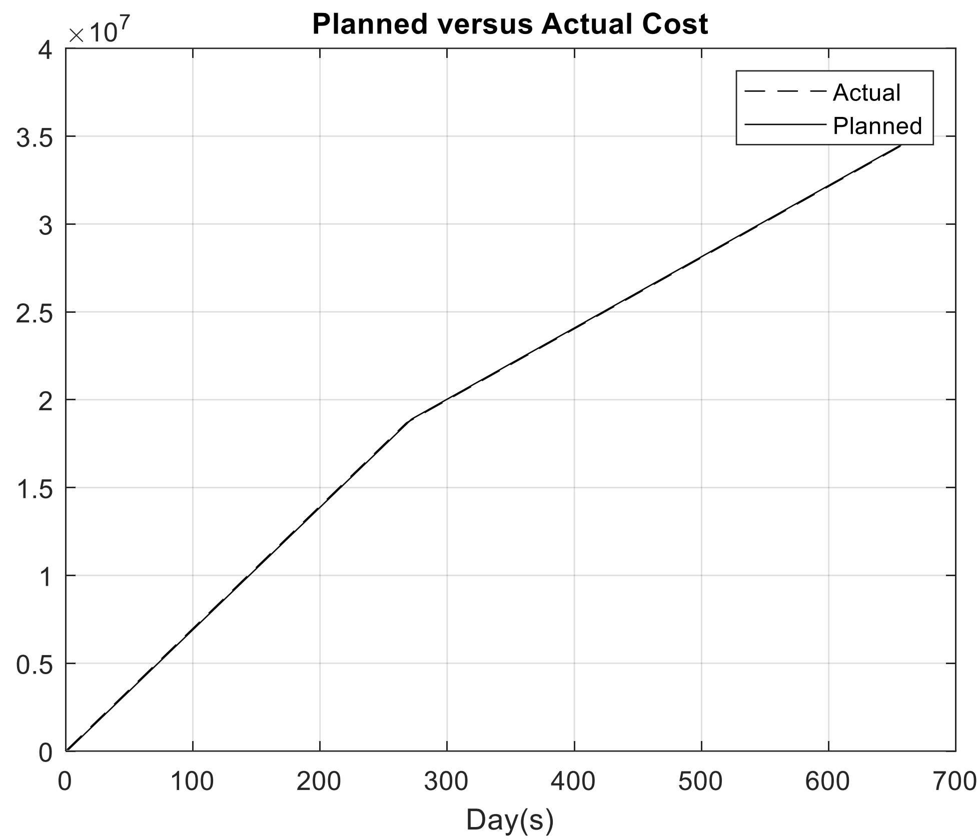


Figure 6-2. Planned versus Actual Program Cost

In order to understand risk tolerance, the margins of error must be determined and compared to our planned schedule and cost. In order to calculate the upper and lower bounds, the planned schedule at each time, t, is multiplied by the program manager's desired risk level for each time, t. The output is added to the planned schedule to determine the upper bound and subtracted from the planned schedule to create the lower bound.

Figure 6-3 provides an example of the upper and lower bound for the planned schedule assuming 5% risk tolerance.

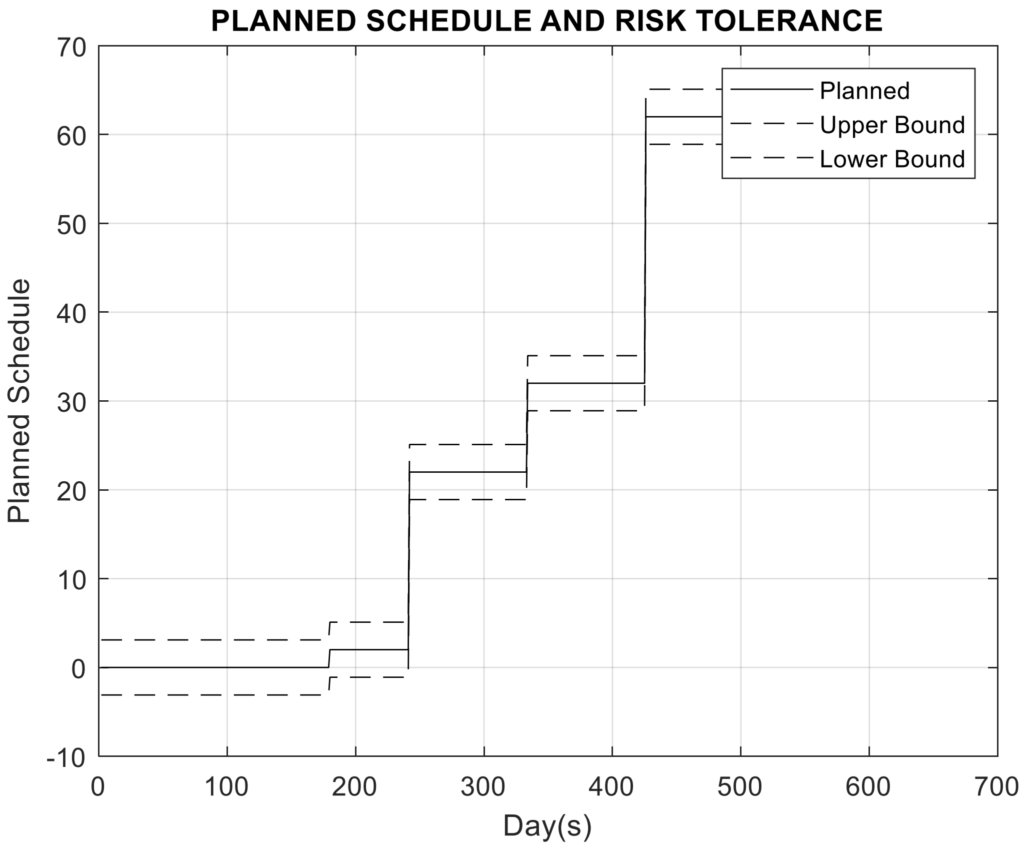


Figure 6-3. Planned Schedule with 5% Risk Tolerance

The following sections will examine the following risk tolerances:

- Case 1: 5% Risk Tolerance
- Case 2: 10% Risk Tolerance
- Case 3: 15% Risk Tolerance
- Case 4: 20% Risk Tolerance
- Case 5: 25% Risk Tolerance
- Case 6: 50% Risk Tolerance

6.2 CASE 1 –ASSUME 5% RISK TO SCHEDULE AND COST

For Case 1, the program manager has assumed a 5% risk tolerance for both schedule and cost. In order to do so, we must first set MSM values as follows:

$$\Pi = \begin{bmatrix} .95 & .05 \\ .02 & .98 \end{bmatrix}. \qquad (6\text{-}1)$$

The first row of the MSM will reflect the program manager's desire for 5% risk tolerance. Hence, the first row and first column value will be .95. This means that at each time, t, there is a 95% chance that the first CV filter will accurately predict future program schedule and cost. When time, t, occurs in which the future estimate is less than 95% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter. The values for the second row of the MSM have been chosen randomly based on the limitations of the data. Alternative approaches will be the focus of a future study.

The number of times that the model switches provides an indication to program managers that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 5% of the planned schedule. See **Figure 6-4**. This gives the program manager a visual representation of risk.

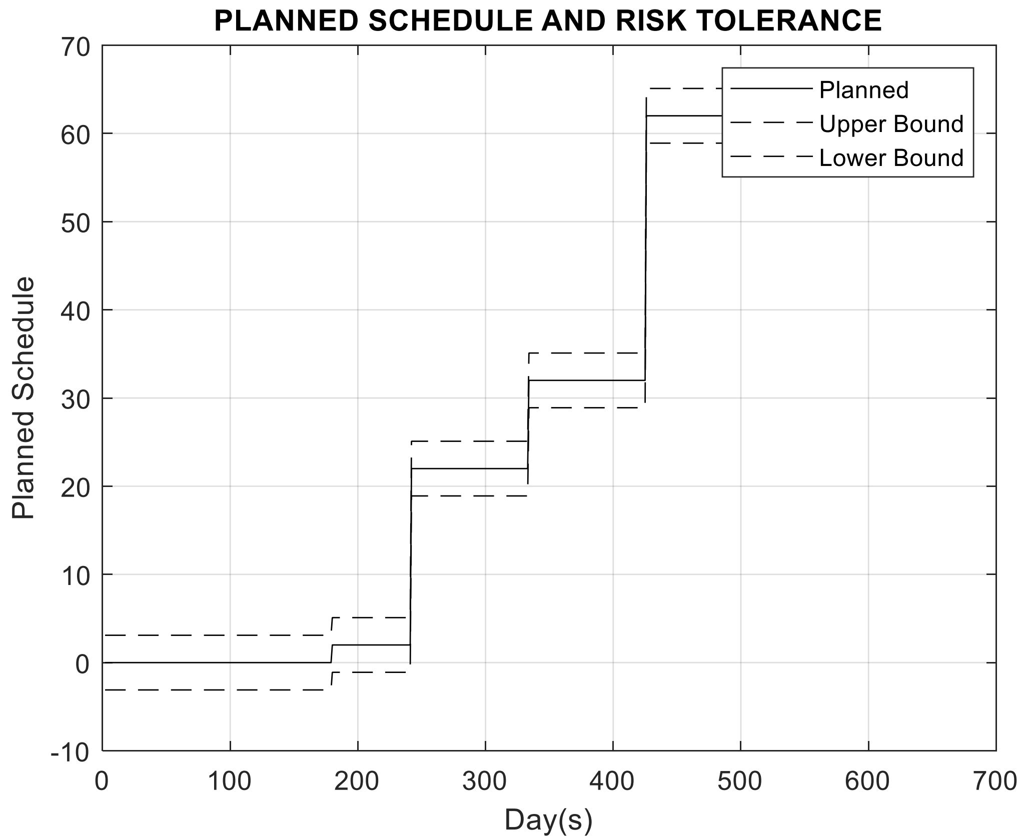


Figure 6-4. Planned Schedule with 5% Risk Tolerance

The comparison between the IMM and the planned schedule is presented in **Figure 6-5**. It is shown that the IMM does track the planned program schedule quite accurately. At day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2. Now examine day 240 in **Figure 6-5**. The planned delivery is 22, but the actual delivery remains at 2. Hence, the program has not accelerated but is actually decelerating. It can be observed that the actual value of day 240 is less than the lower bound; therefore, the program has assumed more than the 5% risk deemed acceptable by the program manager. This logic can be continued by examining the following dates: 340 and 450.

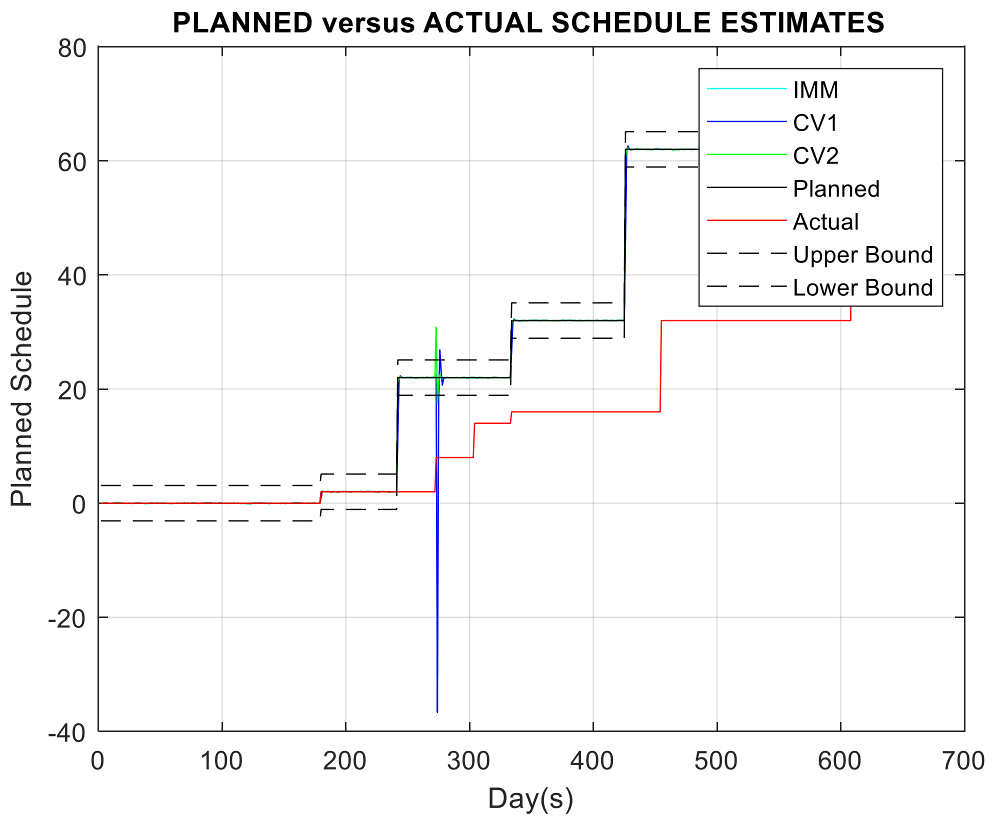


Figure 6-5. Planned versus Actual Program Schedule Estimates Assuming 5% Risk

Figure 6-6 captures the planned versus actual cost estimate. Additionally, a 5% risk tolerance is added to the graph by computing ± 5% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost is examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-6**, around day 272, the planned cost indicates a slight jump in cost. However, the program actual cost is consistent with the planned cost. It is observed that the program has not assumed more than the 5% risk deemed acceptable by the program manager relative to program cost.

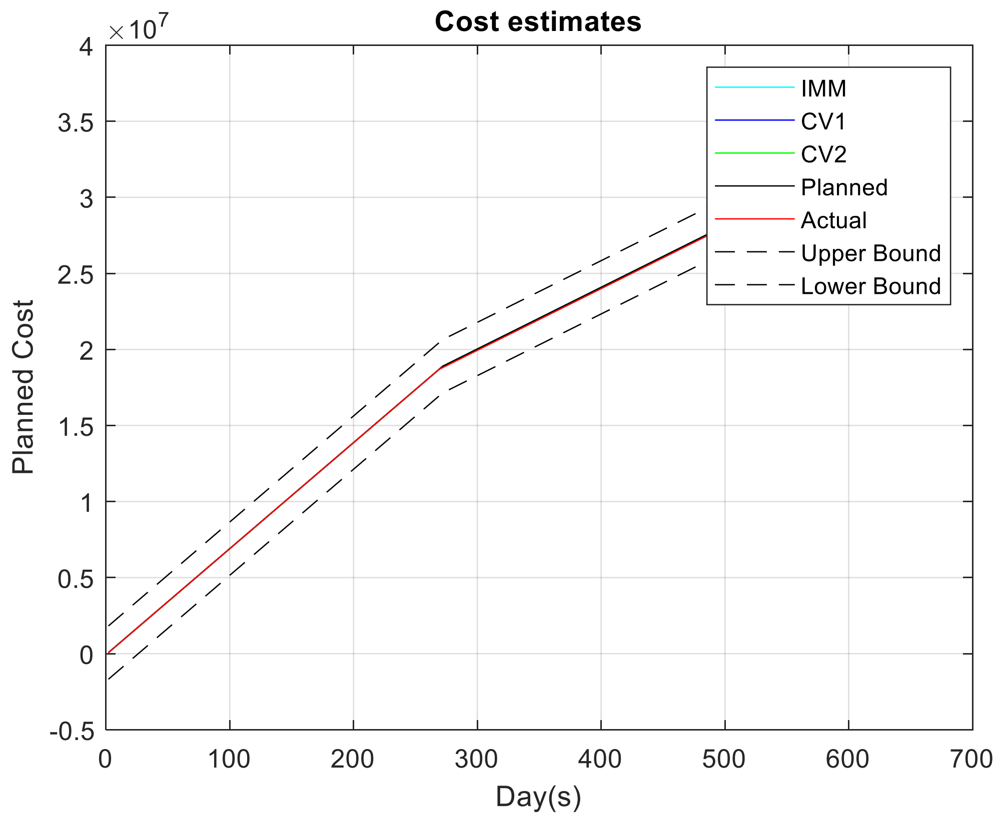


Figure 6-6. Planned versus Actual Program Cost Estimates Assuming 5% Risk

Figure 6-7 shows the switching probabilities of the IMM based on the planned program schedule and cost. It is observed that the IMM picks up the program planned schedule and cost changes consistently with the program days: 190, 240, 272, 340 and 450.

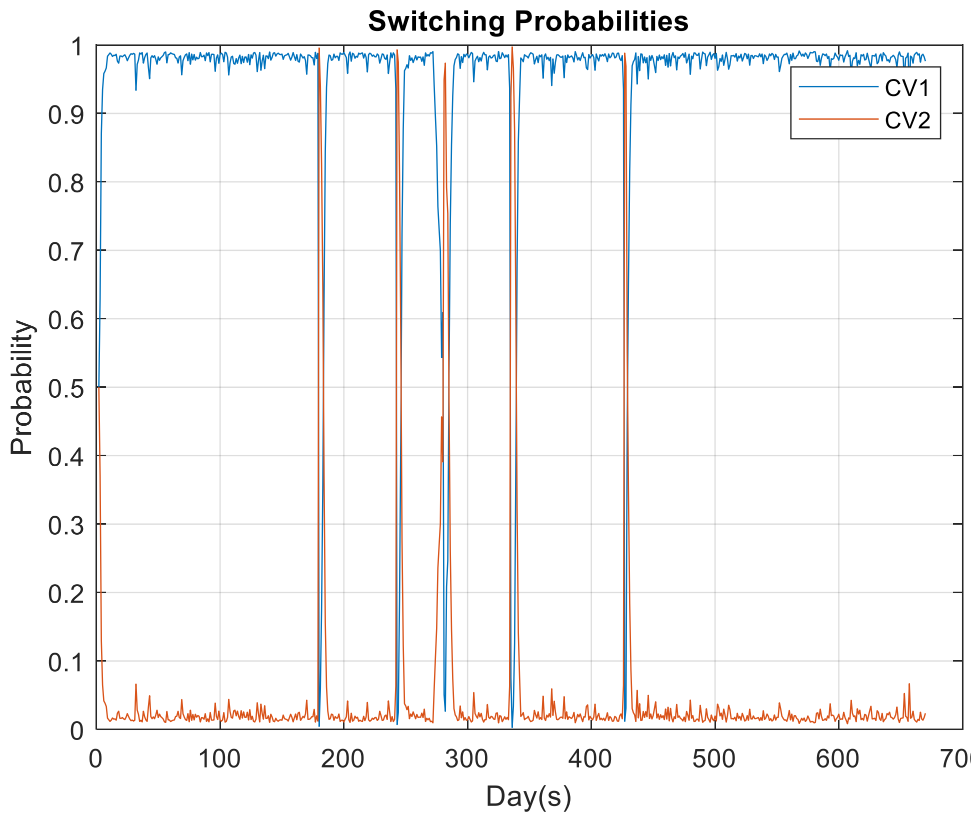


Figure 6-7. Interacting Multiple Model Switching Probabilities Assuming 5% Risk

It can be deduced that the IMM filter does identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance. **Table 6-1** is a summary of the points in which the IMM detected a change in the planned program schedule and cost.

5%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	N	Y	N	N

Table 6-1. Switching Probability Summary for 5% Risk Tolerance

At day 190, the program planned to delivery 2 systems and the delivered 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained 2. By examining the program manager's risk tolerance in **Figure 6-5**, it is clear that the risk tolerance exceeded 5%. On day 272, the planned versus actual cost differences did not exceed the 5% risk tolerance. On day 240, the program planned delivery of 32 systems, while actual program delivery was 16 systems. Once again, by referring to **Figure 6-5**, it can be deduced that the program has again exceeded the 5% risk tolerance threshold. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems. **Figure 6-5** shows that the risk tolerance has been exceeded on day 420.

In order to validate the program estimate, the schedule estimate error is computed by taking the difference between the planned and estimated schedule produced by the IMM. By examining **Figure 6-**, the schedule estimate errors produced by the IMM coincide with the planned program changes on day: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-**.

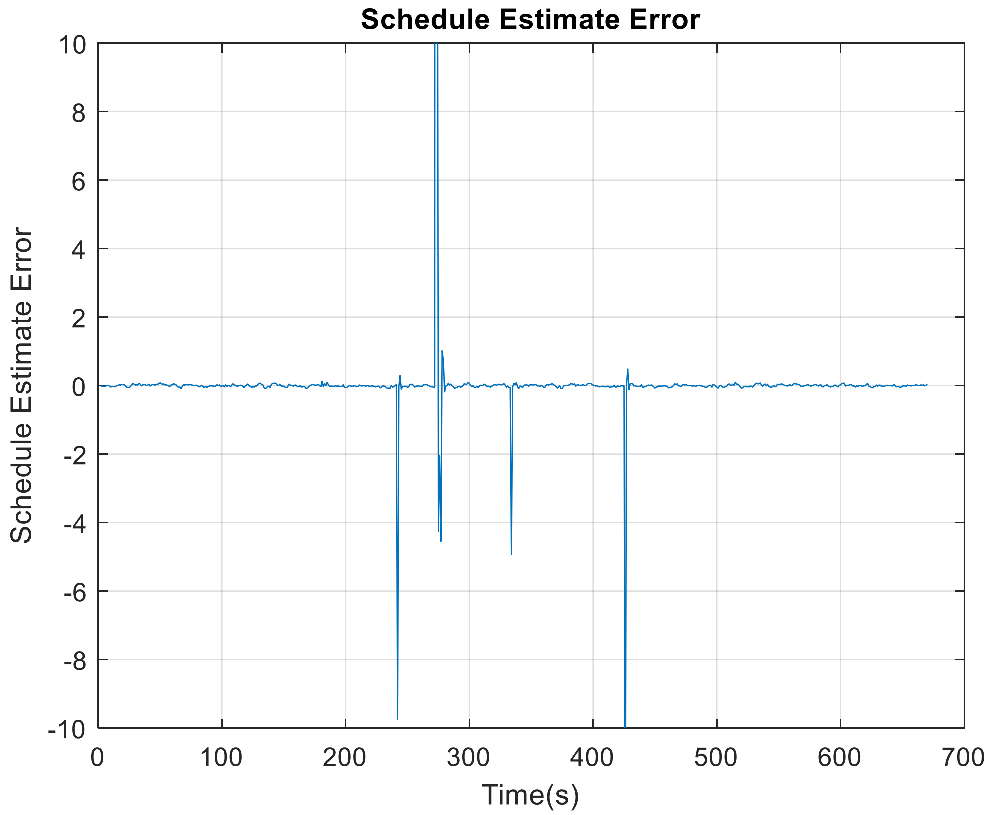


Figure 6-8. Schedule Estimate Error (5%)

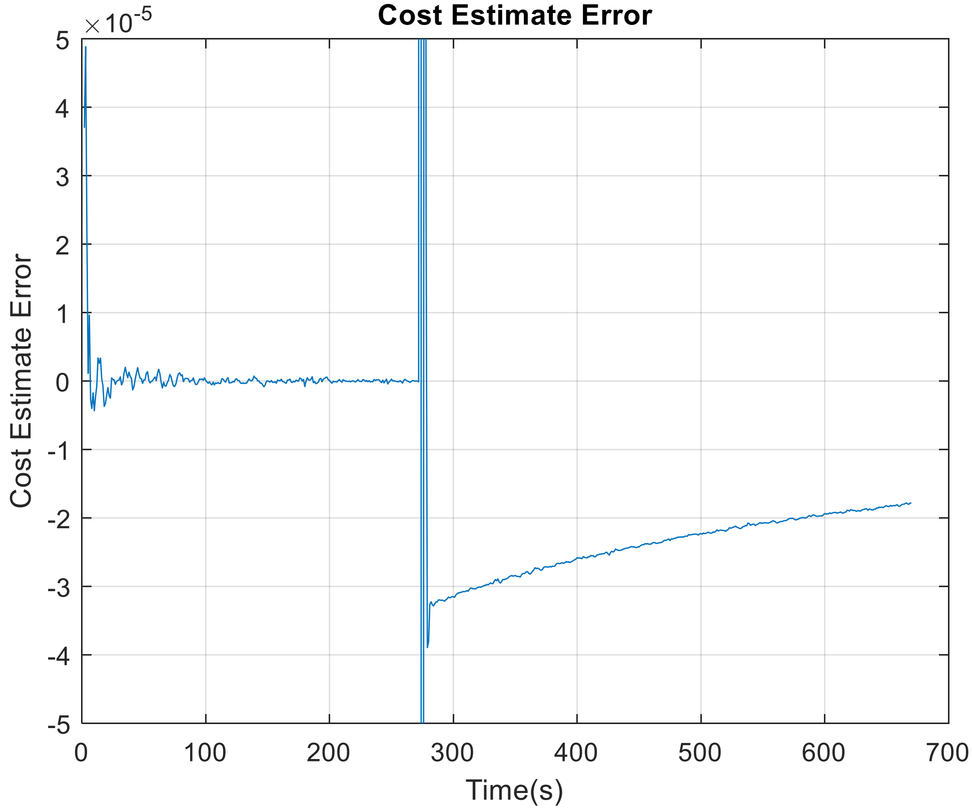


Figure 6-9. Cost Estimate Error (5%)

6.3 CASE 2 – ASSUME 10% RISK TO SCHEDULE AND COST

For Case 2, the program manager has assumed a 10% risk tolerance for both schedule and cost. In order to do so, we must first set MSM values as follows:

$$\Pi = \begin{bmatrix} .90 & .10 \\ .02 & .98 \end{bmatrix}. \tag{6-2}$$

The first row of the MSM will reflect the program manager's desire for 10% risk tolerance. Hence, the first row and first column value will be .90. This means that at each time, t, there is a 90% chance that the first CV filter will accurately predict future program schedule and cost. When time, t, occurs in which the future estimate is less than 90% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter.

The number of times that the model switches provides an indication to the program managers that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 10% of the planned schedule. See **Figure 6-10**. This gives the program manager a visual representation of risk.

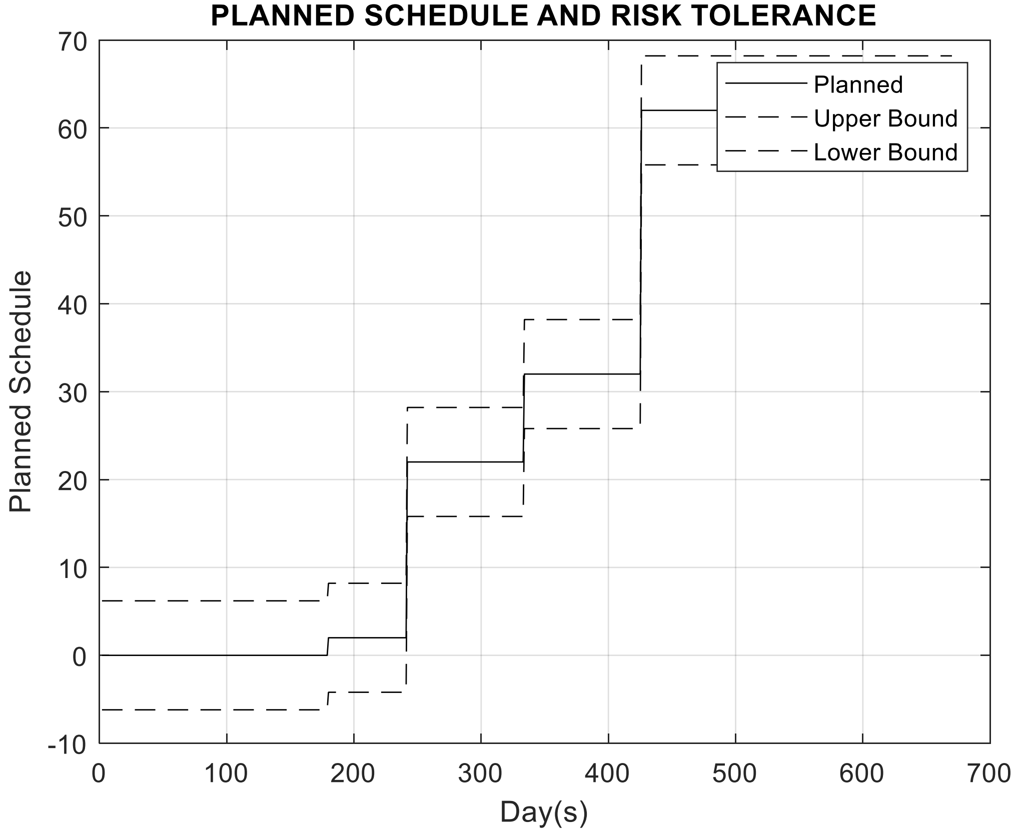


Figure 6-10. Planned Schedule with 10% Risk Tolerance

The comparison between the IMM and the planned schedule is presented in **Figure 6-1**. It is shown that the IMM does track the planned program schedule quite accurately. At day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2.

Now examine day 240 in **Figure 6-1**. The planned delivery is 22, but the actual delivery remains at 2. Hence, the program has not accelerated but is decelerating. It can be observed that the actual value of day 290 is less than the lower bound; therefore, the program has assumed more than the 10% risk deemed acceptable by the program manager. This logic can be continued by examining the following dates: 340 and 450.

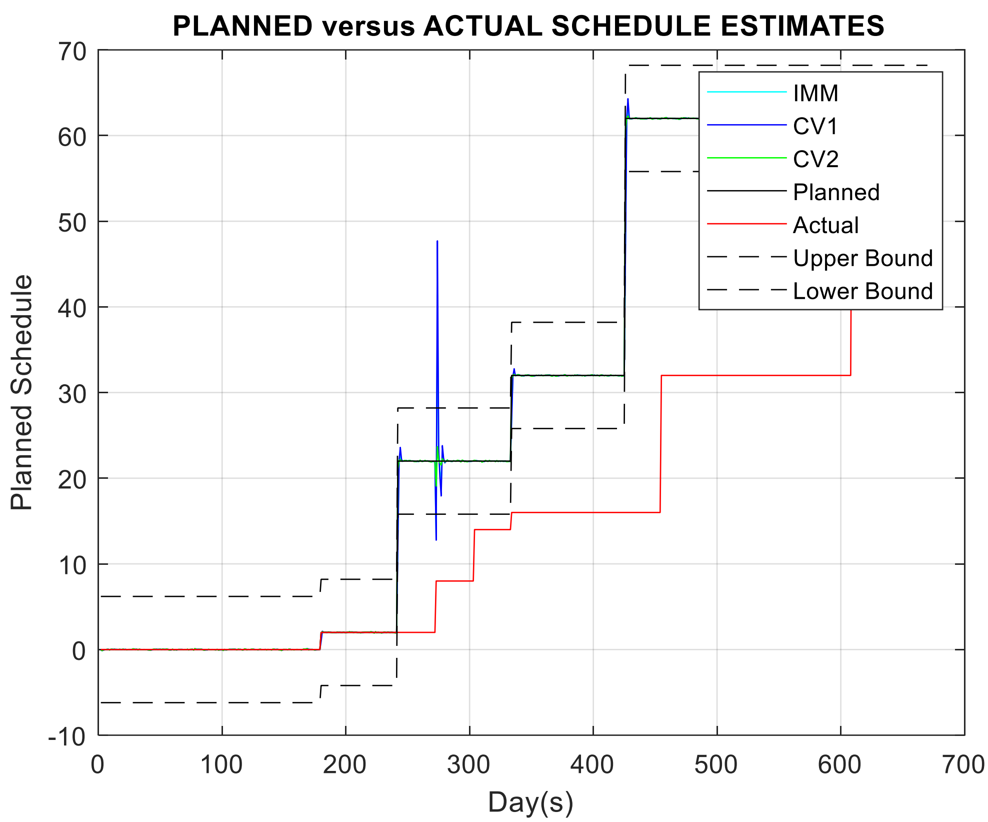


Figure 6-11. Planned versus Actual Program Schedule Estimates Assuming 10% Risk

Figure 6-2 captures the planned versus actual cost estimate. Additionally, a 10% risk tolerance is added to the graph by computing ± 10% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost should be examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-**, around day 272, the planned cost indicates a slight jump in cost. However, the program's actual cost is consistent with the planned cost. It can be observed that the program has not assumed more than the 10% risk deemed acceptable by the program manager relative to program cost.

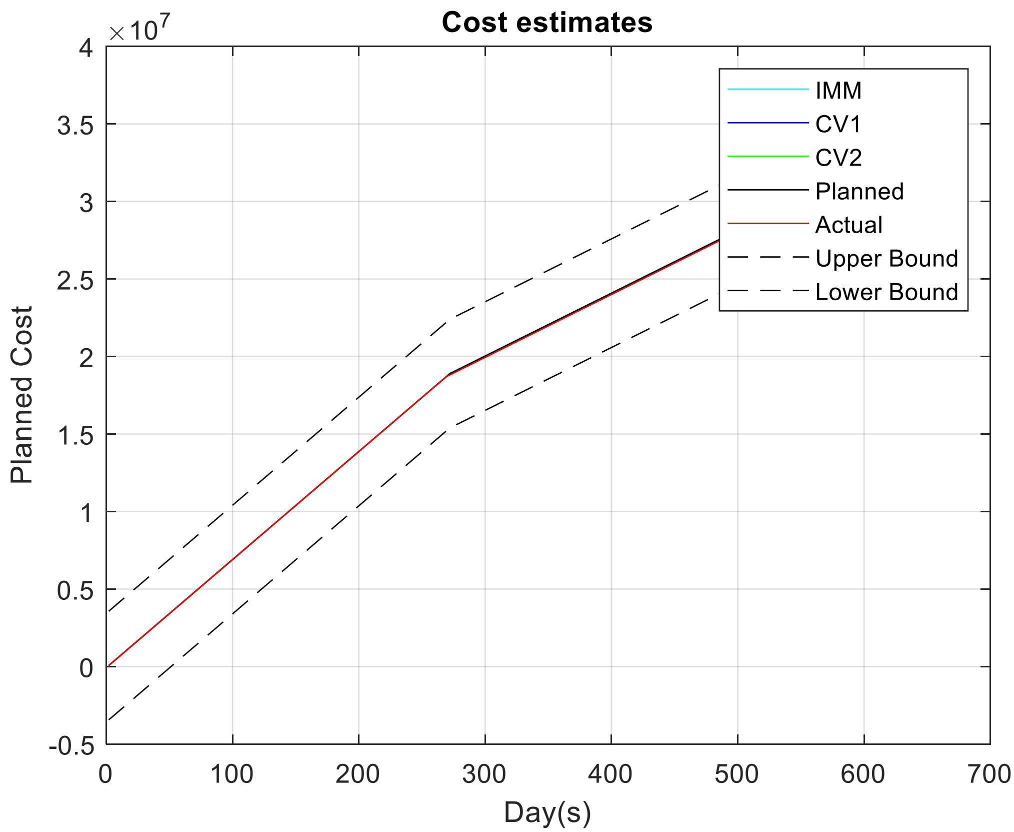


Figure 6-12. Planned versus Actual Program Cost Estimates Assuming 10% Risk

Figure 6-6 shows the switching probabilities of the IMM based on the planned program schedule and cost. It can be observed that the IMM picks up the program planned schedule and cost changes consistently with the program days of 190, 240, 272, 340 and 450.

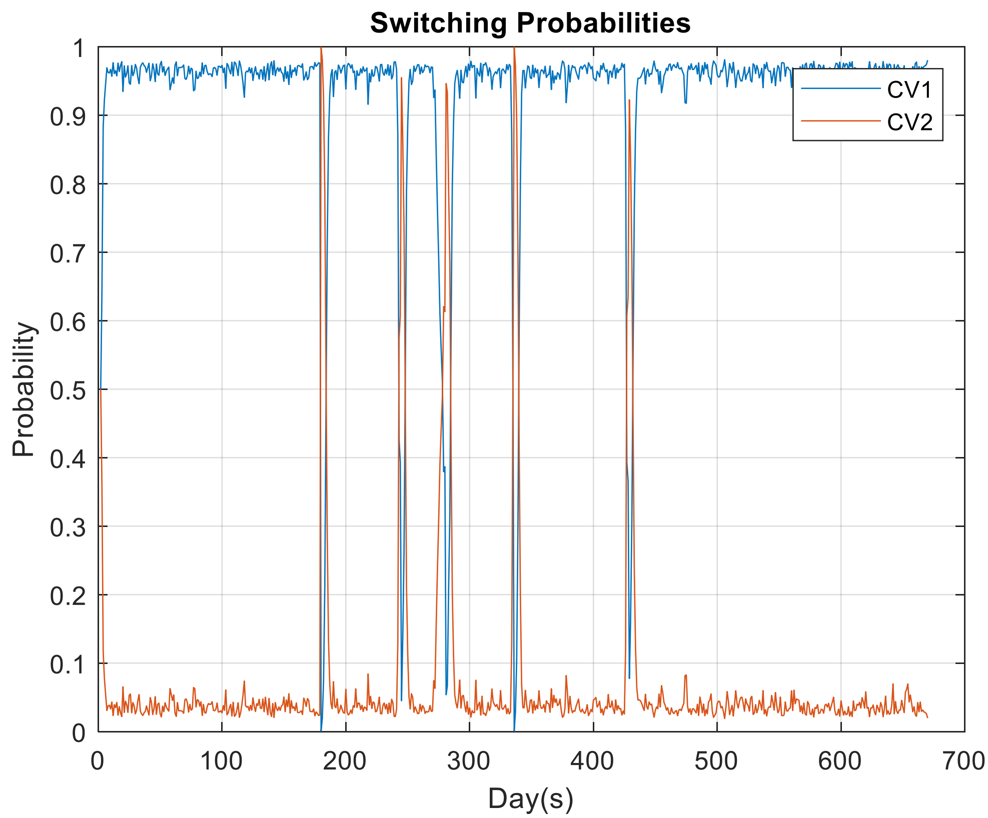


Figure 6-6. Interacting Multiple Model Switching Probabilities Assuming 10% Risk

It can be deduced that the IMM filter did identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance.

Table 6-2 is a summary of the points in which the IMM detected a change in the planned program schedule and cost. At day 190, the program planned to deliver 2 systems and did deliver 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained 2. By examining the program manager's risk tolerance in **Figure 6-**, it is clear, the risk tolerance exceeded 10%. On day 272, the planned versus actual cost differences did not exceed the 10% risk tolerance. On day 240, the program planned delivery of 32 systems, while actual program delivery was 16 systems.

Once again, by referring to **Figure 6-**, it can be deduced that the program has again exceeded the 10% risk tolerance threshold. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems. **Figure 6-**, again, shows that the risk tolerance has been exceeded on day 420.

10%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	N	Y	N	N

Table 6-2. Switching Probability Summary for 10% Risk Tolerance

By examining **Figure 6-7**, the schedule estimate errors produced by the IMM coincide with the planned program changes on day: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-8**.

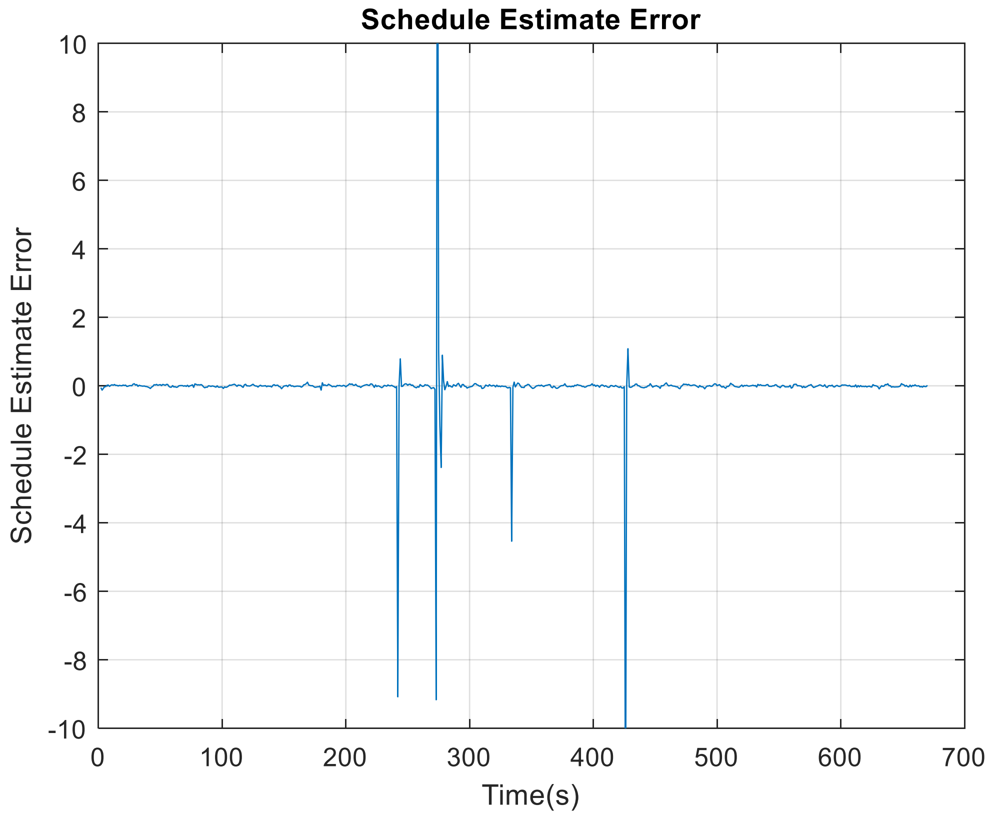


Figure 6-7. Schedule Estimate Error (10%)

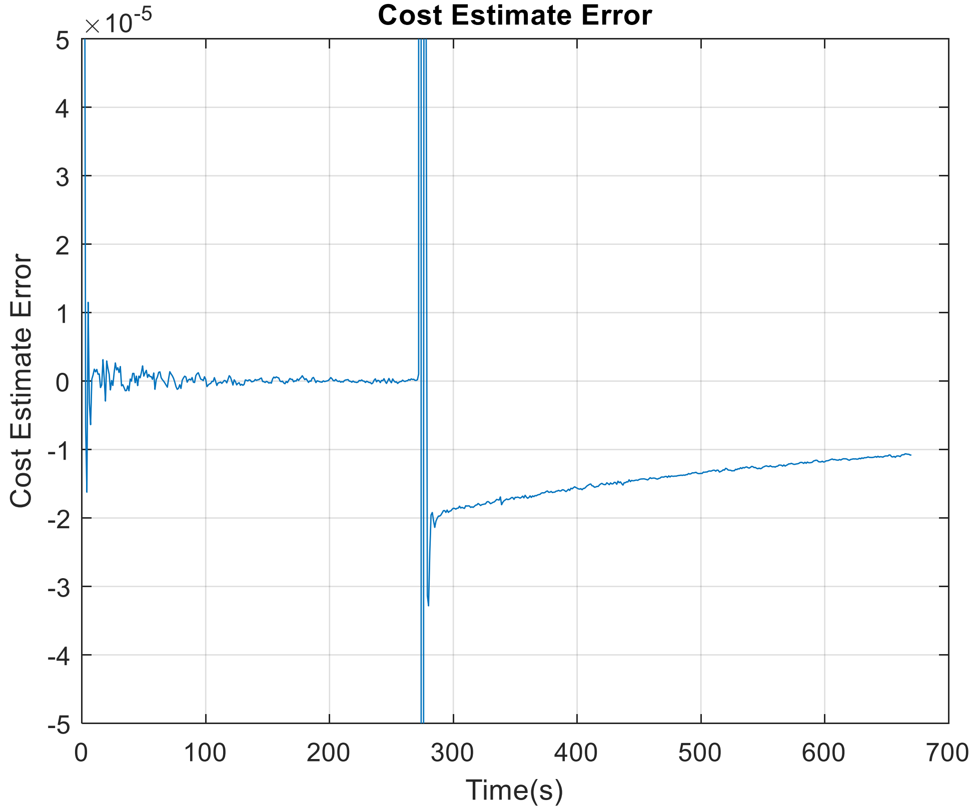


Figure 6-8. Cost Estimate Error (10%)

6.4 CASE 3 – ASSUME 15% RISK TO SCHEDULE AND COST

For Case 3, the program manager has assumed a 15% risk tolerance for both schedule and cost. In order to do so, we must first set MSM values as follows:

$$\Pi = \begin{bmatrix} .85 & .15 \\ .02 & .98 \end{bmatrix}. \tag{6-3}$$

The first row of the MSM will reflect the program manager's desire for 15% risk tolerance. Hence, the first row and first column value will be .85. This means that at each time, t, there is an 85% chance that the first CV filter will accurately predicts future program schedule and cost. When time, t, occurs in which the future estimate is less than 85% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter.

The number of times that the model switches provides an indication to the program managers that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 15% of the planned schedule. See **Figure 6-9**. This will give the program manager a visual representation of risk.

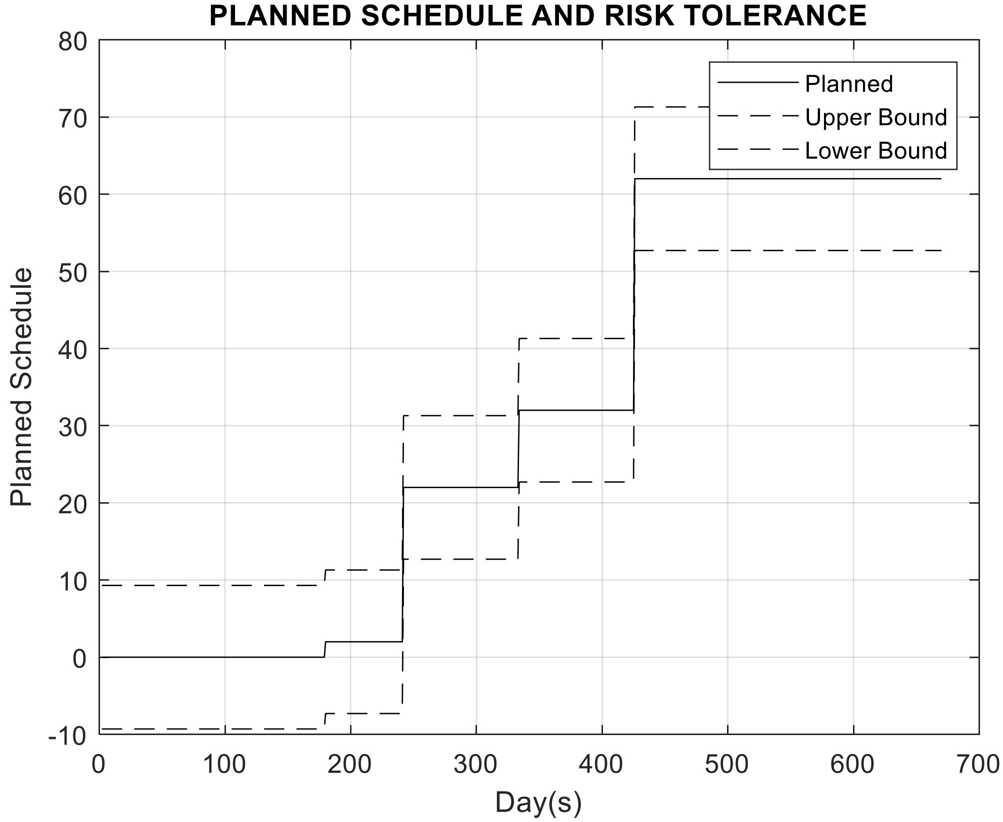


Figure 6-9. Planned Schedule with 15% Risk Tolerance

After running the IMM, the comparison between the IMM and the planned schedule should be examined. It is shown that the IMM does track the planned program schedule quite accurately. For example, in **Figure 6-107** at day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2.

Now examine day 240 in **Figure 6-10**. The planned delivery is 22, but the actual delivery remains at 2. Hence, the program has not accelerated but decelerated. It can be observed that the actual value of day 290 is less than the lower bound; therefore, the program has assumed more than the 15% risk deemed acceptable by the program manager. Alternatively, the actual value of day 340 is within the lower bound; therefore, the program has accelerated and is assumed less than the 15% risk deemed acceptable by the program manager. Finally, day 450 indicates that the program has fallen behind schedule again and has assumed more than 15% risk.

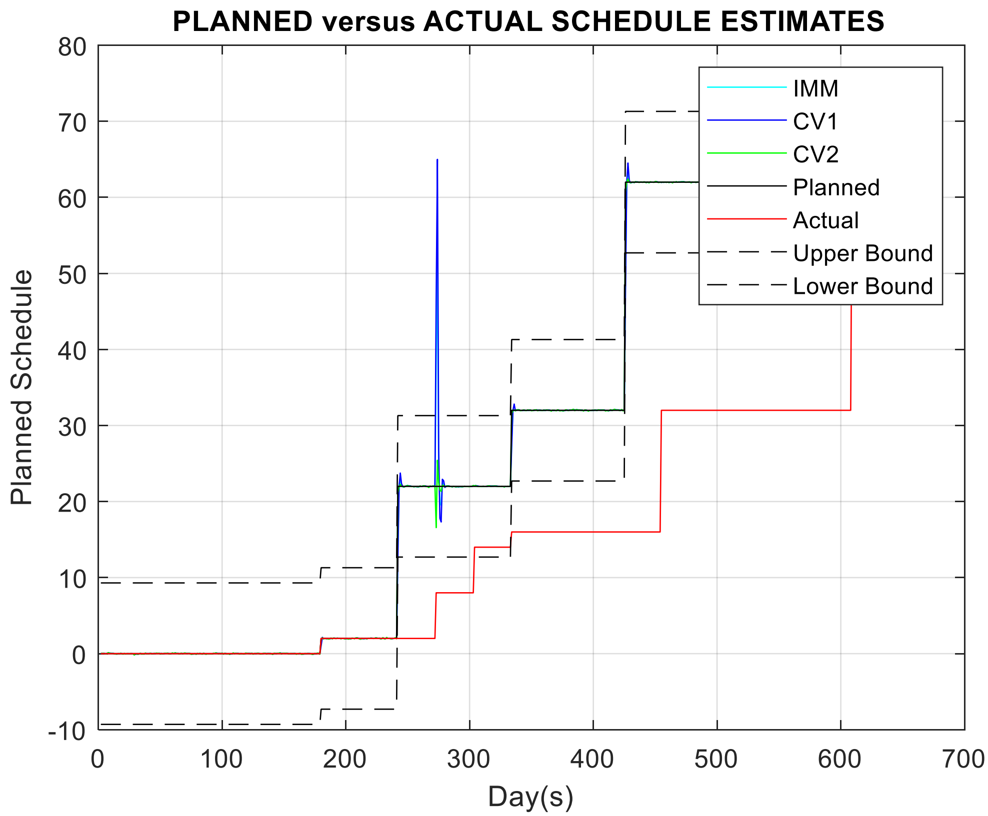


Figure 6-10. Planned versus Actual Program Schedule Estimates Assuming 15% Risk

Figure 6-11 captures the planned versus actual cost estimate. Additionally, a 15% risk tolerance is added to the graph by computing ± 15% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost is examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-11**, around day 272, the planned cost indicates a slight jump in cost. However, the program actual cost is consistent with the planned cost. It can be observed the program has not assumed more than the 15% risk deemed acceptable by the program manager relative to program cost.

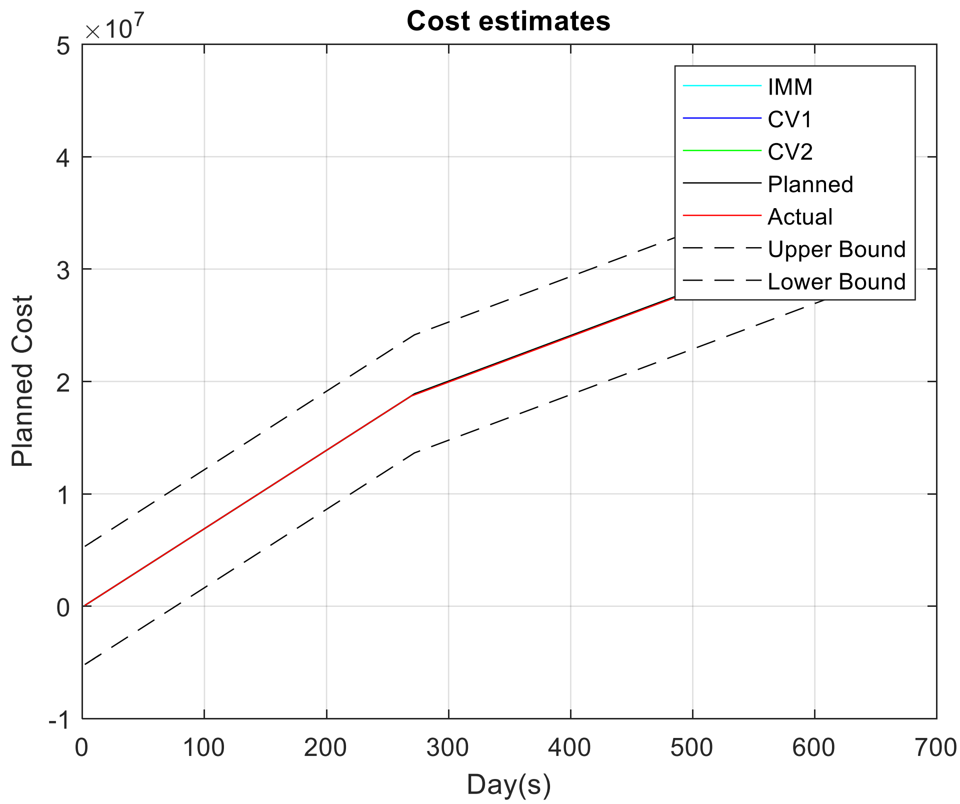


Figure 6-11. Planned versus Actual Program Cost Estimates Assuming 15% Risk

Figure 6- shows the switching probabilities of the IMM based on the planned program schedule and cost. It can be observed that the IMM picks up the program planned schedule and cost changes consistently with the program days: 190, 240, 272, 340 and 450.

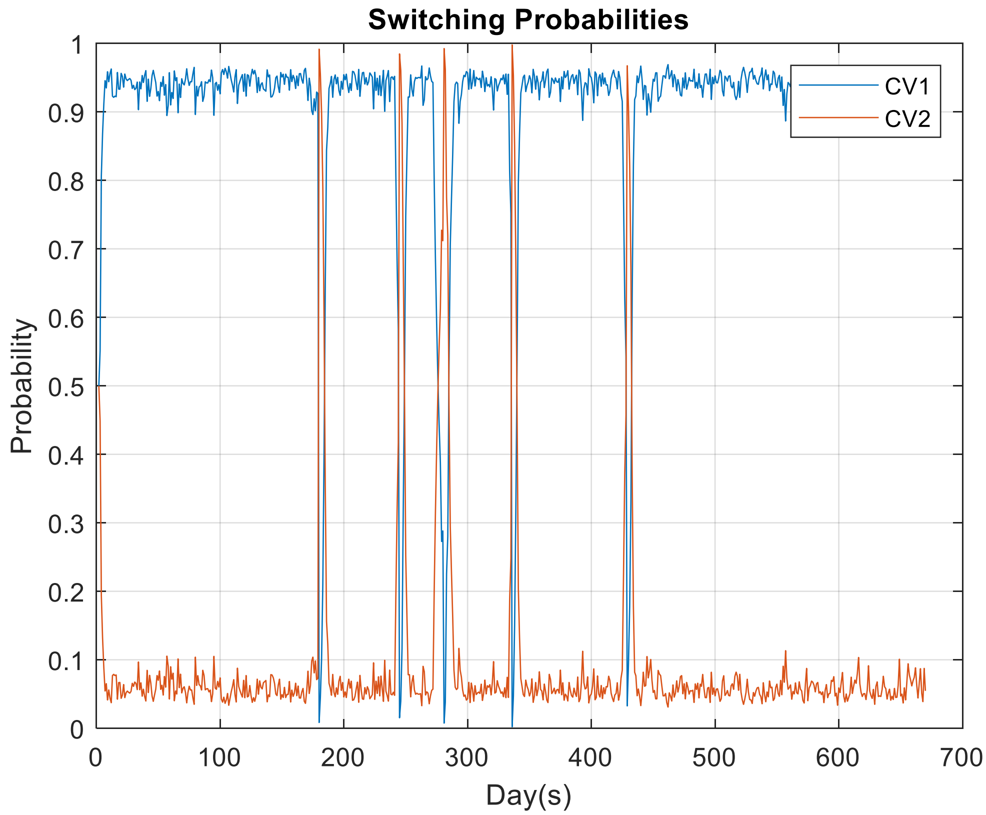


Figure 6-19. Interacting Multiple Model Switching Probabilities Assuming 15% Risk

The IMM filter did identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance. **Table 6-3** is a summary of the points in which the IMM detected a change in the planned program schedule and cost. At day 190, the program planned to deliver 2 systems and did deliver 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained 2.

By examining the program managers risk tolerance in **Figure 6-10**, it is clear, that the risk tolerance exceeded 15%. On day 272, the planned versus actual cost differences did not exceed the 15% risk tolerance. On day 240, the program planned delivery of 32 systems. The actual program delivery was accelerated to 16 systems and was within the program manager's risk tolerance level. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems. **Figure 6-10**, again, shows that the risk tolerance has been exceeded on day 420.

15%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	N	Y	N	N

Table 6-3. Switching Probability Summary for 15% Risk Tolerance

By examining **Figure 6-12**, the schedule estimate errors produced by the IMM coincide with the planned program changes on day: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-**.

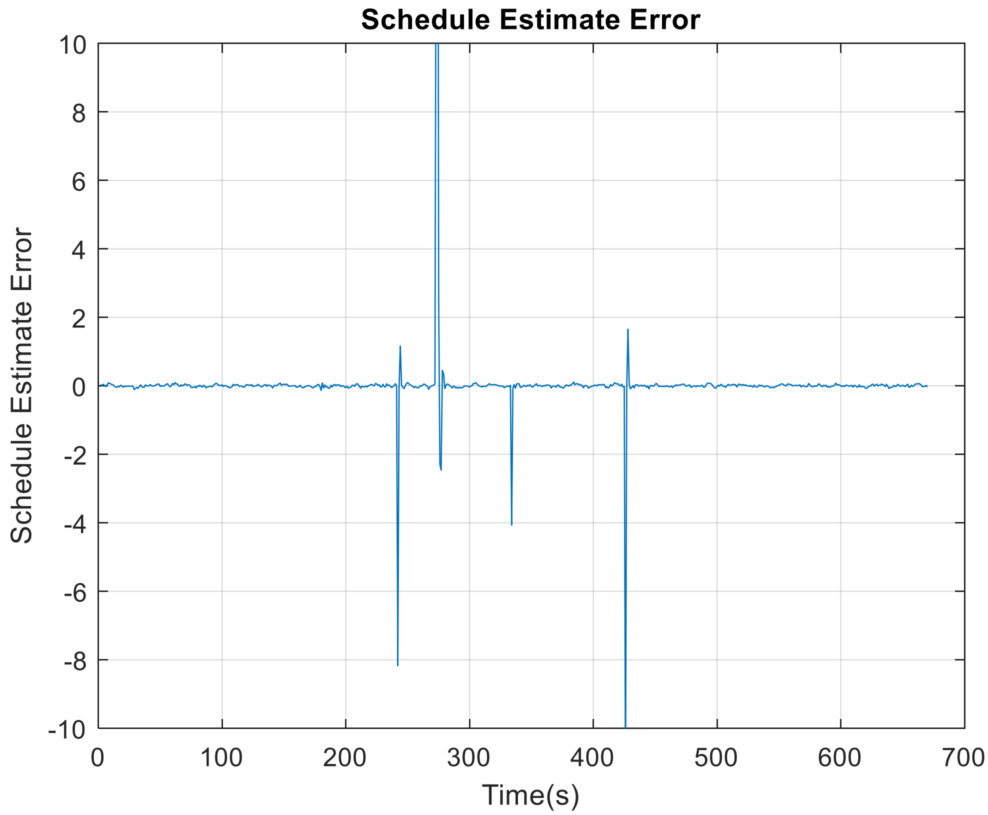


Figure 6-12. Schedule Estimate Error (15%)

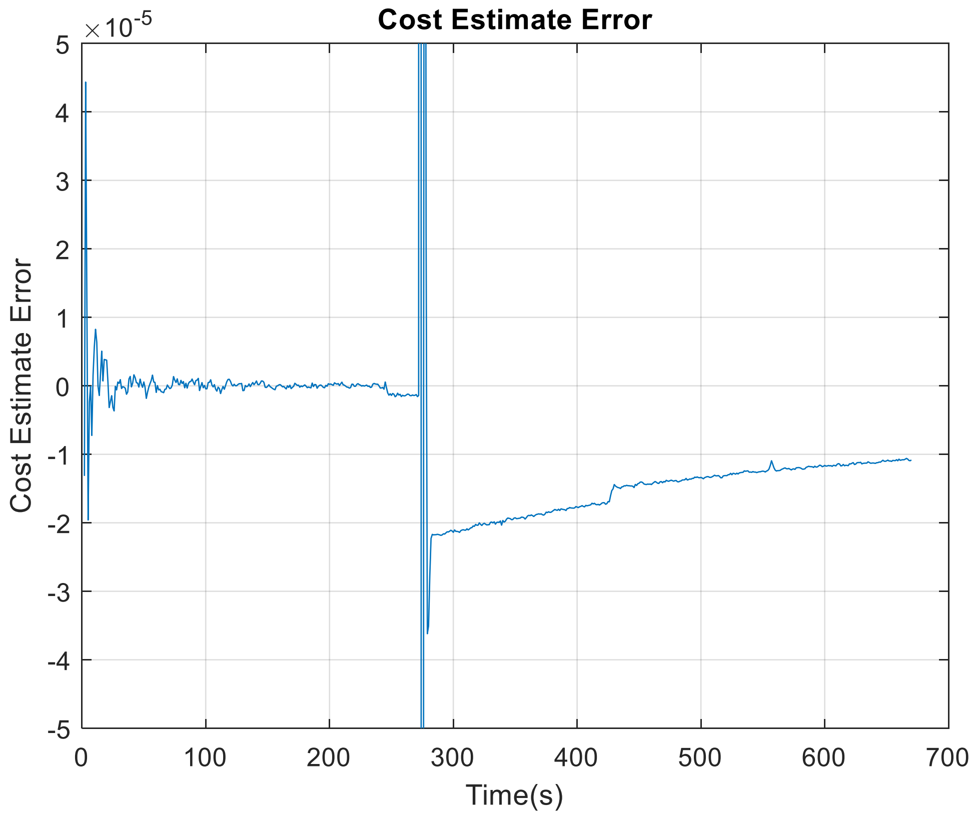


Figure 6-21. Cost Estimate Error (15%)

6.5 CASE 4 – ASSUME 20% RISK TO SCHEDULE AND COST

For Case 4, the program manager has assumed a 20% risk tolerance for both schedule and cost. In order to do so, we must first set out MSM values must be set as follows:

$$\Pi = \begin{bmatrix} .80 & .20 \\ .02 & .98 \end{bmatrix}. \quad (6\text{-}4)$$

The first row of the MSM will reflect the program manager's desire for 20% risk tolerance. Hence, the first row and first column value will be .80. This means that at each time, t, there is an 80% chance that the first CV filter will accurately predict future program schedule and cost. When time, t, occurs in which the future estimate is less than 80% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter.

The number of times that the model switches provides an indication to the program managers that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 20% of the planned schedule. See **Figure 6-13**. This will give the program manager a visual representation of risk.

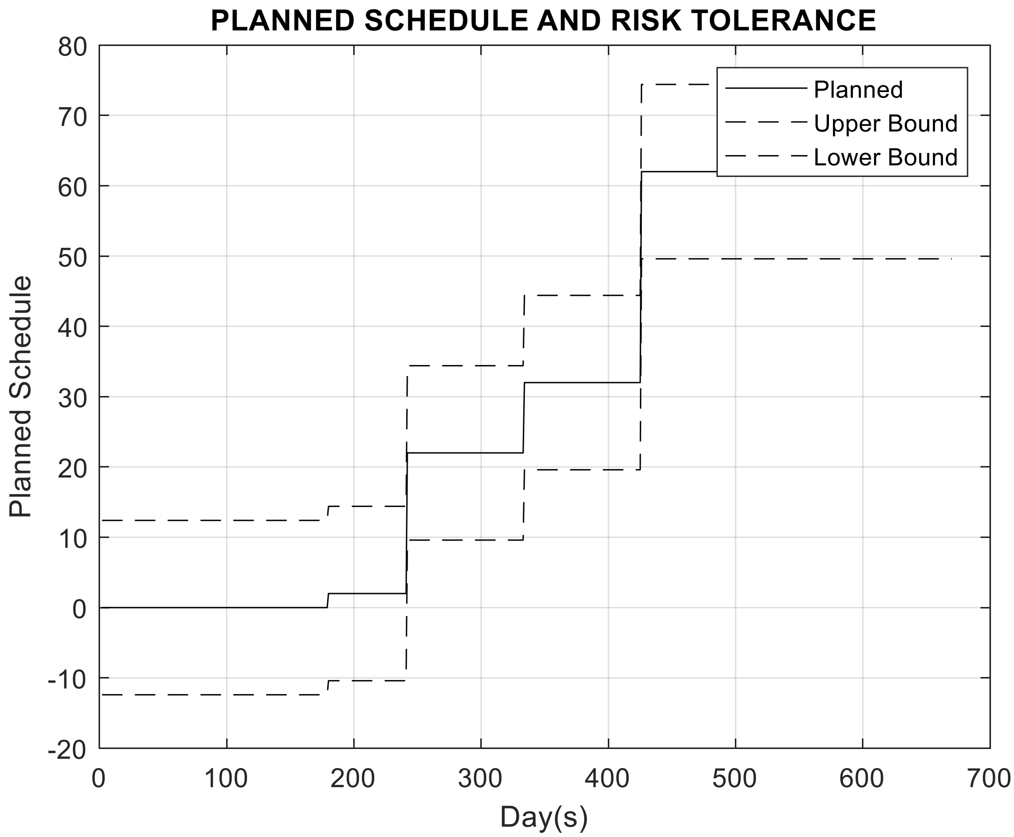


Figure 6-13. Planned Schedule with 20% Risk Tolerance

After running the IMM, the comparison between the IMM and the planned schedule should be examined. It is shown that the IMM does track the planned program schedule quite accurately. For example, in **Figure 6-14** at day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2.

Now examine day 240 in **Figure 6-14**. The planned delivery is 22, but the actual delivery remains at 2. Hence, the program has not accelerated but is decelerating. It can be observed that the actual value of day 290 is less than the lower bound; therefore, the program has assumed more than the 20% risk deemed acceptable by the program manager. The actual value of day 340 is outside the lower bound. The program has accelerated, and it continues to assume more than the 20% risk deemed acceptable by the program manager. Finally, day 450 indicates that the program has fallen behind schedule again and has assumed more than 20% risk.

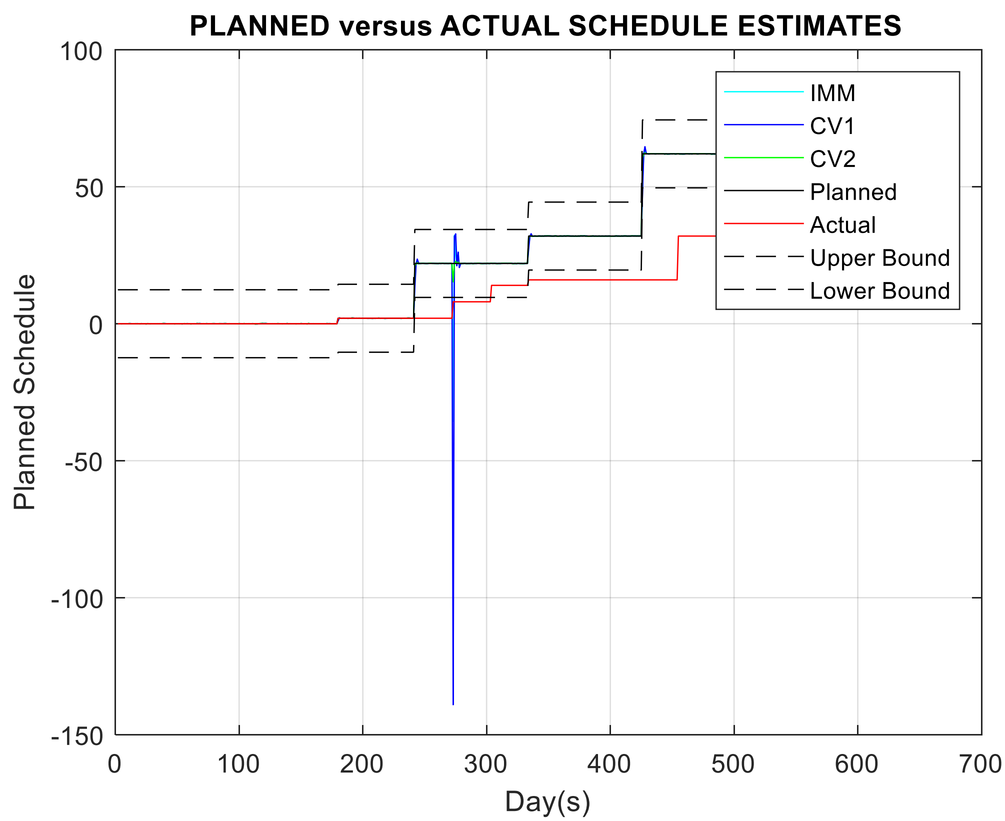


Figure 6-14. Planned versus Actual Program Schedule Assuming 20% Risk

Figure 6-15 captures the planned versus actual cost estimate. Additionally, a 20% risk tolerance is added to the graph by computing ± 20% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost should be examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-15**, around day 20, the planned cost indicates a slight jump in cost. However, the program's actual cost is consistent with the planned cost. It can be observed the program has not assumed more than the 20% risk deemed acceptable by the program manager relative to program cost.

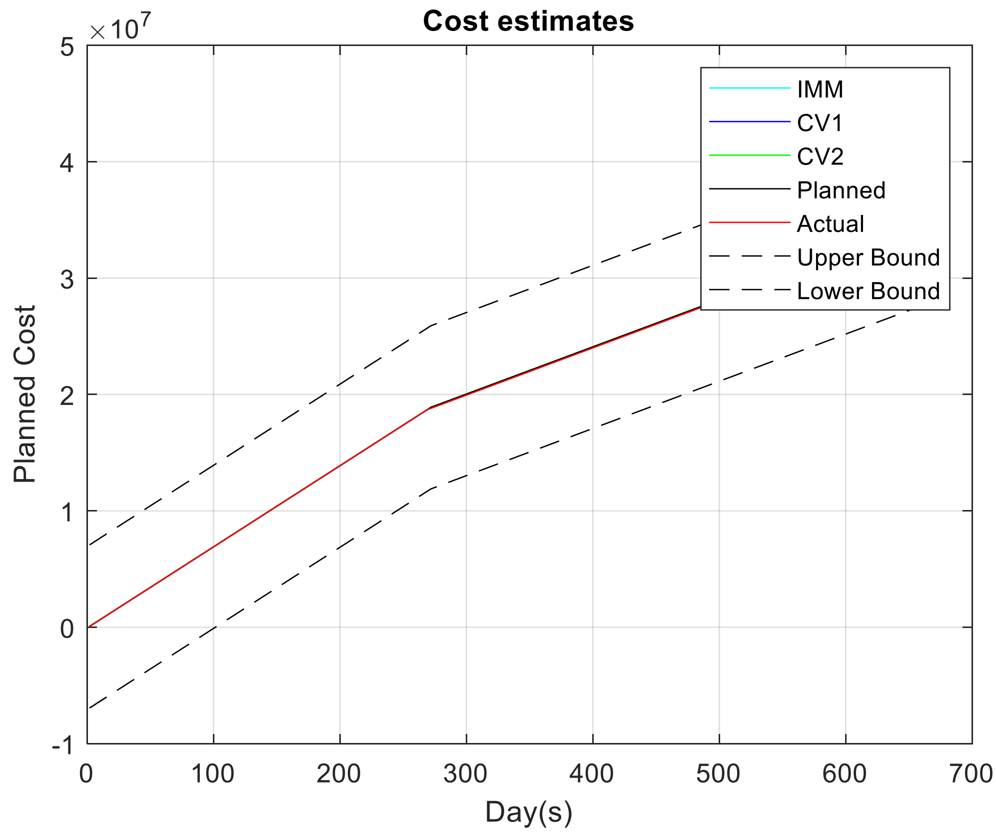


Figure 6-15. Planned versus Actual Program Cost Estimates Assuming 20% Risk

Figure 6-16 shows the switching probabilities of the IMM based on the planned program schedule and cost. It can be observed that the IMM picks up the program planned schedule and cost changes consistently on program days of 190, 240, 272, 340 and 450.

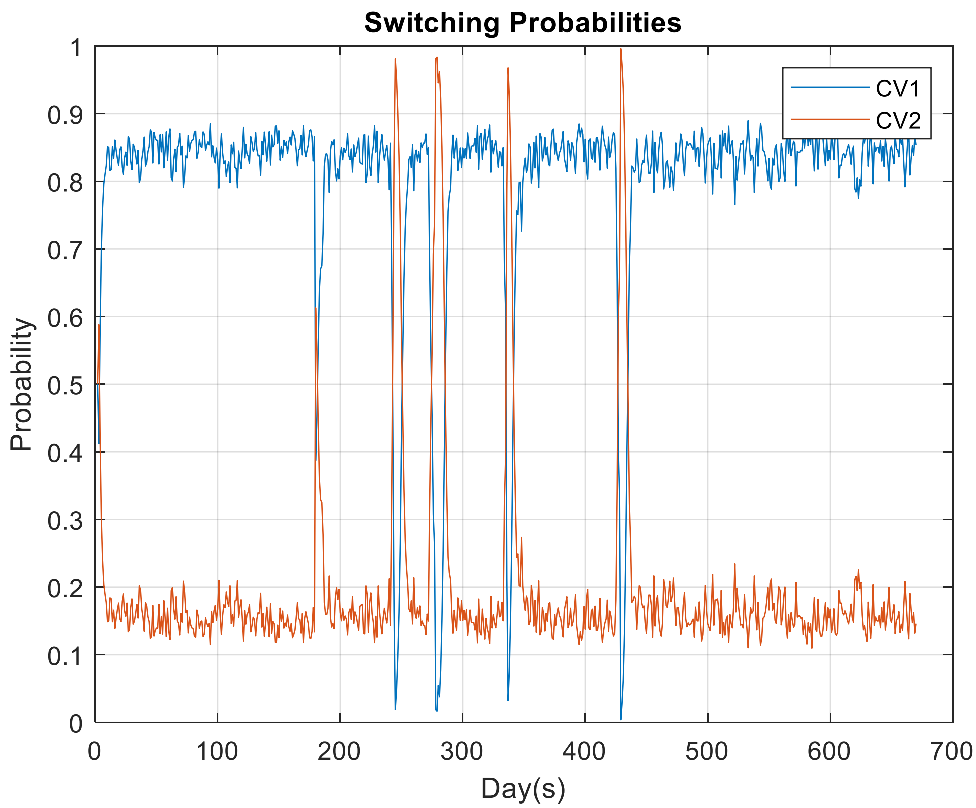


Figure 6-16. Interacting Multiple Model Switching Probabilities Assuming 20% Risk

The IMM filter did identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance. **Table 6-4** is a summary of the points in which the IMM detected a change in the planned program schedule and cost. At day 190, the program planned to deliver 2 systems and did deliver 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained 2.

By examining the program managers risk tolerance in **Figure 6-14**, it is clear that, the risk tolerance exceeded 20%. On day 272, the planned versus actual cost differences did not exceed the 20% risk tolerance. On day 340, the program planned delivery of 32 systems. The actual program delivery was accelerated to 16 systems and remained outside the program manager's risk tolerance level. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems. **Figure 6-14**, again, shows that the risk tolerance has been exceeded on day 420.

20%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	N	Y	N	N

Table 6-4. Switching Probability Summary for 20% Risk Tolerance

By examining **Figure 6-17**, the schedule estimate errors produced by the IMM coincide with the planned program changes on days: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-18**.

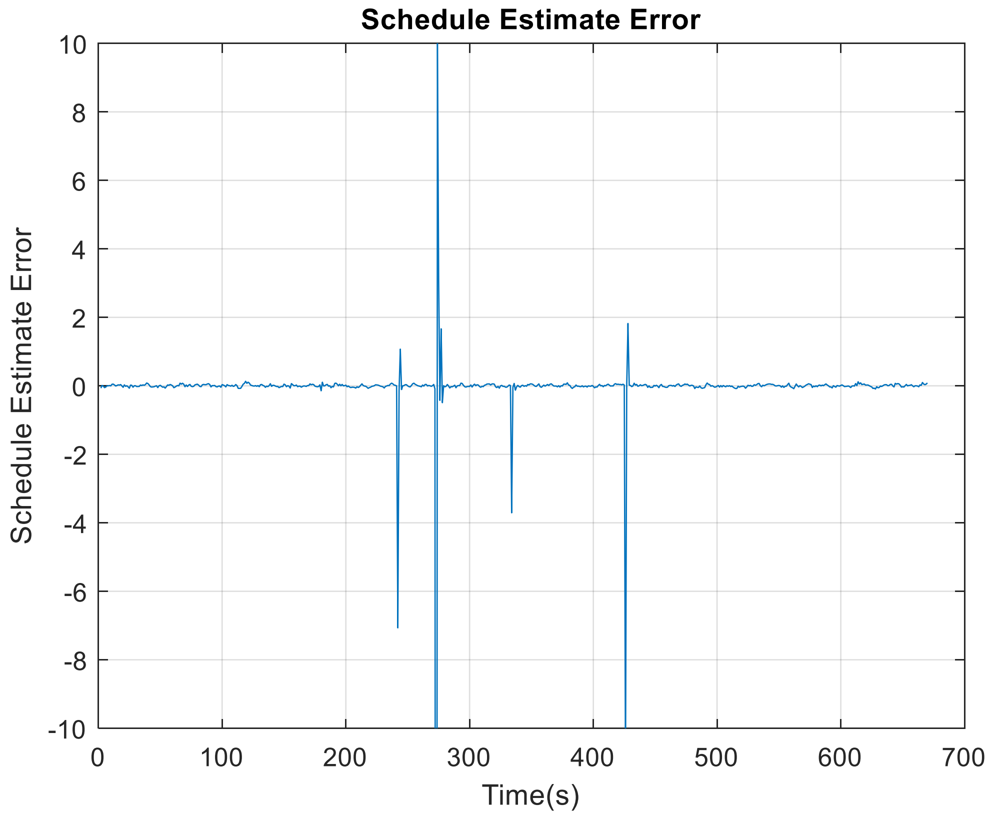


Figure 6-17. Schedule Estimate Error (20%)

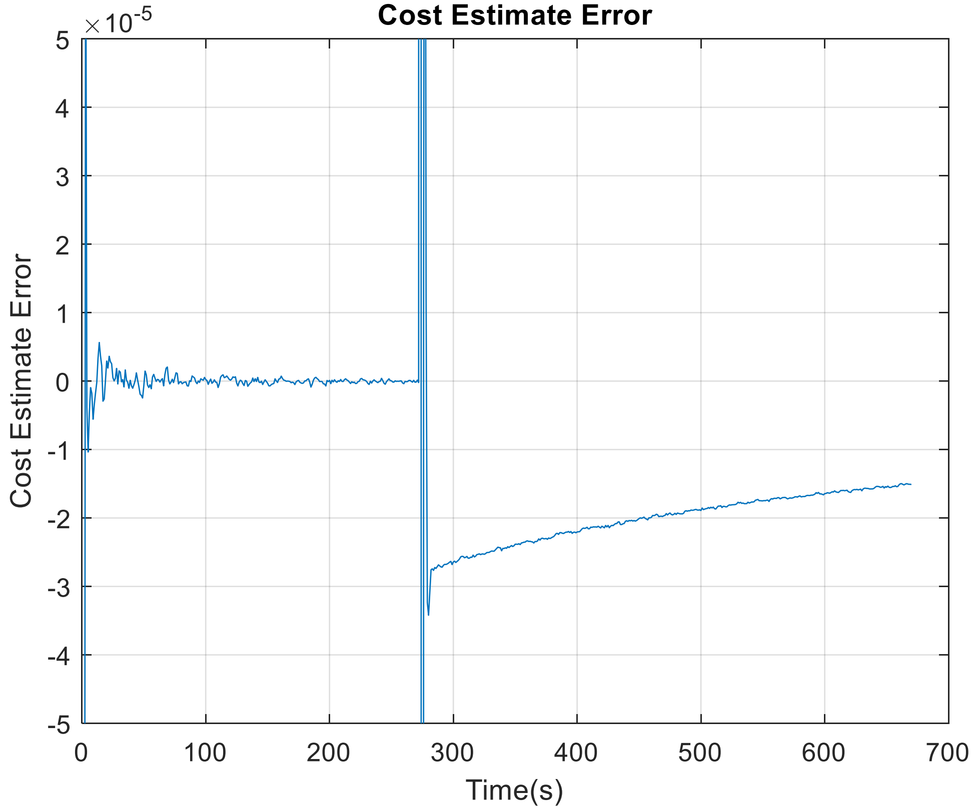


Figure 6-18. Cost Estimate Error (20%)

6.6 CASE 5 – ASSUME 25% RISK TO SCHEDULE AND COST

For Case 5, the program manager has assumed a 25% risk tolerance for both schedule and cost. In order to do so, we must first set MSM values as follows:

$$\Pi = \begin{bmatrix} .75 & .25 \\ .02 & .98 \end{bmatrix}. \tag{6-5}$$

The first row of the MSM will reflect the program manager's desire for 25% risk tolerance. Hence, the first row and first column value will be .75. This means that at each time, t, there is a 75% chance that the first CV filter will accurately predict future program schedule and cost. When time, t, occurs in which the future estimate is less than 75% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter.

The number of times that the model switches provides an indication to the program manager that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 25% of the planned schedule. See **Figure 6-19**. This will give the program manager a visual representation of risk.

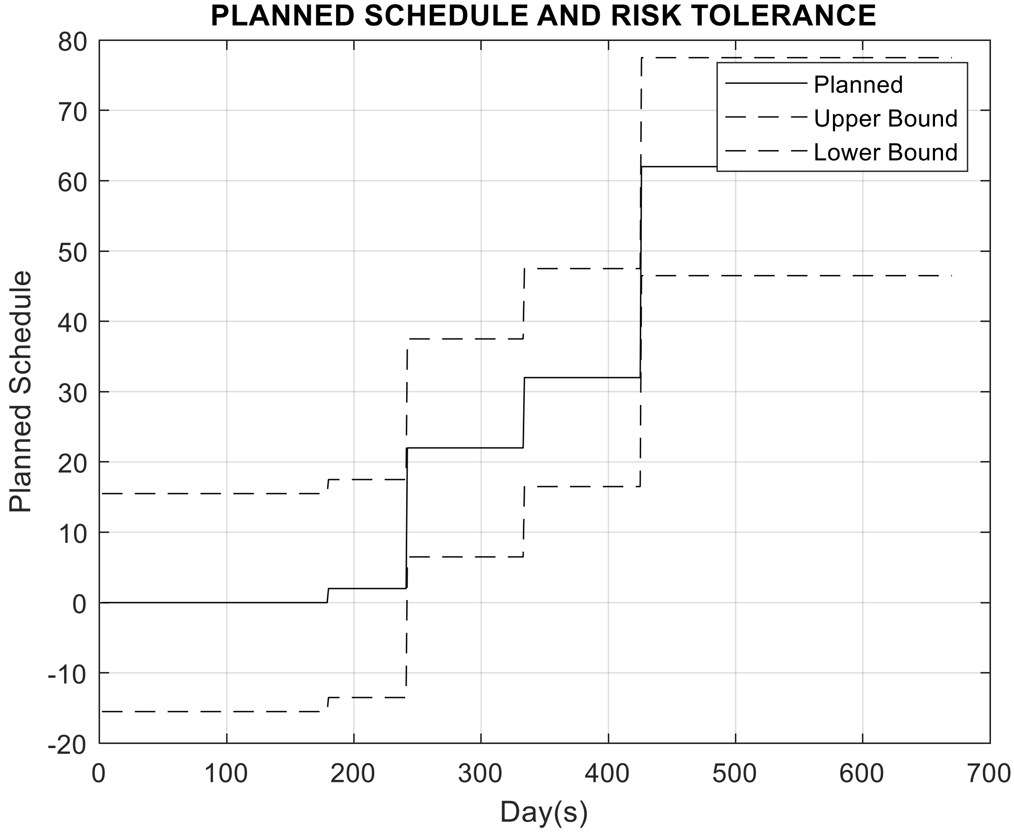


Figure 6-19. Planned Schedule with 25% Risk Tolerance

After running the IMM, the comparison between the IMM and the planned schedule should be examined. It is shown that the IMM does track the planned program schedule quite accurately. For example, in **Figure 6-20** at day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2.

Now examine day 240 in **Figure 6-20**. The planned delivery is 22, but the actual delivery remains at 2. Hence, the program has not accelerated but is decelerating. It can be observed that the actual value of day 290 is less than the lower bound; therefore, the program has assumed more than the 25% risk deemed acceptable by the program manager. Alternatively, the actual value of day 340 is within the lower bound. Therefore the program has accelerated and is assumed less than the 25% risk deemed acceptable by the program manager. Finally, day 450 indicates that the program has fallen behind schedule again and has assumed more than 25% risk.

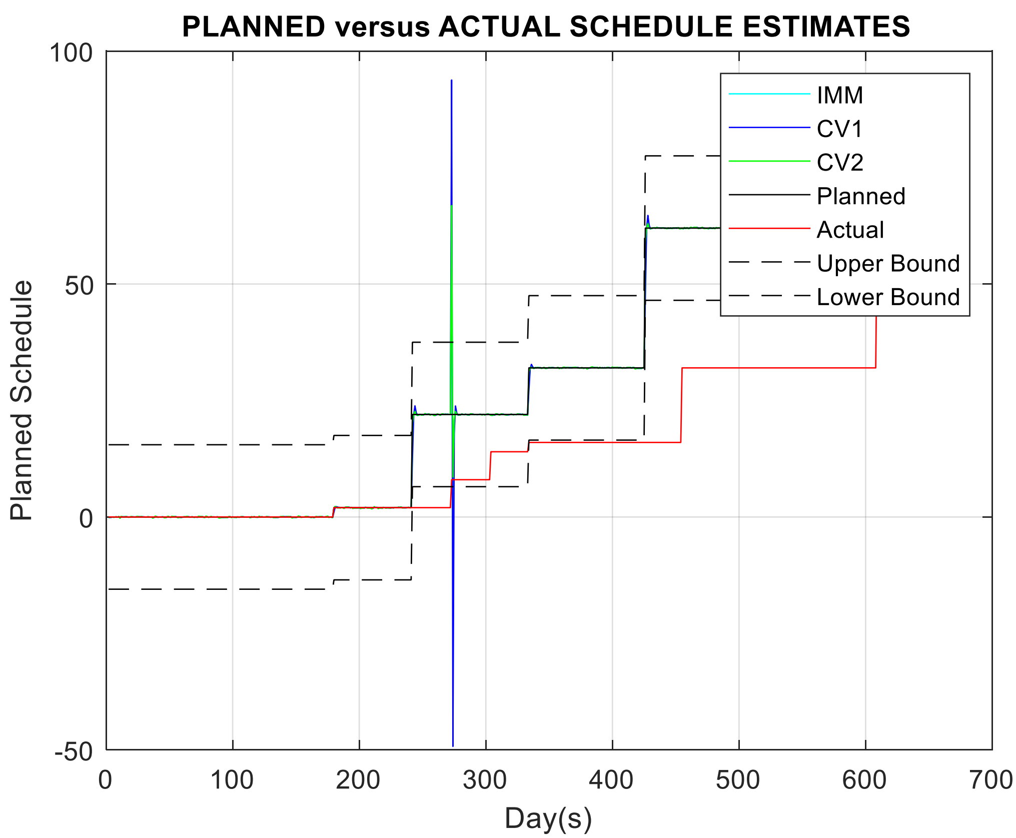


Figure 6-20. Planned versus Actual Program Schedule Estimates Assuming 25% Risk

Figure 6- captures the planned versus actual cost estimate. Additionally, a 25% risk tolerance is added to the graph by computing ± 25% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost should be examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-**, around day 272, the planned cost indicates a slight jump in cost. However, the program actual cost is consistent with the planned cost.

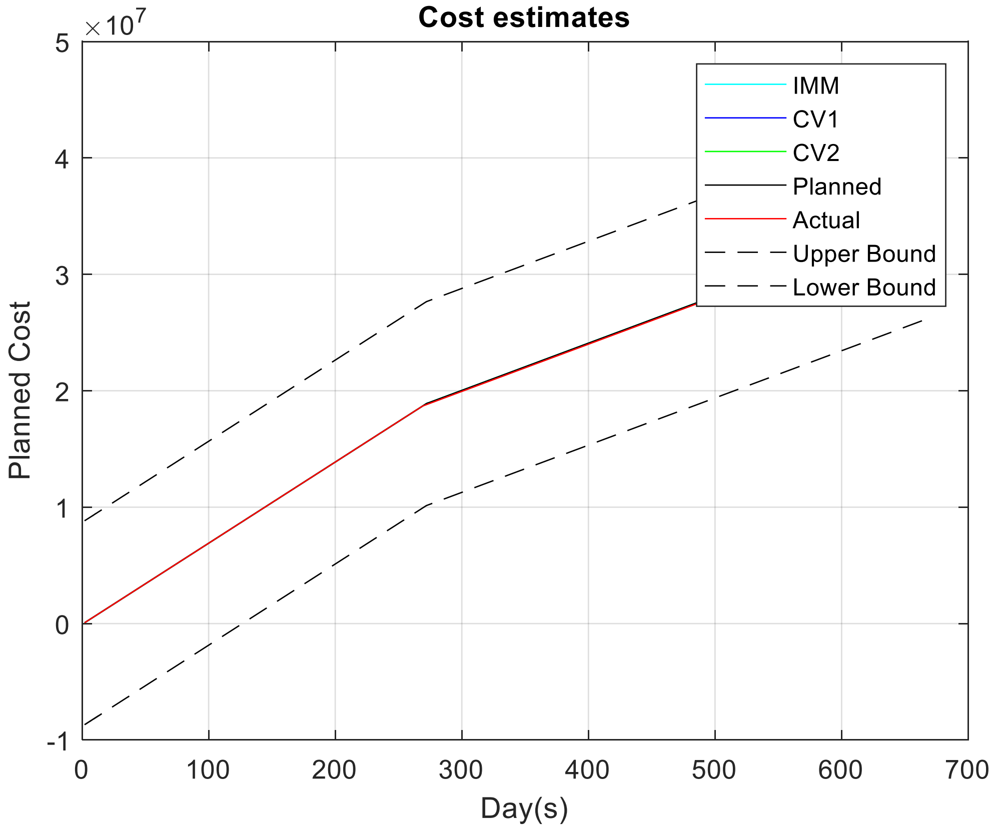


Figure 6-30. Planned versus Actual Program Cost Estimates Assuming 25% Risk

Figure 6-21 shows the switching probabilities of the IMM based on the planned program schedule and cost. It can be observed that the IMM picks up the program planned schedule and cost changes consistently on program days of 190, 240, 272, 340 and 450.

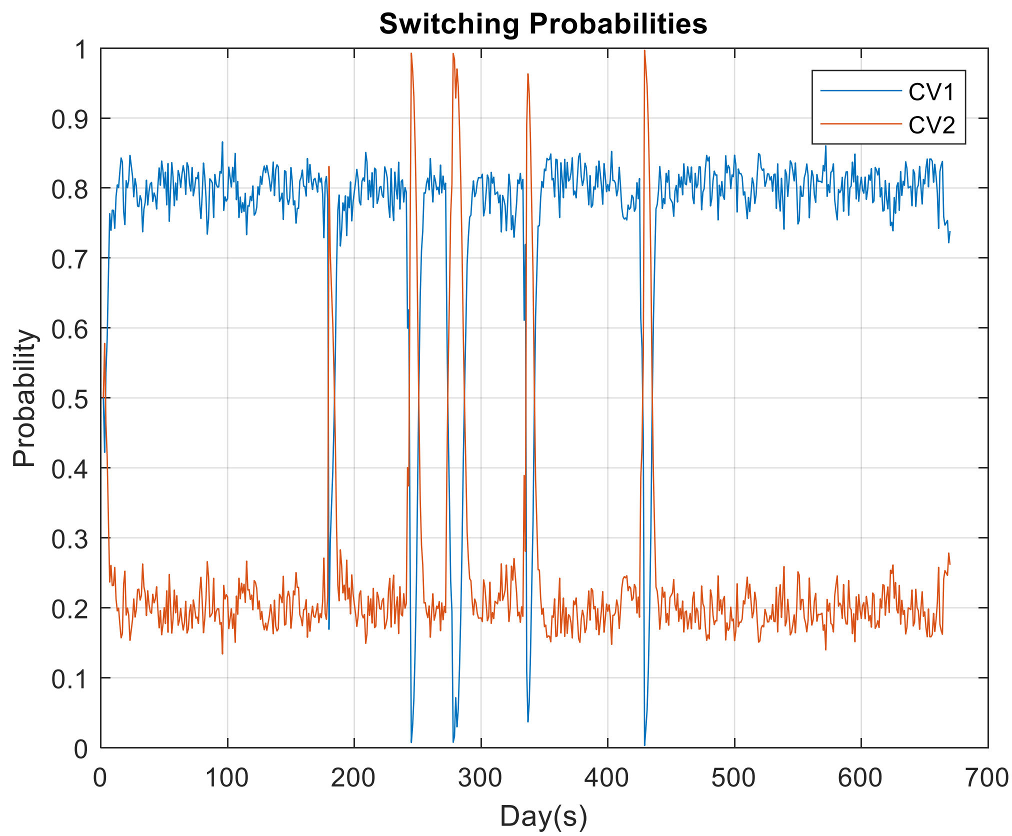


Figure 6-21. Interacting Multiple Model Switching Probabilities Assuming 25% Risk

It can be deduced that the IMM filter did identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance. **Table 6-5** is a summary of the points in which the IMM detected a change in the planned program schedule and cost. At day 190, the program planned to deliver 2 systems and did deliver 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained 2.

By examining the program managers risk tolerance in **Figure 6-20**, it is clear that the risk tolerance exceeded 25%. On day 272, the planned versus actual cost differences did not exceed the 25% risk tolerance. On day 240, the program planned delivery of 32 systems. The actual program delivery was accelerated to 16 systems and was outside the program manager's risk tolerance level. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems. **Figure 6-20**, again, shows that the risk tolerance has been exceeded on day 420.

25%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	N	Y	N	N

Table 6-5. Switching Probability Summary for 25% Risk Tolerance

By examining **Figure 6-22**, the schedule estimate errors produced by the IMM coincide with the planned program changes on days: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-23**.

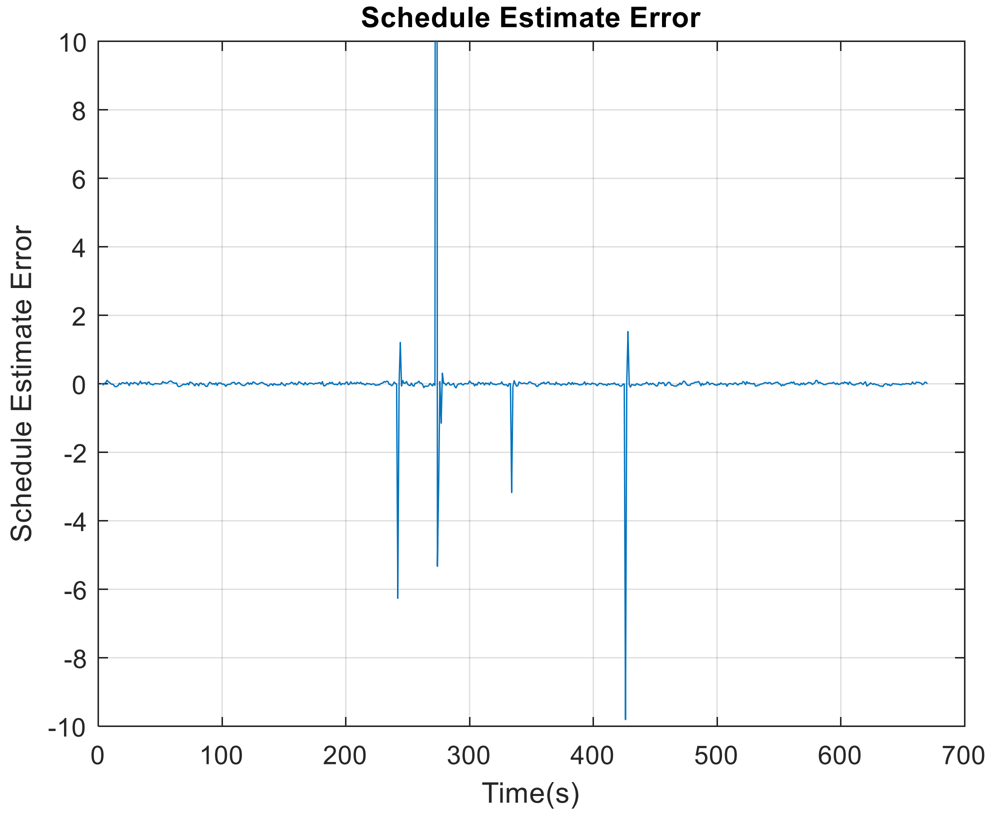


Figure 6-22. Schedule Estimate Error (25%)

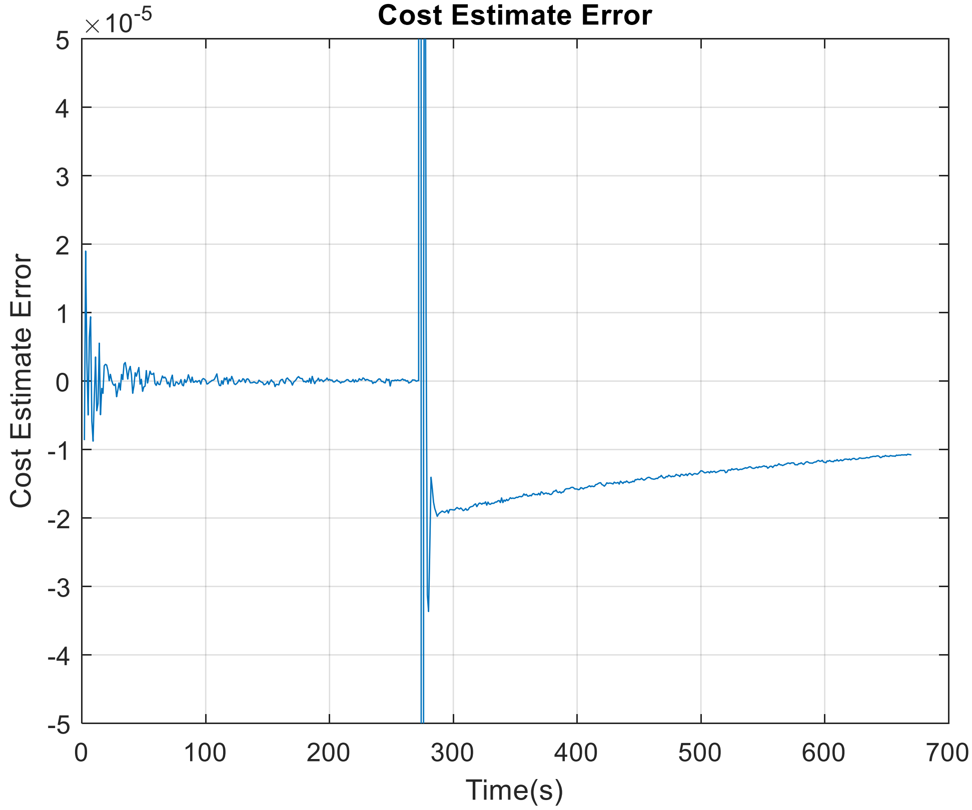


Figure 6-23. Cost Estimate Error (25%)

6.7 CASE 6 – ASSUME 50% RISK TO SCHEDULE AND COST

For Case 6, the program manager has assumed a 50% risk tolerance for both schedule and cost. In order to do so, we must first set MSM values as follows:

$$\Pi = \begin{bmatrix} .50 & .50 \\ .02 & .98 \end{bmatrix}. \qquad (6\text{-}6)$$

The first row of the MSM will reflect the program manager's desire for 50% risk tolerance. Hence, the first row and first column value will be .50. This means that at each time, t, there is a 50% chance that the first CV filter will accurately predict future program schedule and cost. When time, t, occurs in which the future estimate is less than 50% accurate, the model will switch to the second CV filter. The second CV filter will continue to predict the schedule and cost at a 98% accuracy rate. If the estimate is less than 98% accurate, the model will switch back to the first CV filter.

The number of times that the model switches provides an indication to the program managers that the predicted future program schedule or cost has exceeded the defined risk tolerance. The risk tolerance is computed by determining 50% of the planned schedule. See **Figure 6-24**. This will give the program manager a visual representation of risk.

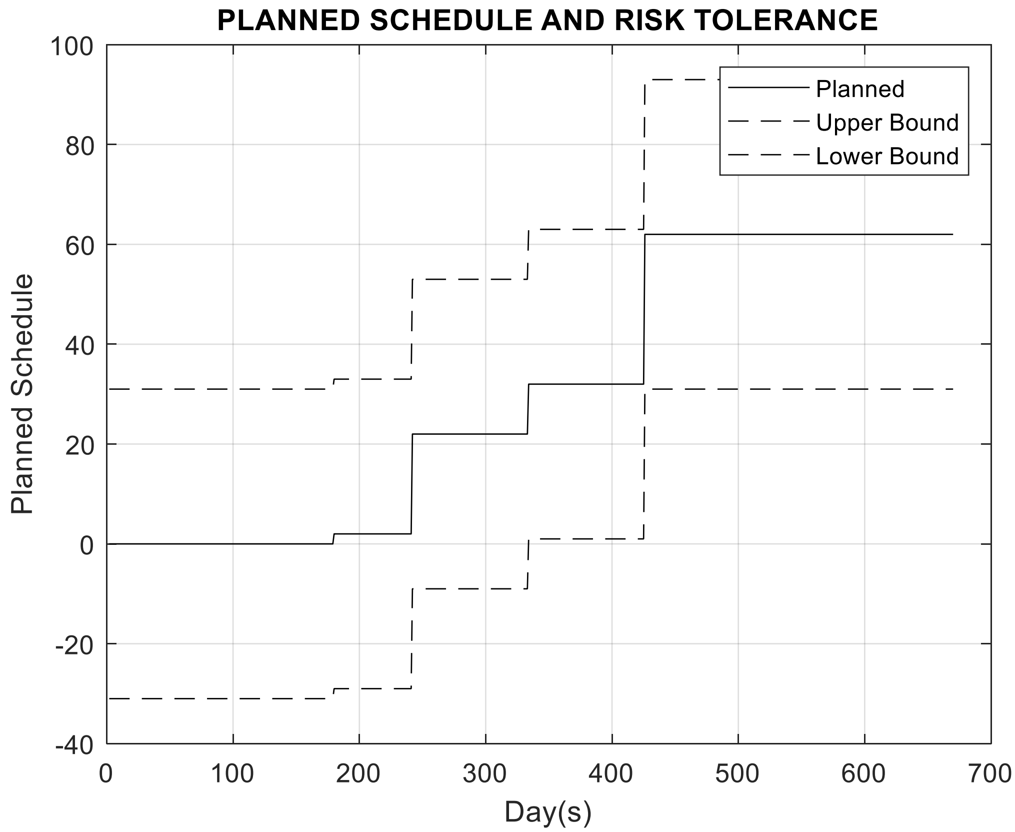


Figure 6-24. Planned Schedule with 50% Risk Tolerance

After running the IMM, the comparison between the IMM and the planned schedule should be examined. It is shown that the IMM does track the planned program schedule quite accurately. For example, in **Figure 6-25** at day 190, the planned schedule indicates 2 systems are planned for delivery. The actual number of deliveries at day 190 is 2.

Now examine day 240 in **Figure 6-25**. The planned delivery is 22, but the actual delivery remains at 2. It can be observed that the actual value of days 290 and 340 are less than the lower bound; therefore, the program has assumed less than the 50% risk. On day 450, the actual value fell below the lower bound of 62.

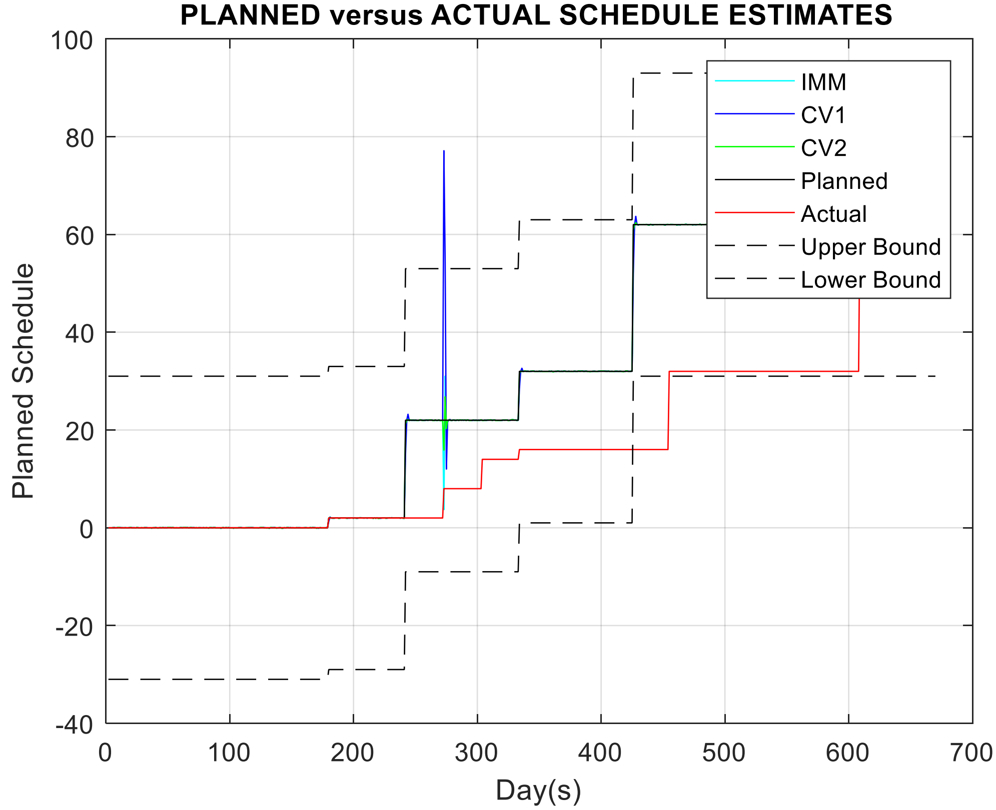


Figure 6-25. Planned versus Actual Program Schedule Estimates Assuming 50% Risk

Figure 6-26 captures the planned versus actual cost estimate. Additionally, a 50% risk tolerance is added to the graph by computing ± 50% of the planned cost. After running the IMM, the comparison between the IMM and the planned cost should be examined. It is shown that the IMM does track the planned program schedule quite accurately.

For example, in **Figure 6-26**, around day 272, the planned cost indicates a slight jump in cost. However, the program's actual cost is consistent with the planned cost.

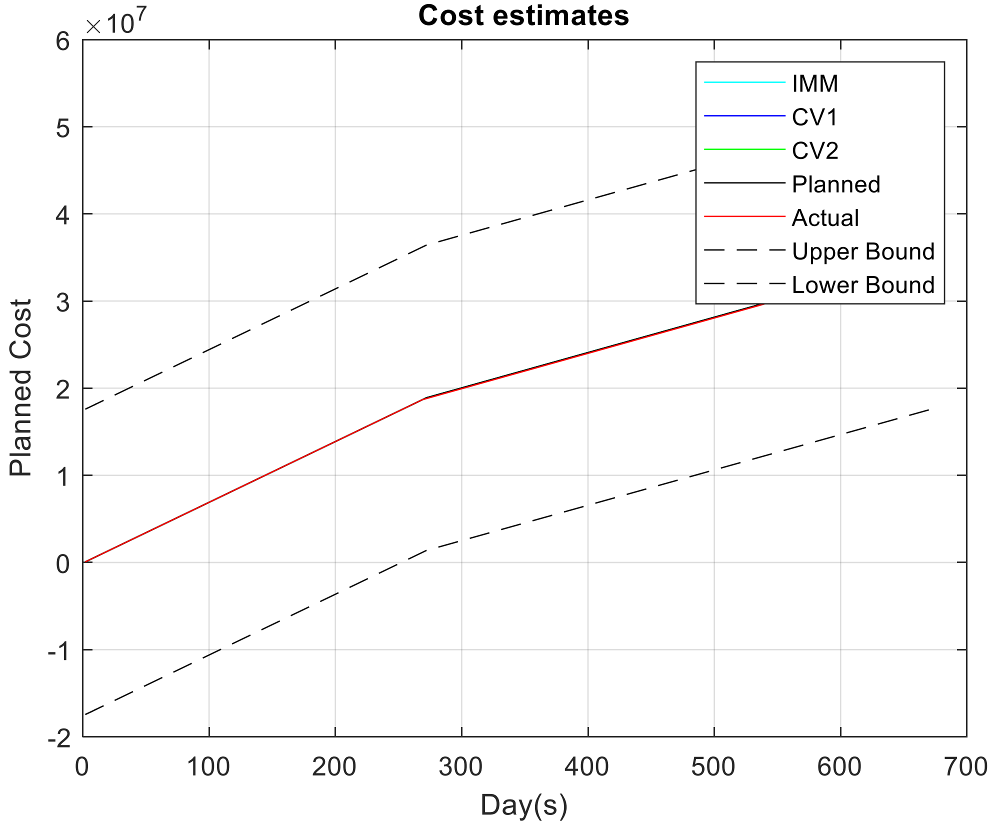


Figure 6-26. Planned versus Actual Program Cost Estimates Assuming 50% Risk

Figure 6-27 shows the switching probabilities of the IMM based on the planned program schedule and cost. It can be observed that the IMM picks up the program planned schedule and cost changes consistently on program days 190, 240, 272, 340 and 450.

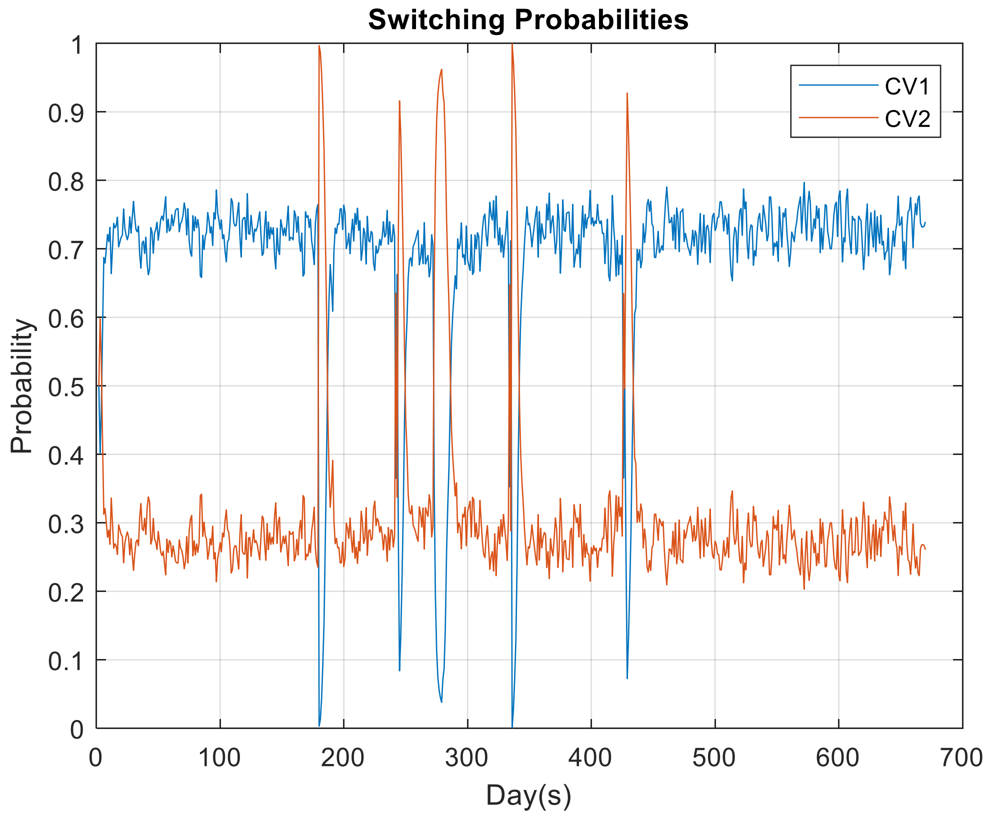


Figure 6-27. Interacting Multiple Model Switching Probabilities Assuming 50% Risk

It can be deduced that the IMM filter did identify the changes in both program schedule and cost which would serve as an indicator to the program manager that some measure of risk has been assumed. It is up to the program manager to determine if that risk is within the established program risk tolerance.

Table 6-6 is a summary of the points in which the IMM detected a change in the planned program schedule and cost. At day 190, the program planned to deliver 2 systems and did deliver 2 systems. At day 240, the program planned to deliver 22 systems. The actual program delivery of systems on day 240 remained at 2.

By examining the program managers risk tolerance in **Figure 6-25**, it is clear that the risk tolerance did not exceed 50%. On day 272, the planned versus actual cost differences did not exceed the 50% risk tolerance. On day 240, the program planned delivery of 32 systems. The actual program delivery was accelerated to 16 systems and was within the program managers risk tolerance level. Finally, on day 420, the program planned delivery of 62 systems but fell short by only delivering 16 systems which is the program manager's risk tolerance of 50%.

50%	Day				
	190	240	272	340	420
Planned	2	22	18894120	32	62
Actual	2	2	18806610	16	16
Upper Bound	Y	Y	Y	Y	Y
Lower Bound	Y	Y	Y	Y	N

Table 6-6. Switching Probability Summary for 50% Risk Tolerance

By examining **Figure 6-28**, the schedule estimate errors produced by the IMM coincide with the planned program changes on day: 240, 272, 340 and 420. The IMM is sensitive enough to pick up the changes in the cost estimate as well. See **Figure 6-29**.

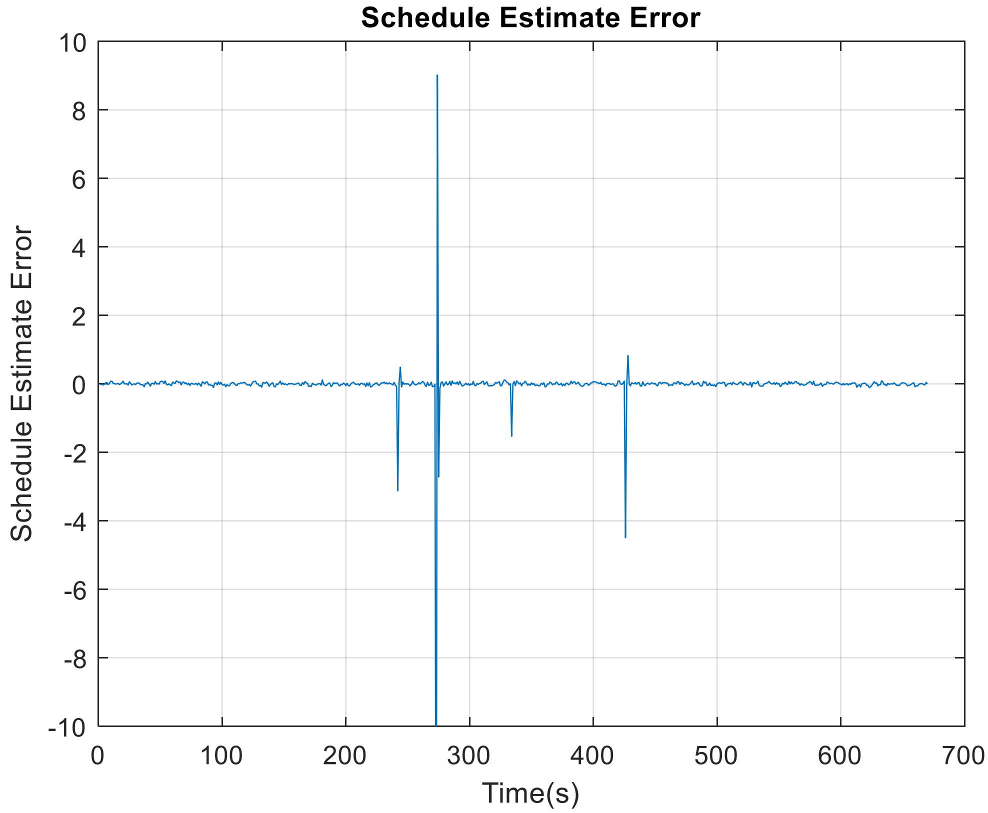


Figure 6-28. Schedule Estimate Error (50%)

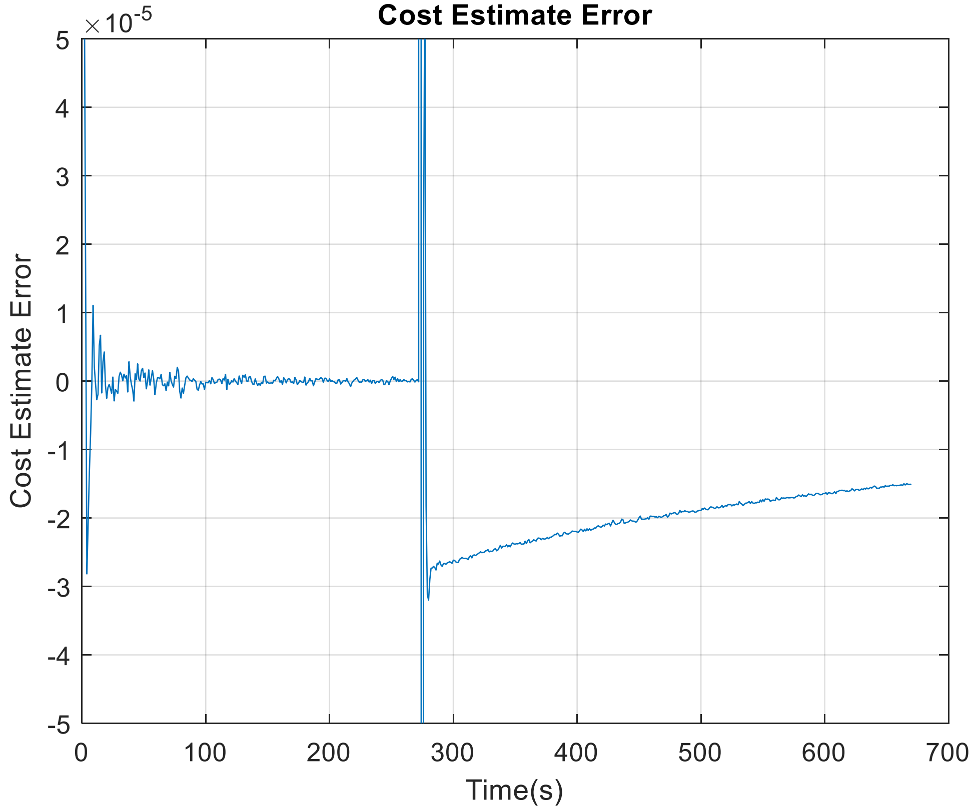


Figure 6-29. Cost Estimate Error (50%)

CHAPTER 7

CONCLUSIONS AND RECOMMENDATIONS

Technology development has increased exponentially. Program managers are pushed to accelerate development. The purpose of this research is to evaluate the modified or redefined use of estimation techniques for target tracking to estimate schedule, cost and performance with a predefined risk tolerance.

This research utilized estimation algorithms used in sensor systems to estimate the current and future state of objects in space to estimate future program cost and schedule in the form of a Kalman filter. More specifically, this research employed two Kalman filters in the form of an Interacting Multiple Model (IMM) to predict the future state of the program. The IMM relies upon a predefined Markov Switching Matrix (MSM) to switch between filters. In this research, the MSM values were used to represent the amount of risk that a program manager was willing to accept.

This research proves that the Interacting Multiple Model can estimate program schedule and cost values and provide an indication of risk based on Markov Switching Matrix (MSM). A deduction process was utilized to build the model, while an induction process was used to analyze the results. As a positivist/empiricist, this research sought to understand real world processes such that controls can be put in place to understand risk associated with acceleration. An assumption about risk tolerance was made such that the risk tolerance was varied between 5% and 50%. The Interacting Multiple Model (IMM) was simulated to estimate future program schedule and cost.

7.1 SUMMARY OF RESULTS

The Interacting Multiple Model (IMM) was run using a Monte Carlo simulation within MATLAB. The results produced by the IMM were compared to the planned program schedule and cost. The risk tolerance was varied between 5% and 50% in order to understand the sensitivity of the model. The Markov Switching Matrix (MSM) was used to vary the risk tolerances. These values within the MSM drive the switching between the two CV filters running in parallel estimating the future state of the program schedule and cost. The CV filters utilize two different process noises to account for uncertainty in the estimates. The CV filter with the closest estimate to the truth is the CV filter favored by the IMM. The MSM is successful in identifying the switching between models that coincides with changes in the program schedule and cost outside the tolerance. The number of times the MSM switches between the high process noise CV filter to the low process noise CV filter and vice versa remains consistent. Of note, the risk tolerance increases; the number of times that the actual schedule and cost exceed either the upper or lower bound decreases (see **Table 7-1** and **Figure 7-1**).

Risk Tolerance	Exceeded Upper or Lower Bound
5%	3
10%	3
15%	3
20%	3
30%	3
50%	1

Table 7-1. Number of Times the Actual Program Schedule or Cost Exceeds Risk Tolerance Upper or Lower Bound

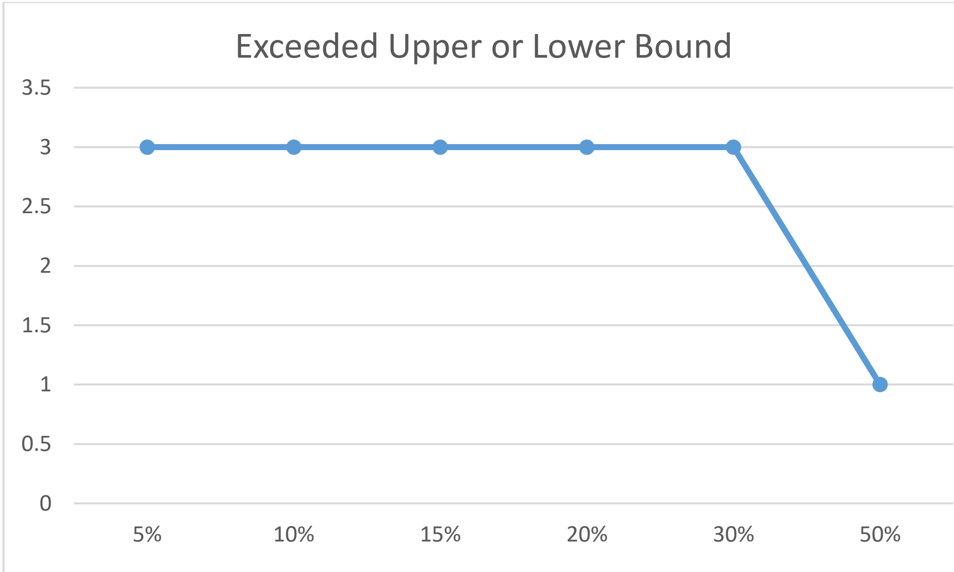


Figure 7-1.   Actual Schedule and Cost versus Risk Tolerance

7.2 RESEARCH CONTRIBUTIONS

This research addresses the following hypotheses. Can the IMM predict future program cost, schedule and performance? It has been shown that the IMM can in fact predict the future state of a program's cost and schedule. Due to data limitations, performance is left to future research. The advantage of the IMM is seeded in the fact that Kalman filters require a motion model to aid in prediction of future state. The selection of two constant velocity motion models with differing process noise allows for better estimation of program future state. Additional motion models, such as the Constant Acceleration or Constant Turning Motion models are left to future research efforts.

Secondly, can the Markov Switching Matrix (MSM) within an Interacting Multiple Model (IMM) predict program risk using upper and lower bounds? It has been demonstrated that the

IMM does an excellent job at weighting each motion model to most accurately estimate program future state. The Markov Switching Matrix (MSM) within the Interacting Multiple Model (IMM) was shown to assist in the determination of program risk by providing an indication based on model switching where risk might be incurred by the program manager. By adding the additional risk tolerance upper and lower bounds, the MSM provides the program manager with indications of when the program might be exceeding the program managers pre-determined level of risk.

The concept that risk can outweigh acceleration is true (i.e. it does not matter how many resources are added to accelerate a program, acceleration is not always the end result). The program identified in this research was attempting to be accelerated. Around day 240, it was clear that the program was decelerating. The program tried to rebound but again failed to accelerate on day 450. By evaluating the risk associated with considering the program to be accelerating, the risk tolerance indicating acceleration was 50%. Hence, the program manager was accepting a 50% risk tolerance in order to actually be considered to be accelerating.

Additionally, with more detailed program data available, the approach provides the potential to perform forensic analysis on programs to determine which methods of acceleration have proven successful or caused significant delay in program development. Finally, the IMM provides the ability for the program manager to perform situational awareness at a high level and alternatively to inform decision making at the project level.

7.3 FUTURE RESEARCH

Due to data limitations, second and third order derivatives are left for future research. Additional data regarding simultaneous tasks/projects executing within a program provide an opportunity to evaluate further the effects of acceleration. Program managers will also have the ability to consider the data from a high-level perspective (i.e. situational awareness) or from a very detailed level in order inform decision making.

Not all programs are the same. Program development takes many sizes and shapes. Due to the IMM being seeded by various motion models, the opportunity to closely match the program schedule, cost and performance profile is within reason. The use of other motion models, such as the Constant Acceleration, Constant Jerk, or Constant Turning motion models, may provide better estimates if the program has very sharp increases in cost, schedule or performance.

The IMM also allows for more than one motion model to be employed at once. In this research, two Constant Velocity motion models were used to seed the Kalman filters. Additional Kalman filters can be employed using a number of different motion models in order to accurately estimate program schedule, cost and performance. The process noise values can also be varied widely and are based on trial and error. Simulations are necessary to establish process noise values that provide meaningful outcomes.

Finally, the method in which the error bounds are derived is left to future research. The focus in this research was on the IMM and its associate MSM. There are other methods to determine upper and lower bounds for risk that could have been considered.

REFERENCES

al., B. e. (1994). *USA Patent No. 5325098.*

all, A. e. (1993). *USA Patent No. 5214433.*

Allchin, D. (2001). Error Types. *Perspectives on Science, 9*(1), 38-58.

Attarzadeh, I. O. (2008). Project Management Practices: The Criteria for Success or Failure. *Communications of the IBIMA*, 234-241.

Axelrod, R. (2005). Advancing the Art of Simulation in the Social Sciences. In J.-P. Rennard (Ed.), *Handbook of Research on Nature Inspired Computing for Economy and Management.* Hershey, PA: Idea Group.

Babu, A., & Suresh, N. (1994). Theory and Methodology: Project Management with time, cost and quality considerations. *European Journal of Operational Research, 88*, 320-327.

Ballesteros-Perez, P. (2017). Modeling the boundaries of project fast tracking. *Automation in Construction*, 231-241.

Bar-Shalom, Y., Rong Li, X., & Kirubarajan, T. (2001). *Estimation with Applications to Tracking and Navigation.* Wiley-Interscience.

Beleiu, I., Crisan, E., & Nistor, R. (n.d.). Main Factors Influencing Project Success. *Interdisciplinary Management Research XI*, 59-72.

Bondugula, S. (2009). *Optimal control of projects based on Kalman fitler apprach for tracking and forecasting the project performanc.* Master's Thesis, Texas A&M University. Retrieved from http://hdl.handle.net/1969.1/ETD-TAMU-2009-05-263

Bratley, P., Fox, B., & Schrage, L. (1987). *A Guide to Simulation* (Second Edition ed.). New York: Springer-Verlag.

Byung, C. K. (2007). *Forecasting Project Progress and earling Warning of Project Overruns with Probabilistic Methods.* Doctorial Dissertation, Texas A&M University. Retrieved from https://oaktrust.library.tamu.edu

Chairman of the Joint Chiefs of Staff. (2009). *Chairman of the Joint Chiefs of Staff Instruction 317.1G Joint Capabilities Integration And Development System.* Chairman of the Joint Chiefs of Staff, J-8. Chairman of the Joint Chiefs of Staff J-8.

Clancy, T. (1995). *The Standish Group Report.*

Copi, I., Cohen, C., & Flage, D. (2006). *Essentials of Logic* (2nd ed.). Upper Saddle River, NJ: Pearson Eduction.

Crawford, L. E. (2004). Mapping the links between project managment and systems. *Proceedings of IRNOP, 6*.

Davis, J. P., Eisenhardt, K. M., & Bingham, C. B. (2007). Developing theory through simulation methods. *Academy of Management Review, 32*(2), 480-499.

Deductive and Inductive Arguments. (n.d.). Retrieved from Lane Community College: https://www.lanecc.edu/sites/default/files/trio/deductive_and_inductive_arguments.pdf

Dehghan, R., & Ruwanpura, J. (2011). The Mechanism of Design Activity Overlapping Construction Projects and the Time-Cost Tradeoff Function. *The Twelfth East Asia-Pacific Conference on Structural Engineering and Construction. 14*, pp. 1959-1965. Procedia Engineering.

Development, R. a. (2012, October 19). *https://www.wonderlic.com/blog/validity-and-reliability*.

Dooley, K. (2002). Simulation research methods. In K. Dooley, & J. Baum (Ed.), *Companion to Organizations* (pp. 829-848). London: Blackwell.

Firesmith, D. (2015). Four Types of Shift-Left Testing.

Forrester, J. (1961). *Industrial Dynamics.* Cambridge, MA: Productivity Press.

Gerk, J., & Qassim, R. (2008). Project Acceleration via Activity Crashing, Overlapping and Substitution. *IEEE Transactions on Engineering Management, 55*(4), 580-587.

Gibbs, M. N. (2000, November). Variational Gaussian Process Classifiers. *IEEE Transactions on Neural Networks, 11*(6), 1458-1464.

Gibbs, S. (2014, Dec 05). *The Guardian*. Retrieved from www.theguardian.com: https://www.theguardian.com/technology/2014/dec/05/steve-wozniak-apple-starting-in-a-garage-is-a-myth

Gilbert, N. (2005). *Chapter 2: Simulation as a Method.* Retrieved from Electronic Pubs: epubs.surrey.ac.uk/1609/1/fulltext.pdf

Gilbert, N., & Troitzsch, K. G. (2005). *Simulation for the social scientist* (2nd ed.). Maidenhead: Open University Press.

Golafshani, N. (2003, December). Understanding Reliability and Validity in Qualitative Research. *The Qualitative Report, 8*(4), 597-607. Retrieved from http://www.nova.edu/sss/QR/QR8-4/golafshani.pdf

Gren, L. (2017). A fourth explanation to Brooks' Law - The aspect of group developmental psychology. *IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, (pp. 86-89).

Hagel, C. (2014, November 15). The Defense Innovation Initiative Memorandum .

Harrison, J. R. (n.d.). *The Concept of Simulation in Organizational Research.* Dallas: University of Texas at Dallas.

Harrison, J. R. (n.d.). The Concept of Simulation in Organizational Research. *Univeristy of Texas at Dallas, School of Management*. RIchardson, TX, USA.

Heale, R., & Twycross, A. (2015). Validity and reliability in quantitative studies. *Evidence Based Nursing*, 66-67.

Hsai, P., Hsu, C., & Kung, D. (n.d.). Brooks' Law Revisited: A Systems Dynamics Approach. Arlington, TX: Computer Science Engineering Department, university of Texas Arlington.

Hsai, P., Hsu, C., & Kung, D. (n.d.). Brooks' Law Revisited: A Systems Dynamics Approach. Arlington, TX: Computer Science Engineering Department, University of Texas.

(12 April 2001 (As Amended Through 17 October 2007)). *Joint Publication 1-02, Department of Defense Dictionary of Military and Associated Terms.*

Karagozolgu.N., & Brown, W. (1993). Time-based management of the new product development process. *Journal of Product Innovation Management, 10*, 204-215.

Kendall, F. (19 September 2014). *Better Buying Power 3.0.* White Paper, Office of Under Secretary of Defense Acquisition, Technology, and Logistics.

Keyton, J. (2002). *Communicating in groups: Building relationships for effectiveness.* New York: McGraw-Hill.

Killion, T. (2012, 10). *Office of Naval Research.* Retrieved 11 24, 2014, from ONR Naval S&T Partnership Conference and ASNE Expo Future Naval Capabilities (FNC) Portfolio: http://www.onr.navy.mil/conference-event-onr/science-technology-partnership/~/media/Files/Conferences/Science-Technology-Partnership/2012/Oct22/Killion-FNCs-Oct22.ashx

Kimberlin, C. L., & Winterstein, A. (2008, Dec 1). Validity and reliability of measurement instruments used in research. *AM J. Health-Sys Pharm, 65*, 22276-22284.

Kurzweil, R. (2001). The Law of Accelerating Returns.

Langerak, F., & Hultink, E. (2005). The Impact of New Product Development Acceleration Approaches on Speed and Profitability: Lessons for Pioneers and Fast Followers. *IEEE Transactions on Engineering Management, 40*(2), 247-262.

Law, A., & Kelton, D. (1982). *Simulation Modeling and Analysis.* NY: McGraw-Hill.

Le Cren, M. (n.d.). *Top 13 Project Management Methodologies and Styles*. Retrieved from https://blog.azendoo.com/top-project-methodologies-and-styles/

Li, X. a. (2003, October). Survey of Maneuvering Target Tracking, Part 1: Dynamic Models. *IEEE Transactions on Aerospace and Electronic Systems, 39*(4).

Mabus, H. R. (2015, January 22). Task Force Innovation. *Memorandum for Undersecretary of the Navy*.

McCabe, D. S.-C. (2005). *USA Patent No. 7030809.*

Meshguides. (n.d.). Retrieved from How can validity and reliabilty be improved?: www.meshguides.org

Mohajan, H. K. (2017). Two Criteria for Good Measurements in Research: Validity and Reliability. *Annals of Spiru Haret University, 17*(3), 58-82.

Morgan, J. (2015, July 6). *Forbes.* Retrieved from The 5 Types of Organizational Structures: Part 1, The Hierarchy: https://www.forbes.com/sites/jacobmorgan/2015/07/06/the-5-types-of-organizational-structures-part-1-the-hierarchy/#5e3fdc42525

Morgan, J. (2015, July 13). *Forbes.* Retrieved from The 5 Type of Organizational Structures: Part 3, Flat Organizations: https://www.forbes.com/sites/jacobmorgan/2015/07/13/the-5-types-of-organizational-structures-part-3-flat-organizations/#29b48a666caa

Munns, A., & Bjeirmi, B. (1996). The role of project management in achieving project success. *International Jounal of Project Management*, 81-87.

Nicoletti, S., & Nicolo, F. (1998). A concurrent enginering decision model: Management of the project activities information flow. *International Journal of Production Economics, 54*, 115-127.

Opelt, K. (2008). Overcoming Brooks' Law. *Agile 2008 Conference* (pp. 208-211). IEEE Computer Society.

Pitch, M., Loch, C., & De Meyer, A. (2002). On Uncertainty, Ambiquity and Complexity in Project Management. *Management Science*, 1008-1023.

Pub. L. 96-517. (1980, December 12).

Report to the Presidential Commission on the Space Shuttle Challenger Accident. (1986, June 6). Retrieved from NASA History Division: https://history.nasa.gov/index_new.html

Roemer, T., Ahmadi, R., & Wang, R. (2004). Concurrent crashing and overlapping in product development. *Operations Research, 52*(4), 311-326.

Rose, S., Spinks, N., & Canhoto, A. (2015). Management Research; Applying the Principles. 1-6.

Rose, S., Spinks, N., & Canhoto, A. I. (2015). *Management Research: Applying the Principles.*

Rothchild, I. (2011, December 27). *Induction, Deduction, and the Scientific Method*. Retrieved from An Eclectic Overview o fthe Practice of Science: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.2694

Sauser, B., Reilly, R., & Shenhar, A. (2009). Why Projects Fail? How contingency Theory can provide new insights - A comparative analysis of NASA's Mars Climate Orbiter loss. *International Journal of Project Management, 27*, 665-679.

Senge, P. (1990). *The Fifth Discipline.* NY: Doubleday.

Sharif, M. N., & Islam, M. N. (1980). The Weibull distribution as a general model for forecasting technological change. *Technological Forecasting and Social Change, 18*(3), 247-256.

Singer, D., Doerry, N., & Buckley, M. (2009). What is Set Based Design? *Naval Engineers Journal*, 31-43.

Sterman, J., Repenning, N., & Kofman, F. (1997). Unanticipated Side Effects of Successful Quality Programs: Exploring a Paradox of Organizational Improvement. *Management Science*.

Surbhi, S. (2017, March 9). *Difference Between Validity and Reliabilty*. Retrieved from Key Differences: https://keydifferences.com/difference-between-validity-and-reliability

Swink, M. (2003, December). Completing projects on-time: how project acceleration affects new product development. *Journal of Engineering and Technology Management, 20*(4), 319-344.

Tuckerman, B. W. (1993). Developmental sequences in small groups. *Psychological Bulletin, 63*(6), 384.

Ulgen, O. M. (n.d.). *Simulation Methodology -- A Practitioner's Perspective.*

Verner, J. M., Overmeyer, S. P., & McCain, K. (1999). In the 25 years since The Mystical Man-Month what have we learned about project management? *Information and Software Technology, 41*, 1021-1026.

Wallace, S. E., & Little, G. (2013). *Analyzing Fleet Capabilities Using Mission Threads/Kill Chains.* Online publication: American Society of Naval Engineers. Retrieved from https://www.navalengineers.org/ProceedingsDocs/ASNEDay2013/Wallace_Paper.pdf

Weaver, P. (2010). Understanding Programs and Projects: Oh, There's a Difference. *PMI Global Congress Asia Pacific.* Melbourne, Australia.

Wheelan, S. (2009). Group size, group development and group productivity. *Small Group Research, 40*(2), 247-262.

Winter, G. (1998, February 25). *A comparative discussion o fthe notin of validity in qualitative and quantitative research*. Retrieved from The Qualitative Report: http://www.nova.edu/sss/QR/QR4-3/winter.html

Yassine, A., Sreenivas, R., & Zhu, J. (2008). Managing the exchange of information in product development. *European Journal of Operations Research*, *184*, 311-326.

APPENDIX

KALMAN FILTER EQUATIONS

The Kalman filter is an optimal state estimator for single mode systems, provided that an exact motion model for the program dynamics are available. Many experiments have tried to broaden the Kalman filter to provide optimal state estimates for multiple mode systems, i.e. accelerating schedule or cost.

$$P_{k+1|k} = \Phi P_{k|k} \Phi^T + Q \quad \text{(A-1)}$$

$$K_{k+1} = P_{k+1|k} H^T \left[H P_{k+1|k} H^T + R \right]^{-1} \quad \text{(A-2)}$$

$$P_{k+1|k+1} = (I - K_{k+1} H) P_{k+1|k} \quad \text{(A-3)}$$

State Estimate Predict equations

$$\hat{X}_{k+1|k} = \Phi \hat{X}_{k|k} \quad \text{(A-4)}$$

State Estimate Update (with measurement) equations

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \left[Z_{k+1} - H \hat{X}_{k+1|k} \right], \quad \text{(A-5)}$$

where the Gain matrices are solved for by minimizing J, the *Trace* of the Fused covariance matrix P_f

$$J = Trace\{P_f\} = Trace\{(I - K_z H) P_X (I - K_z H)^T + K_z R K_z^T\} \quad \text{(A-6)}$$

$$\frac{\partial}{\partial K_z} Trace\{(I - K_z H) P_X (I - K_z H)^T + K_z R K_z^T\} \quad \text{(A-7)}$$

$$\frac{\partial J}{\partial K_z} = -2(I - K_z H) P_X H^T + 2 K_z R \equiv 0. \quad \text{(A-8)}$$

Solving we get

$$K_z [H P_X H^T + R] = P_X H^T. \quad \text{(A-9)}$$

Optimum Gain Matrix:

$$K_z = P_X H^T [H P_X H^T + R]^{-1}]. \tag{A-10}$$

The above matrix gradient expression was obtained from the following two identities

$$\frac{\partial}{\partial K} Trace\{K P_X K^T\} = 2 K P_X \tag{A-11}$$

$$\frac{\partial}{\partial K} Trace\{(I - K) P_N (I - K)^T\} = -2(I - KH) P_N K^T. \tag{A-12}$$

The optimum estimate is given by

$$\hat{X}_f = \hat{X} + K_z (Z - H\hat{X}) \tag{A-13}$$

where

$$K_z = P_X H^T [H P_X H^T + R]^{-1}]. \tag{A-14}$$

Optimum covariance matrix

$$P_f = (I - K_z H) P_X. \tag{A-15}$$

The covariance matrix associated with $\hat{X}_f$ is given by the following expression (also known as the Josephson Stabilized Form)

$$P_f = (I - K_z H) P_X (I - K_z H)^T + K_z R K_z^T \tag{A-16}$$

This simplifies to

$$P_f = \left\langle I - P_X H^T [H P_X H^T + R]^{-1} H \right\rangle P_X \left\langle I - P_X H^T [H P_X H^T + R]^{-1} H \right\rangle^T + P_X H^T [H P_X H^T + R]^{-1} R [H P_X H^T + R]^{-1} H P_X \tag{A-17}$$

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X \left\langle I - H^T [HP_X H^T + R]^{-1} HP_X \right\rangle \tag{A-18}$$

$$+ P_X H^T [HP_X H^T + R]^{-1} R [HP_X H^T + R]^{-1} HP_X$$

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X \tag{A-19}$$

$$- \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} R [HP_X H^T + R]^{-1} HP_X$$

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X \tag{A-20}$$

$$- P_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} HP_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} R [HP_X H^T + R]^{-1} HP_X \,.$$

The above expression can be written more concisely by factoring the last two lines:

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X \tag{A-21}$$

$$- P_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} [HP_X H^T + R][HP_X H^T + R]^{-1} HP_X$$

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X - P_X H^T [HP_X H^T + R]^{-1} HP_X + P_X H^T [HP_X H^T + R]^{-1} HP_X \tag{A-22}$$

$$P_f = (I - P_X H^T [HP_X H^T + R]^{-1} H) P_X = (I - K_z H) P_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} R [HP_X H^T + R]^{-1} HP_X$$

$$P_f = \left\langle I - P_X H^T [HP_X H^T + R]^{-1} H \right\rangle P_X \tag{A-23}$$

$$- P_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} HP_X H^T [HP_X H^T + R]^{-1} HP_X$$

$$+ P_X H^T [HP_X H^T + R]^{-1} R [HP_X H^T + R]^{-1} HP_X \,.$$

VITA

Amy Sunshine Smith-Carroll is currently a student in the Old Dominion University, School of Engineering and Technology, Department of Engineering Management and Systems Engineering located at 1000 Rollins Hall, Norfolk, VA 23529. She received her Master's degree in Mathematics from Mississippi State University in 1999 and her Bachelor's in Mathematics from Mississippi University for Women in 1997. She is also a graduate of the Federal Executive Institute Leadership in a Democratic Society.

Ms. Smith-Carroll has spent over 20 years in the Department of Defense's acquisition community. Ms. Smith-Carroll is currently serving as the Director for Counter Unmanned Air Systems (C-UAS) programs for the Deputy Assistant Secretary of the Navy for Air and Ground Programs. She is responsible for the coordination and oversight of development and acquisition plans for Navy and U.S. Marine Corps C-UAS efforts.

Ms. Smith-Carroll is actively involved in the target tracking and sensor fusion community. She has chaired numerous national and international conferences, published over 30 technical papers and 4 publications in the area of target tracking and sensor fusion. Ms. Smith-Carroll also developed a multiple model filter switching logic design that does not employ a Markov Switching Matrix, which was patented under Navy Case No. 84317.

Ms. Smith-Carroll has received the Navy Meritorious Civilian Service Award, NSWC Indian Head Excellence Award, and numerous Letters of Commendation. She is also a member of the Acquisition Professional Community and DAWIA certified SPRDE Level III.