Volume 13 | Issue 1 | March 2026

ISSN: 2160-4746

Journal of Cost Analysis and Parametrics

Editor in Chief: David L. Peeler, Jr., CCEA®

ICEAA
International Cost Estimating & Analysis Association

www.iceaaonline.com

ICEAA

Journal of Cost Analysis and Parametrics

Volume 13 | Issue 1 | March 2026 ISSN: 2160-4746

Editor's Note

David L. Peeler, Jr., CCEA®

Thank you for opening the Journal of Cost Analysis and Parametrics (JCAP). We appreciate your interest and attention to our publication. Up front, we want to express our gratitude to those who have written, submitted, and revised pieces for publication, as well as those who have taken the time and mental energy to review the submissions. The issue contains papers detailing professional work and academic pursuits ranging across cost interests in technology readiness, learning curves, engineering, radar, software, WBS structure, and market segmentation. We hope you find something, if not everything, of interest in these pages.

We kick things off with the best paper awarded at the 2025 ICEAA workshop. Not only is this a superb paper, it was a great presentation by Kyle Coughlin. In his paper *Statistical Modeling of Technology Readiness Levels to Inform Project Acquisition and Schedule Risk*, Coughlin shows the development of a statistical model to measure the probability a technology will advance from one technology readiness level (TRL) to another in a specified period of time.

The second and eighth articles explore learning curves. The former looks at the impact of common parts usage on learning curves. A popular strategy to reduce program costs in commercial and military applications is the use of parts, designs, and tools across multiple aircraft models. However, commonality poses unique challenges to learning curve practitioners. In *Flavors of Commonality: Learning in a Multiple Variant Environment*, Brent Johnstone examines approaches to estimate multiple variant programs, using different learning curve techniques. In the latter piece, *A Markov Model of the Learning Curve*, Harry Larson investigates the use of more statistically robust approaches to learning curves. He illustrates the use of a power function to advance beyond logarithmic transformations to model learning curves, as well as combining them with weighted cost estimating relationships.

Analyzing Development Phase NRE/REC Costs in Defense Acquisition Programs won the 2025 ICEAA Outstanding Air Force Institute of Technology Thesis Award. In this paper, Jason Aristizabal provides novel insights into the development phase of acquisition by analyzing empirical nonrecurring and recurring engineering cost ratios across major work breakdown structure elements, commodity types, and time periods.

Dustin Brewer provides analysis into *The Impact of Advancing Technology on Fire Control Radar Costs*. He examines an integral component of fighter aircraft vital for navigation and target identification. The rapid evolution of radar technology is crucial for aligning with changing mission requirements and is expected to significantly influence radar system costs. His results are critically informative to estimating efforts on upcoming fighter platforms.

Article five is the summary of Stephen Chatterton's thesis research into, *Navigating Defense Software Costs Through Multinomial Logistic Regression*. Using historical data, Chatterton develops two models for categorizing non-recurring software costs in defense projects. His result clashes with conventional wisdom in extant defense cost models and has implications for improved decision-making in defense project planning.

Kyle Marquis presents his study into *Assessing Cost Growth Correlations Between Work Breakdown Structures and System Test and Evaluation*. His research supports the body of knowledge in both project cost management and project risk management. The analysis primarily focuses on the system test and evaluation WBS element, which has been identified as commonly neglected by decision-makers, yet significantly influential on program cost growth.

In the seventh article, *Market Dimensional Expansion, Collapse, Costs, and Viability*, Douglas Howarth examines the demand frontiers for product groupings. The paper looks specifically at the impact of new market segments, with enhanced functionality, on displaced sectors which may contract across one or more demand frontiers. Another interesting paper from Doug that ranges cost analysis through microeconomics, providing decision space for corporate choices.

We hope this overview is helpful in focusing your reading choices. If you don't have time to read the issue in its entirety, maybe this outline will allow you to have cocktail party familiarity with some current topics. Enjoy the pieces you chose to read. May you find something in these pages helpful in your professional pursuits and satisfying to your personal interests.

Statistical Modeling of Technology Readiness Levels to Inform Project Acquisition and Schedule Risk

Kyle Coughlin

The Aerospace Corporation has developed a statistical model based on the National Aeronautics and Space Administration (NASA) TechPort database. This statistical model can measure the probability that a technology will advance from TRL *X* to TRL *Y* in a specified amount of time. It provides an alternative methodology to measuring schedule risk for technologies in the early phases of development. This alternative methodology has the advantage of producing general results without requiring project input information. As detailed below, this methodology also allows for incorporation of information as acquired to tailor the model to the specifics of the project of interest.

The NASA TechPort database is a web-based database that contains information on NASA technology [1]. The home page is shown in Figure 1, and a sample of the available data is shown in Figure 2. The database is publicly accessible through the NASA TechPort website. Techport contains NASA's portfolio of active and completed technology projects, where a technology is defined as "a solution that arises from applying the disciplines of engineering science to synthesize a device, process, or subsystem to enable a specific capability" [1]. TechPort's goal is to "facilitate opportunities for collaboration and partnerships,

Figure 1: Screenshot of the TechPort home page. Source: NASA

analyses of how the Agency is meeting mission needs, and data visualizations of technology drivers that enable key decisions" [1]. Each technology contains information including project description, anticipated benefits, organizational responsibility, project management, work locations, project duration, technology maturity (represented by TRL),

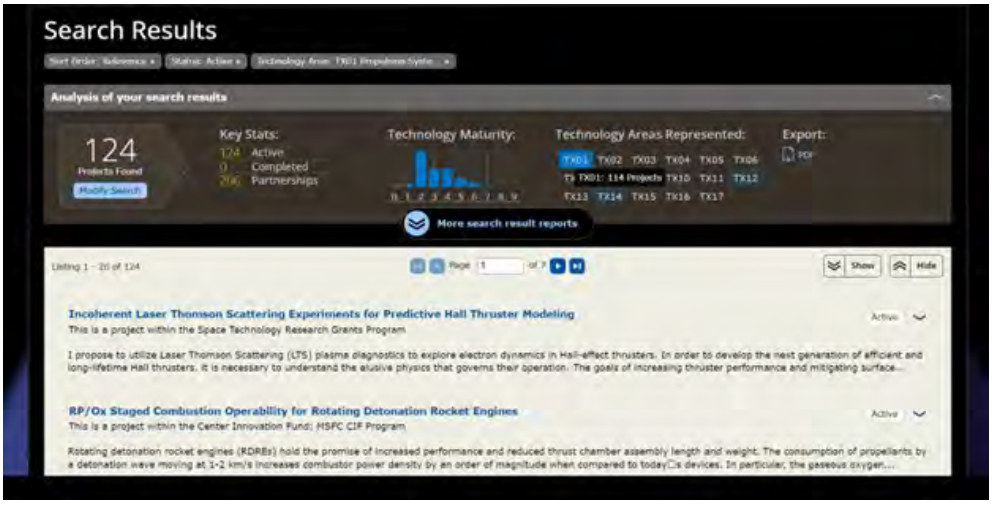


Figure 2: Sample TechPort database search results on Propulsion systems.

and more. An explanation of NASA's TRLs is shown in Figure 3. TRL definitions differ between the Department of Defense (DOD) and NASA. This paper uses NASA definitions.

The model uses start TRL, end TRL, and elapsed project time from all TechPort projects to produce a model that predicts the probability of achieving TRL *Y* from TRL *X* within a given time T. Figure 4 shows the probability of advancing from TRL 1 to TRL 9 in 10 years. The full result is a cumulative distribution function (CDF), where a user can examine the advancement probability at any time. Additionally, an 80 percent prediction interval (PI) is shown on the x axis. This PI shows where the tool predicts 80 percent of future projects will lie when advancing from TRL 1 to TRL 9.

As of January 9, 2025, the TechPort contains 18,953 projects (The database is dynamic, and the quantity of data is increasing). Some of these are missing TRL/date information or have erroneous information, such as the start TRL being 0. When we remove these projects, we are left with 12,163 projects (shown in Figure 5 broken down by status). Note, stalled projects are those designated as completed, but the start and end TRLs are the same. Discarding active projects leaves 10,490 successfully completed projects and 791+63=854 canceled or stalled projects. Thus, we assume that any future project has a

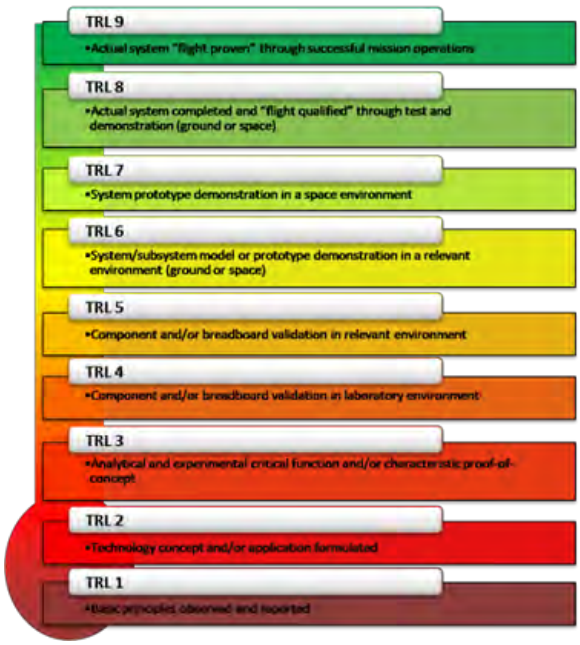


Figure 3: NASA TRL levels. Source: NASA

$$\frac{854}{10490} = 8.1\% \qquad (1)$$

chance of stalling or being canceled. The model is fit to the 10,490 completed projects, and any result is multiplied by 0.924 (92.4 percent) to account for chance of project failure.

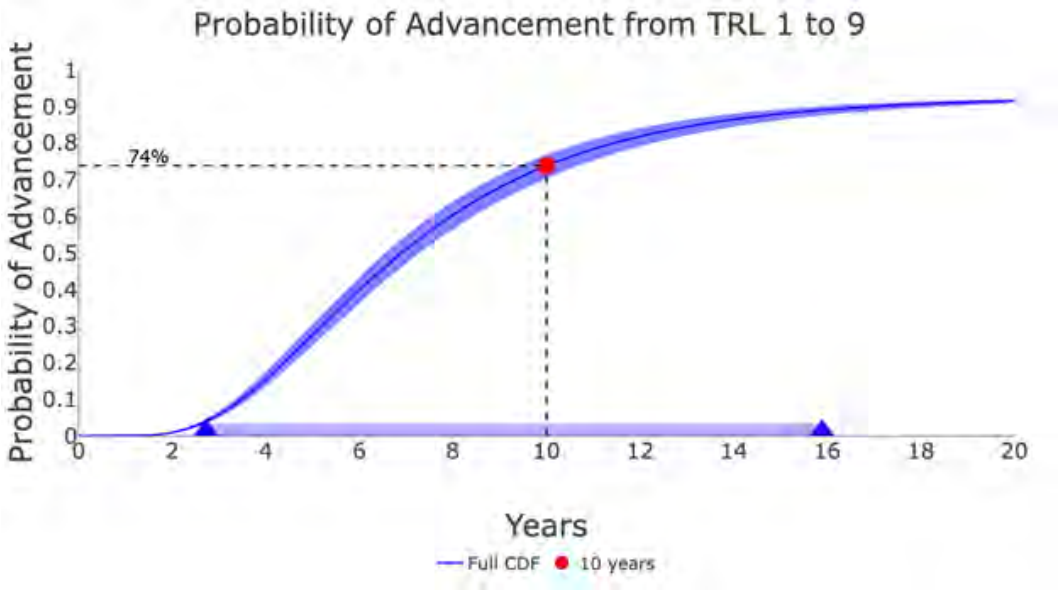


Figure 4: TRL 1 to TRL 9 CDF.

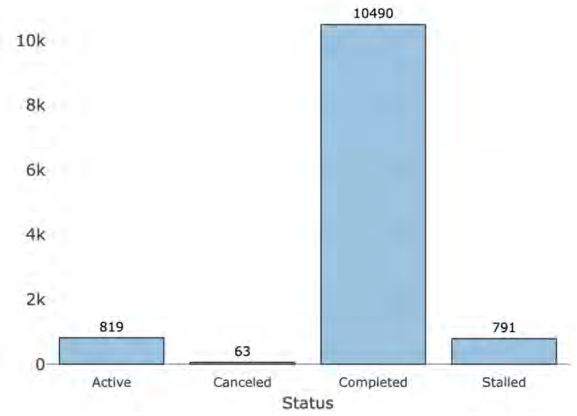


Figure 5: NASA TechPort projects by status.

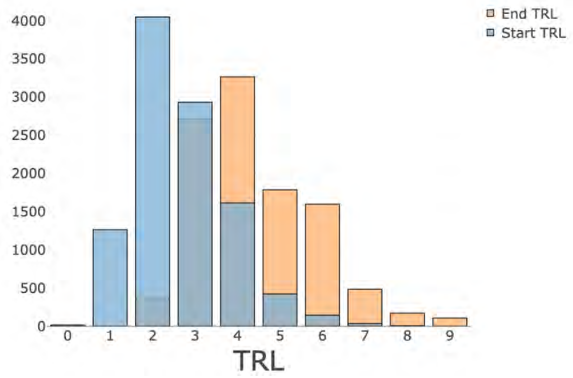


Figure 6: Distribution of start and end TRLs.

Additionally Figure 6 shows the distribution of the start and end TRLs. It shows that the start TRLs cluster around 2 to 4 TRL and the end TRLs cluster around 3 to 6. Thus, the model is best used for projects starting at lower than TRL 4 and ending below TRL 6 — in other words, acquisition programs closer to the beginning of development. The project can technically evaluate all TRL levels as we do have data points at TRLs 7, 8, and 9. Unfortunately, the sample size at those levels combined is fewer than 1,000. This model provides useful data to acquisition programs because, traditionally, in the beginning phases of development, little quantitative information exists, and program planners must rely on personal experience, lessons learned, best practices, and experts' estimates [2].

This schedule risk management tool provides information on technology research and development projects that contain both start and end TRLs and start and end dates. This predictive model can be used in acquisitions to measure which projects have the highest probability of succeeding within a given time frame. For the early phases of an acquisition project where little project specific data is available, this tool provides a quantitative methodology for measuring schedule risk [2]. Given the data available in the TechPort database, the model works best for these stages. The model can be used to evaluate planned or active projects. Suppose a project is planned to take two years to get from TRL 2 to TRL 5, the tool can provide the probability that the project will accomplish this (41 percent).

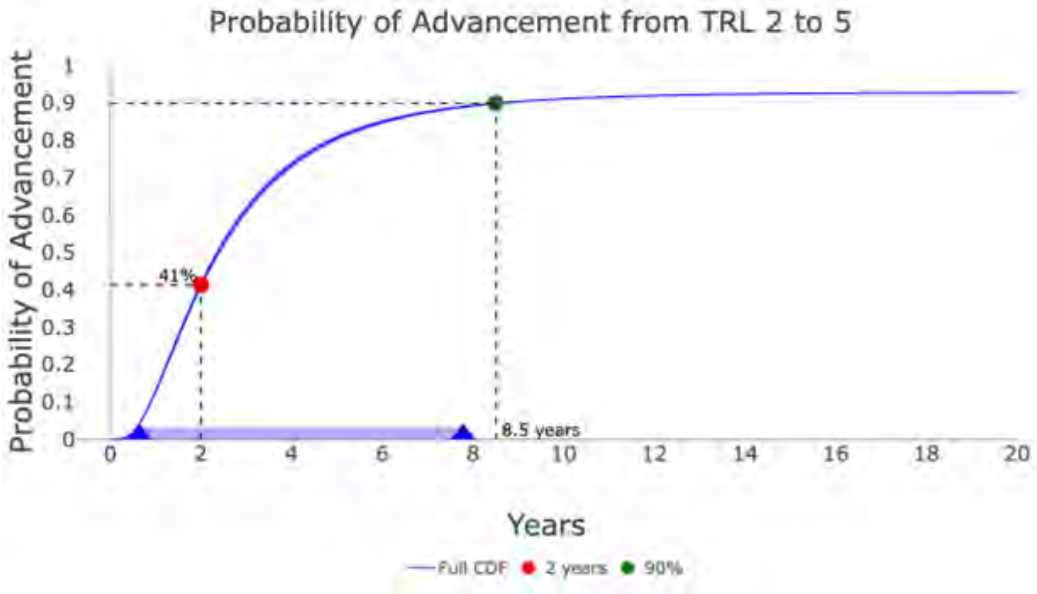


Figure 7: TRLs 2 to 5. Highlighted probability at two years and when probability equals 90 percent.

Number	Category Name
TX01	Propulsion Systems
TX02	Flight Computing and Avionics
TX03	Aerospace Power and Energy Storage
TX04	Robotic Systems
TX05	Communications, Navigations, and Orbital Debris Tracking and Characterization Systems
TX06	Human Health, Life Support, and Habitation Systems
TX07	Exploration Destination Systems
TX08	Sensors and Instruments
TX09	Entry, Descent, and Landing
TX10	Autonomous Systems
TX11	Software, Modeling, Simulation, and Information Processing
TX12	Materials, Structures, Mechanical Systems, and Manufacturing
TX13	Ground, Test, and Surface Systems
TX14	Thermal Management Systems
TX15	Flight Vehicle Systems
TX16	Air Traffic Management and Range Tracking Systems
TX17	Guidance, Navigation, and Control (GN&C)

Table 1: NASA TechPort Database Naming Taxonomies

The scenario can also be flipped. Suppose a project wants to advance from TRL 2 to TRL 5, the tool will determine how long the program should plan to fund it to accomplish the TRL advancement with a 90 percent probability of success (8.5 years). Figure 7 shows both of these results.

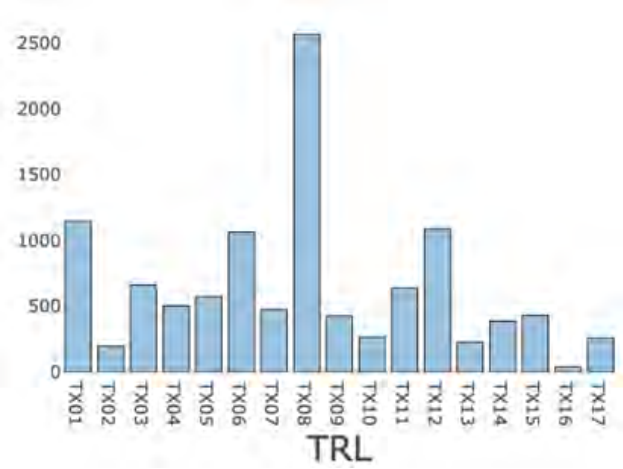


Figure 8: Distribution of NASA TechPort technology types by taxonomy.

As the model is currently being generated using only NASA data, the projects included are primarily focused on the space domain. Programs such as Knudsen-Pump-Based Propulsion for Atmospheric and Martian Exploration, Sapphire Composite Lattices with Ultrafast Laser-Tunable Thermal Expansion, and Secure and Safe Assured Autonomy (S2A2) are included. Table 1 and Figure 8 show that the data included in TechPort is heavily weighted toward a few specific technology areas: Propulsion systems; human health, life support, and habitation systems; sensors and instruments; and materials, structures, mechanical systems, and manufacturing. As such, the model will provide the best results for estimating schedule risk within these areas. These areas are highlighted in Table 1.

Additionally, the model can be configured to include text-based data fields; i.e., natural language data, such as descriptions and taxonomy. This adjustment allows users to input natural language data for their projects, and the model results are generated on a subset of the TechPort database that reflect projects similar to the users' projects, ensuring relevance. Subsetting is done using a large language model (LLM) to determine the technologies in TechPort that are closest to the user project; we are using an LLM because of the size of the data set and LLM's inherent ability to work with the classification natural language processing task. The assumption behind breaking down the data set into areas is that TRL advancement will behave differently for the different types of technologies; for example, rocket fuel development versus biological research. When a relevant technology subset is found from the database, users will receive a prediction better tailored to their specific project. The disadvantage of this subsetting is reduction of sample size and, hence, prediction interval and larger confidence bounds.

An example is shown in Figure 9 where we input a TechPort project[1]:

- **Title:** Development of a Compact Plasma Spectrometer
- **Description:** The objective of this IRAD[2] is the development of new technologies for GSFC-led charged particle instruments. The goal is to develop an instrument concept that is similar in resource footprint to heritage instruments but with enhanced capabilities.
- Start TRL: 2
- End TRL: 5

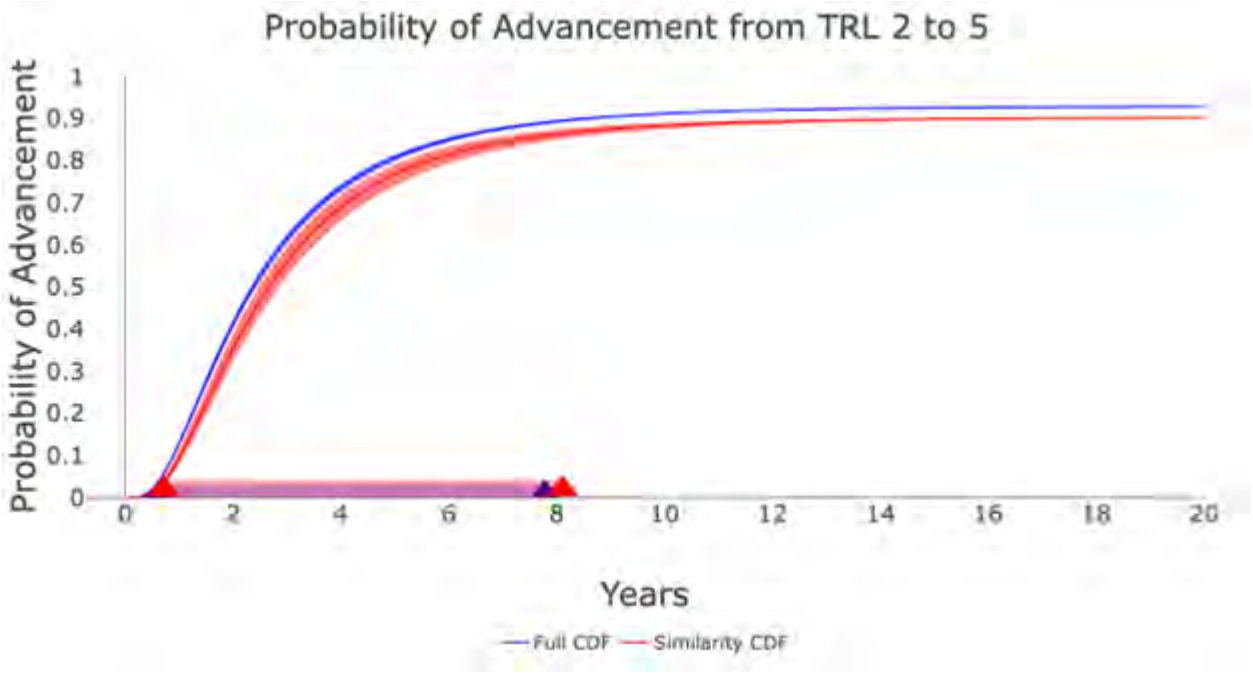


Figure 9: Example of natural language processing (NLP) subsetting.

1. This example is taken directly from the publicly accessible Techport database website.

2. Internal research and development

The model is based on combinations of lognormally distributed random variables. For a given single transition from TRL A to A+1, the time to achieve is a random variable X_A. We can then say any general transition from TRL A to TRL B is another random variable:

$$Y_{A,B} = X_A + X_{A+1} + \cdots + X_{B-1} = \sum_{i=A}^{B-1} X_i \quad (2)$$

Assuming each X_i is lognormally distributed, we can approximate $Y_{A,B}$ as another lognormal distribution. The first step is to fit the mean and standard deviation for each X_i. These are fit by maximum likelihood estimation against the TechPort data. These parameters (in natural log-days) are:

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
μ	4.8313	5.4565	5.015	5.348	5.552	4.971	5.839	4.269
σ	1.0513	1.0513	1.0513	1.0513	1.0513	1.0513	1.0513	1.0513

Table 2: NASA TechPort Database Naming Taxonomies

Then for any transition from TRL A to TRL B, we will approximate the sum of these lognormals by applying Fenton-Wilkinson (F-W) moment-matching. For example, if we want the distribution for TRL 4 to TRL 7, we would estimate the distribution of

$$Y_{4,7} = X_4 + X_5 + X_6 \quad (3)$$

The estimation of the new parameters follows:

$$\sigma_X = \sqrt{log\left(\frac{\sum_4^6 (e^{2\mu_i+\sigma^2})(e^{\sigma^2}-1)}{(\sum_4^6 (e^{\mu_i+.5\sigma^2}))^2} + 1\right)} \quad (4)$$

$$\mu_X = log\left(\sum_{i=4}^{6} \mu_i + .5\sigma^2\right) - .5\sigma_X^2 \quad (5)$$

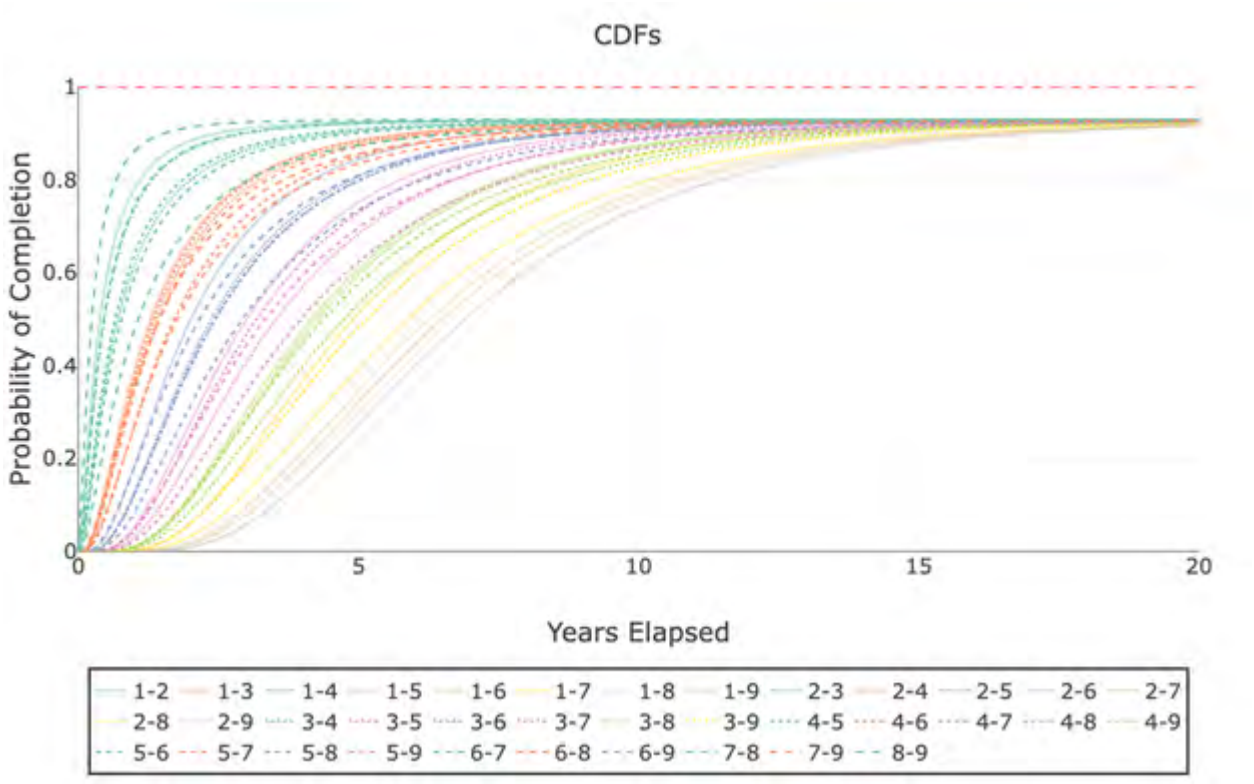


Figure 10: CDFs for time-to-complete of all TRL transitions.

The statistical model without any NLP has been validated by k-fold cross-validation. The flowchart in Figure 11 shows the overview of the process, and Figure 12 visually displays how we are calculating the proportion enclosed. In detail, the process follows six steps.

1.**Partition data into training and holdout sets.** In our study, we chose k = 5 in our k-fold cross-validation. This means we take the 10,490 completed projects and split them into roughly five equally sized groups (also called folds).

2.**Train model on training set.** Selecting one of the five groups to be the holdout set; the other four are grouped together as the training set. The statistical model is trained on this set.

3.**Choose alpha value.** An alpha value from 0.1 to 0.99 is chosen. Repeat the next three steps for each alpha between 0.1 and 0.99.

4.**Determine if each project in holdout is within (1 -alpha)% PI.** For each project in the holdout set, determine a (1-alpha)% PI based on that project's start and end TRLs and determine if its elapsed time is within that PI. If alpha is 0.1, we are looking at a 90 percent PI (i.e., where the model predicts 90 percent of projects will fall).

5.**Count how many projects are within PI.** Divide by size of holdout. In step 4, we can count how many are within the specific PIs and divide by the holdout size. This gives a percentage within the PI. If alpha is 0.1, "perfect" model performance would be 90 percent of the holdout set being inside the PI.

6.**Repeat for each fold.** First, repeat steps 3 through 5 in a single fold, then repeat steps 2 through 5. We have two loops occurring.

Figure 11: Validation flowchart.

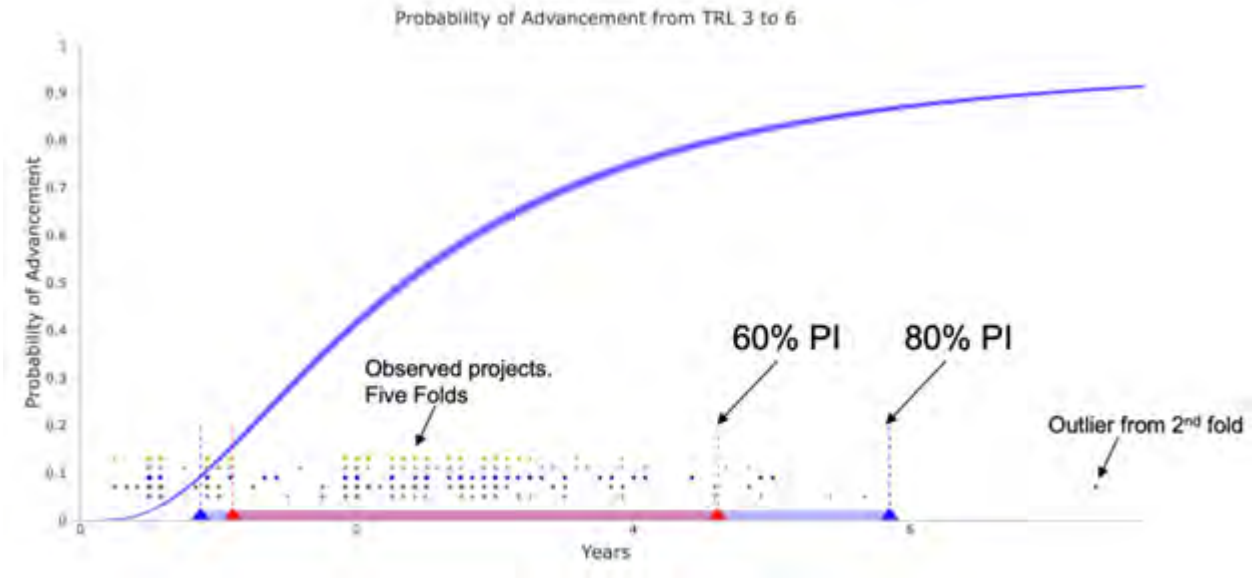


Figure 12: Visual representation of validation.

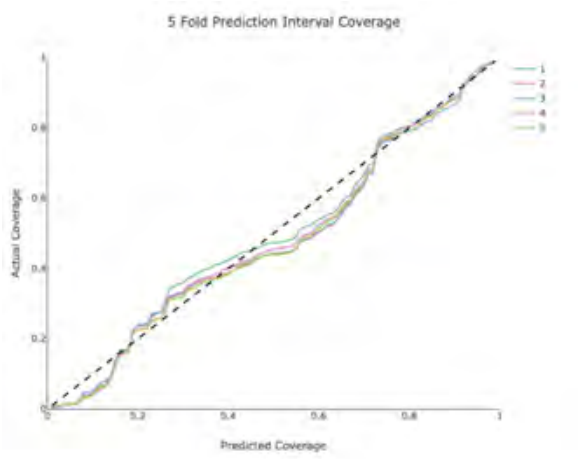


Figure 13: Validation results for five folds.

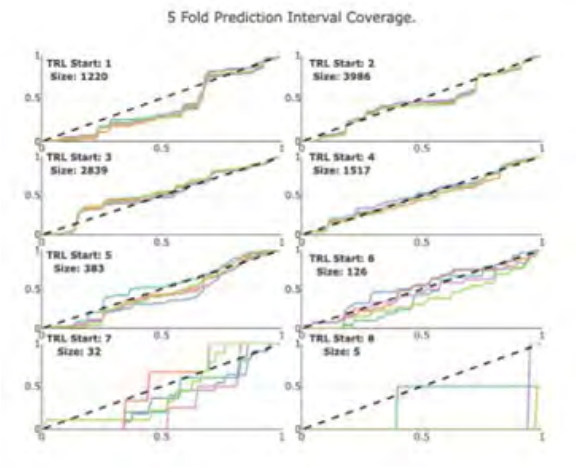


Figure 14: Validation results for five folds broken down by start TRL.

Once this is complete for each fold, the expected coverage and the actual coverage can then be plotted at each alpha level. This is shown in Figure 13. Figure 14 shows the same results but broken down by start TRL. Results from each fold show the model can predict NASA technology transitions within ~10 percent accuracy.

Only the statistical model has been validated. Future development will be focused on finding the best method to subset the TechPort database to generate results on natural subsets of technologies, NLP being the current method. We will then validate the chosen method(s).

In conclusion, The Aerospace Corporation has successfully developed a validated statistical model on the TechPort NASA database. This model can accurately predict the probability that a project will advance from TRL A to TRL B at a given time. This result can be effectively useful for program managers deciding between different projects, especially in the early acquisition stages before detailed information is known. The tool is in the process of being expanded to take advantage of all the available data in the TechPort database, allowing users to consider subsets of the data that better reflect their portfolio.

The next phases of development for this project include:

- Incorporating more of the TechPort database to deliver more interesting and dynamic insights.
  - Adding the ability for users to add taxonomies to their projects and subsets to those taxonomies.
  - Performing qualitative studies of taxonomies with small sample sizes (n<100).
- Examining and mining other databases for information to expand the set of technologies.
  - Primarily interested in acquiring financial data.
  - Other agencies than NASA.
- Incorporating a standardized taxonomy structure with a similar search to generate submodels on specific technology categories.
- Working directly with NASA to conduct validation studies.

References

National Aeronautics and Space Association (NASA), "TechPort - Find it, Build it, Share it," NASA, 02 October 2023. [Online]. https://www.nasa.gov/techport-find-it-build-it-share-it/#:~:text=TechPort%20is%20a%20web%2Dbased,%2C%20collaboration%2C%20partnership%20and%20communication. (Accessed July 12, 2024.)

Defense Acquisition University, "AcqNotes: Schedule Risk," Defense Acquisition University, July 21, 2023. [Online]. https://acqnotes.com/acqnote/tasks/schedule-risk. (Accessed July 12, 2024.]

S. Weglarczyk, "Kernel density estimation and its application," ITM Web of Conferences, vol. 23, November 2018.

E. Lower, "What is Kernel Density Estimation?," 2023.

M. Wand and M. Jones, Kernel Smoothing, London: Chapman & Hall/CRC, 1995.

Kyle Coughlin *works as an electric transmission and distribution data scientist for Rocky Mountain Power in Salt Lake City. He previously worked as a reliability statistician for The Aerospace Corporation. He won Best Paper Overall at the 2025 ICEAA Professional Development & Training Workshop for his work on TRL project advancement predictions.*

Flavors of Commonality: Learning in a Multiple Variant Environment

Brent M. Johnstone

Commonality – the reuse of parts, designs and tools across multiple aircraft models -- is a popular strategy to reduce program costs in commercial and military applications. But its use poses unique challenges to learning curve practitioners. This paper examines five approaches to estimating multiple variant programs using different learning curve techniques. A notional dataset is created, and the accuracy of each method is measured to highlight the advantages and disadvantages of each.

Commonality refers to the reuse of parts, designs, tools, engineering, and/or manufacturing processes between different models or variants of a product. As a design and build strategy, commonality is frequently observed in both commercial and military aircraft. In the commercial world, design commonality is promoted to reduce development costs, shorten the design cycle, and create greater market penetration. (Zhang, 2019) Similarly, in military usage commonality is advanced as a strategy to save development, production and sustainment costs. The Joint Advanced Strike Technology (JAST) program identified a potential Engineering and Manufacturing Development (EMD) savings of 30-40% in airframe design, 40% savings in test, 30-40% savings in manufacturing and 60-70% savings in avionics for a common fighter program relative to three unique stand-alone programs[1]. (*JAST Commonality Study*, 1996).

Commonality can appear in various forms and degrees, including:

- Fighter aircraft which come in a one-seat (combat) or two-seat (trainer) configuration where the cockpit is the only distinguishable structural difference. The F-16C/D and the F/A-18E/F fighters are current examples.
- Commercial jetliners where fuselage lengths are stretched or shortened from a baseline configuration, allowing airlines to choose aircraft with greater or lesser seats to support a particular market. The Boeing 737 MAX comes in four versions (-7/-8/-9/-10) with the same basic aircraft but fuselage length ranging from 116 to 143 feet with seating ranging from 138 to 204 passengers. (*About the Boeing 737 MAX*, n.d.)
- Military aircraft designed to support a single military service but come in multiple configurations. For example, the C-130J aircraft comes in a standard cargo model (J-30), a tanker version (KC-30), special operations version (HC/MC), short body (J), weather reconnaissance (WC) or electronic variant (EC).
- Military aircraft designed to support more than one military service. Examples include the F-4 fighter, A-7 attack aircraft, JPATS T-6A, JSTARS E-8 and V-22. (Lorell, 2013) The most complex example is the F-35 fighter, whose A model (conventional takeoff and landing), B model (short takeoff and vertical landing), and C model (carrier variant) have sold over 1,000 aircraft to US and international services.

What impact might we expect to see on build hours? Learning curve theory assumes the same product is built repetitiously over multiple cycles resulting in a reduction of hours over time. If the product is not the same, however, we would expect that some learning loss from prior builds when the alternate configuration is built. This situation is like an engineering design change where the configuration is altered, resulting in some degree of lost learning, evidenced by a regression on the overall learning curve and higher hours per unit. (Johnstone, 2023) But building multiple models or variants of an

1. The idea that commonality results in cost savings is not universally accepted. A 2013 RAND paper argued that joint aircraft programs historically experienced higher acquisition cost growth over single-service aircraft programs and did not save costs over the program life cycle. (Lorell, 2013) Further analysis of that assertion is beyond the scope of this paper..

aircraft has additional complexities over and above an engineering design change. For one thing, the impact of an individual engineering design change is time-limited: after the initial learning loss, there is a relatively rapid return to the underlying curve. The impact of building multiple models, by contrast, can extend across the entire program life cycle.

Theoretically, the degree of learning loss should vary proportionally to the extent of the configuration differences between the different variants. How much learning we can expect to be shared or lost between variants will depend on several factors. For example, for an aircraft program we might ask:

1. How common are the airframe engineering designs between the different variants?
2. Do they use a common set of mission and vehicle systems?
3. Will the different variants be built on a common production line, or will they be built on separate production lines, possibly even by different companies?
4. To what extent will the different variants be built using common tooling or manufacturing processes?
5. Will each variant be built using dedicated crews of assemblers? Or will crews be cycled between models as aircraft move down the production line?

The answers to these questions will determine how much learning transfer between variants we might reasonably expect. They will also inform us which estimating methodology would serve us best.

Jones (2019) identifies four estimating options for dealing with commonality. We will deal with each of these in turn in the following sections:

1. Ignore Differences (ID) – Assume a common learning curve and ignore any cost impact of multiple models.
2. Fixed Factors (FF) – Assume a common underlying curve and adjust for variant differences through a fixed factor or relationship between variants.
3. Total Separation (TS) – Assume each variant has a unique learning curve slope and no learning transfer occurs between variants.
4. Proportional Representation (PR) – Assume a given combination of common or unique work has its own peculiar learning curve, but all of them share a common rate of learning.

To this list, we will add a fifth alternative:

5. Partial Separation (PS) – Assume each variant has a unique learning curve slope but allow that there is learning transfer between variants.

For simplicity of reference, this paper will also use Jones' nomenclature to refer to each of these options.

Three of these five options assume that the multiple variants will share at least a common rate of learning. Why might we expect different variants to share common learning curves? Multiple variants in a single program may nonetheless share many common decisions -- such as common investment strategies, design tools, tooling and build philosophy. A 1981 study of the contributors to learning reported that 78% of cost improvement could be attributed to causes other than operator learning (tooling, design engineering, management controls). (Jefferson, 1981) Even mechanics should be able to continue learning on similar, if not identical, operations, provided their crews are permitted to cycle between model versions. This view argues that the observed rate of learning on a multi-variant program will tend to be more common than unique.

A Notional Program

To illustrate application of these methodologies but avoid compromising proprietary information, a dummy set of data has been constructed. This notional data presumes a two variant aircraft program with 80% of the production aircraft built in an U.S. Air Force (USAF) configuration while the remaining 20% are a U.S. Navy (USN) configuration. We will designate the USAF aircraft as Model A and the USN aircraft as Model B. A total of eight Engineering and Manufacturing Development (EMD) aircraft and 500+ production aircraft are built. For this analysis, we will assume that 65% of the manufacturing effort is common between the USAF and USN versions, with the remaining 35% being unique to each variant.

Figure 1 displays a subset of this database:

Total	Phase	Service	Variant	Adj Hours
1	EMD	USAF	A	384,354
2	EMD	USAF	A	392,722
3	EMD	Navy	B	359,041
4	EMD	Navy	B	366,820
5	EMD	USAF	A	316,530
6	EMD	USAF	A	303,031
7	EMD	Navy	B	355,896
8	EMD	Navy	B	329,786
9	Lot 1	USAF	A	294,270
10	Lot 1	USAF	A	283,824
⋮				⋮
505	Lot 14	USAF	A	62,361
506	Lot 14	USAF	A	62,911
507	Lot 14	USAF	A	61,762
508	Lot 14	Navy	B	73,903
509	Lot 14	USAF	A	70,701
510	Lot 14	USAF	A	57,930
511	Lot 14	USAF	A	61,808
512	Lot 14	USAF	A	65,429
513	Lot 14	Navy	B	76,368
514	Lot 14	USAF	A	65,222

Figure 1. Subset of Notional Program Data

Figure 2 shows the program data graphically.

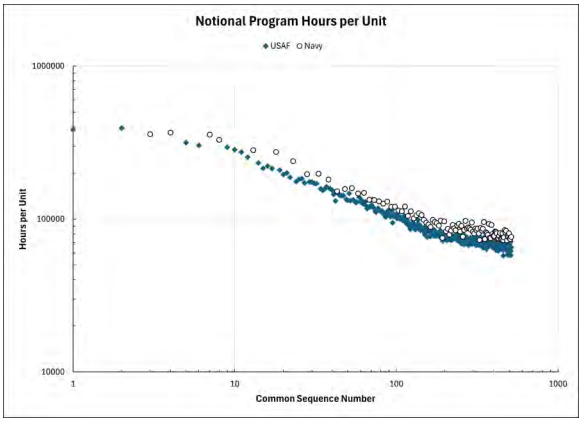


Figure 2. Notional Program Hours per Unit

The program data shows the familiar S-curve shape seen on many historical programs, where the development phase shows relatively shallow rates of learning, followed by an initial production phase where the rate of learning steepens and significant reductions in hours are made due to reduced engineering changes and improved processes and tooling. The third phase that follows displays a flattening of the learning curve slope as manufacturing processes stabilize, and the program settles into higher production rates. (Engwall, 2001; Jones, 2001; Cochran, 1968; Cochran, 1961)

As demonstrated in prior articles (Johnstone, 2022), we can analyze data following the S-curve pattern using a piecewise regression. We start from our familiar improvement curve model:

$$y = \alpha_1 x^{\beta_1} \quad (1)$$

Where:

y = Manufacturing hours per unit

x = Cumulative units built to date

α_1 = Y-intercept, equal to theoretical first unit (TFU) hours

β_1 = Rate of learning, such that 2^{β} equals learning curve slope

After hours per unit and cumulative quantities are converted to natural logarithms, this yields the following linear form:

$$\ln y = \ln \alpha_1 + \beta_1 \ln x \quad (2)$$

Kennedy (1992) outlines a method for using dummy variables to capture a change in the intercept and slope coefficients between two periods. To create a two-leg segmented learning curve, we introduce breakpoint unit *T*. Based on our *a priori* selection for *T*, data is separated into pre-break period 1 ($x < T$) and post-break period 2 ($x \geq T$). In addition, dummy variable *D* is created such that *D* is zero for period 1, and one for period 2. Product dummy variable *Dx* is also created such that *Dx* takes the value *x* in period 2 but is 0 otherwise. This creates the regression equation:

$$\ln y = \ln \alpha_1 + \ln \alpha_2 D + \beta_1 \ln x + \beta_2 \ln Dx \quad (3)$$

Equation (3) represents two separate cases. Where $x < T$, variables *D* and *Dx* are 0 and equation (3) reduces back to our standard improvement curve equation (2). But where $x \geq T$ and *D* takes the value of one, different intercept and slope values are introduced such that:

$$\ln y = \ln(\alpha_1 + \alpha_2) + (\beta_1 + \beta_2) \ln Dx \quad (4)$$

Where:

y = Manufacturing hours per unit (HPU)

α_1 = Y-intercept for leg #1, equal to theoretical first unit hours for leg #1

α_2 = Intercept adjustment for leg #2, such that $\alpha_1 + \alpha_2$ equals the Y-intercept for leg #2

β_1 = Rate of learning for leg #1, such that 2^{β} equals learning curve slope #1

β_2 = Rate of learning for leg #2, such that $2^{(\beta 1 - \beta 2)}$ equals learning curve for leg #2

Similarly, this methodology can be expanded to account for three or more legs of the learning curve.

We will use the first 370 aircraft (EMD and Lots 1-11) to develop historical learning curve slopes applying a piecewise regression as well as using each of these methodologies. We will then apply the resulting historical learning curves to forecast the next 144 aircraft (Lots 12-14). Comparing the forecast to the realized hours for those later aircraft will demonstrate some of the potential forecasting issues that may arise.

Ignore Differences (ID)

The first estimating option to deal with variant commonality is to simply ignore it. While this hardly seems like an option at all, there may be good reasons to consider it. Not all the differences between aircraft configurations result in a significant cost difference. For example, the Lockheed L-1011 commercial jetliner came in five unique configurations (-1, -100, -200, -250, -500) but from a learning curve perspective, only the shortened -500 configuration displayed a statistically significant variation in cost. The rest of the models could simply be combined for analysis purposes. (Benkard, 2000)

A visual review of Figure 2 tells us that there is a cost differential between the USAF and USN models, and that the ID methodology will probably not provide either a good fit to the data or provide particularly accurate forecasts. Nonetheless, we will set up our data for regression as follows under Figure 3. In each case we examine, the "common sequence number" (the actual build sequence, regardless of model) will stay the same. On the other hand, the "effective sequence number" will vary depending on the particular methodology we choose. For learning curve regression purposes, the effective sequence number will be the cumulative unit variable.

In the ID methodology, the effective sequence number is identical to the common sequence number. Thus, we are implicitly assuming that for cost purposes there is no difference between the variants and that 100% of the prior learning will be transferred between models. Note that there is nothing in the independent variables that identifies USAF or USN models.

The results from the best fit regression are shown in Figure 4.

				Curve Breakpoints		Dependent Variable	Independent Variables				
Common Sequence Number	Effective Sequence Number	Model	HPU	T_1	T_2	LN(HPU)	β1	α_2	β2	α_3	β3
1	1	A	384,354	9	151	12.86	-	-	-	-	-
2	2	A	392,722	9	151	12.88	0.69	-	-	-	-
3	3	B	359,041	9	151	12.79	1.10	-	-	-	-
4	4	B	366,820	9	151	12.81	1.39	-	-	-	-
5	5	A	316,530	9	151	12.67	1.61	-	-	-	-
6	6	A	303,031	9	151	12.62	1.79	-	-	-	-
7	7	B	355,896	9	151	12.78	1.95	-	-	-	-
8	8	B	329,786	9	151	12.71	2.08	-	-	-	-
9	9	A	294,270	9	151	12.59	-	1	2.20	-	-
10	10	A	283,824	9	151	12.56	-	1	2.30	-	-
⋮											
149	149	A	87,845	9	151	11.38	-	1	5.00	-	-
150	150	A	79,812	9	151	11.29	-	1	5.01	-	-
151	151	A	78,318	9	151	11.27	-	-	-	1	5.02
152	152	A	81,745	9	151	11.31	-	-	-	1	5.02
153	153	B	94,523	9	151	11.46	-	-	-	1	5.03
154	154	A	86,816	9	151	11.37	-	-	-	1	5.04
⋮											
366	366	A	66,039	9	151	11.10	-	-	-	1	5.90
367	367	A	66,241	9	151	11.10	-	-	-	1	5.91
368	368	B	78,852	9	151	11.28	-	-	-	1	5.91
369	369	A	72,902	9	151	11.20	-	-	-	1	5.91
370	370	A	71,358	9	151	11.18	-	-	-	1	5.91

Figure 3. Subset of Notional Data Set Up for Regression (ID Methodology)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	370
Equation in Unit Space:	LN_HRS = 12.89 + (-0.09531) * BETA1 + (-0.4265) * BETA2 + (-0.1968) * BETA3 + 0.6366 * ALPHA2 + (-0.5584) * ALPHA3

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.8913	0.0601		214.6371	0.0000	1.0000
BETA1	-0.0953	0.0406	-0.0559	-2.3484	0.0193	0.9807
BETA2	-0.4265	0.0092	-2.4114	-46.4237	0.0000	1.0000
BETA3	-0.1968	0.0200	-1.4561	-9.8498	0.0000	1.0000
ALPHA2	0.6366	0.0716	0.8415	8.8915	0.0000	1.0000
ALPHA3	-0.5584	0.1259	-0.7451	-4.4363	0.0000	1.0000

TFU - Leg 1	396,828
Slope - Leg 1	93.6%
Slope - Leg 2	74.4%
Slope - Leg 3	87.3%
TFU - Leg 2	750,051
TFU - Leg 3	227,039

Figure 4. Best Fit Regression – ID Methodology

Figure 5 shows the derived learning curve projected through the end of Lot 14. As expected, while the ID methodology provides a good forecast at the bottom-line, it does not compare well to Lot 12 and on actual hours at the individual variant level. For instance, it does a poor job of forecasting the B model, which is understated by almost 13%.

However, if for a given estimate we are not concerned with accuracy at the variant level, the ID methodology may provide an answer. In addition, where the two models differ only in a particular area, i.e., the cockpit for a one-seat versus a two-seat variant, the ID methodology will probably work for all the unaffected component areas.

Fixed Factors (FF)

The Fixed Factors methodology is identical to the ID methodology, but with one critical difference: dummy variables (1 or 0) are established to account for the difference in aircraft models. Like the ID methodology, the effective sequence number is identical to the common sequence number and is not adjusted whenever there is a model change.

Figure 6 shows the setup of the data for regression, where we have added two columns at the far right of the table to add the B model dummy variable. The results from the best fit regression are shown in Figure 7.

Interpretation of the B model dummy variable needs some explanation. The regression result (0.1520) is in logarithmic form and requires transformation.

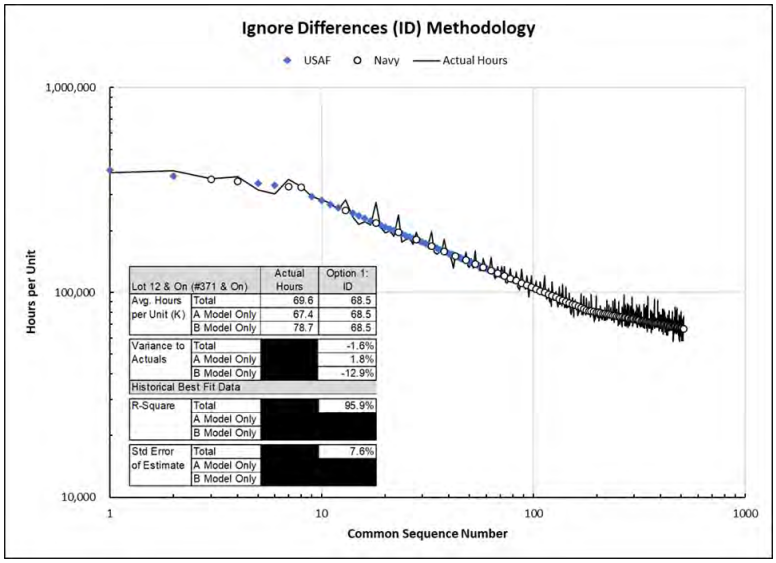


Figure 5. Learning Curve Best Fit & Forecast, ID Methodology

Common Sequence Number	Effective Sequence Number	Model	HPU	Curve Breakpoints		Dependent Variable	Independent Variables					
				T_1	T_2	LN(HPU)	β1	α_2	β2	α_3	β3	B Model Dummy
1	1	A	384,354	9	151	12.86	-	-	-	-	-	-
2	2	A	392,722	9	151	12.88	0.69	-	-	-	-	-
3	3	B	359,041	9	151	12.79	1.10	-	-	-	-	1
4	4	B	366,820	9	151	12.81	1.39	-	-	-	-	1
5	5	A	316,530	9	151	12.67	1.61	-	-	-	-	-
6	6	A	303,031	9	151	12.62	1.79	-	-	-	-	-
7	7	B	355,896	9	151	12.78	1.95	-	-	-	-	1
8	8	B	329,786	9	151	12.71	2.08	-	-	-	-	1
9	9	A	294,270	9	151	12.59	-	1	2.20	-	-	-
10	10	A	283,824	9	151	12.56	-	1	2.30	-	-	-
					⋮			⋮				
149	149	A	87,845	9	151	11.38	-	1	5.00	-	-	-
150	150	A	79,812	9	151	11.29	-	1	5.01	-	-	-
151	151	A	78,318	9	151	11.27	-	-	-	1	5.02	-
152	152	A	81,745	9	151	11.31	-	-	-	1	5.02	-
153	153	B	94,523	9	151	11.46	-	-	-	1	5.03	1
154	154	A	86,816	9	151	11.37	-	-	-	1	5.04	-
					⋮			⋮				
366	366	A	66,039	9	151	11.10	-	-	-	1	5.90	-
367	367	A	66,241	9	151	11.10	-	-	-	1	5.91	-
368	368	B	78,852	9	151	11.28	-	-	-	1	5.91	1
369	369	A	72,902	9	151	11.20	-	-	-	1	5.91	-
370	370	A	71,358	9	151	11.18	-	-	-	1	5.91	-

Figure 6. Subset of Notional Data Set Up for Regression (FF Methodology)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	370
Equation in Unit Space:	LN_HRS = 12.89 + (-0.1483) * BETA1 + (-0.4283) * BETA2 + (-0.1968) * BETA3 + 0.6199 * ALPHA2 + (-0.583) * ALPHA3 + 0.152 * B_MODE

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.8855	0.0351		366.9231	0.0000	1.0000
BETA1	-0.1483	0.0238	-0.0870	-6.2282	0.0000	1.0000
BETA2	-0.4283	0.0054	-2.4215	-79.7236	0.0000	1.0000
BETA3	-0.1968	0.0117	-1.4562	-16.8467	0.0000	1.0000
ALPHA2	0.6199	0.0419	0.8193	14.8043	0.0000	1.0000
ALPHA3	-0.5830	0.0736	-0.7780	-7.9213	0.0000	1.0000
B_MODEL	0.1520	0.0057	0.1669	26.4904	0.0000	1.0000

TFU - Leg 1	394,563
Slope - Leg 1	90.2%
Slope - Leg 2	74.3%
Slope - Leg 3	87.2%
TFU - Leg 2	733,354
TFU - Leg 3	220,252
B Model Factor	1.164

Figure 7. Best Fit Regression – FF Methodology

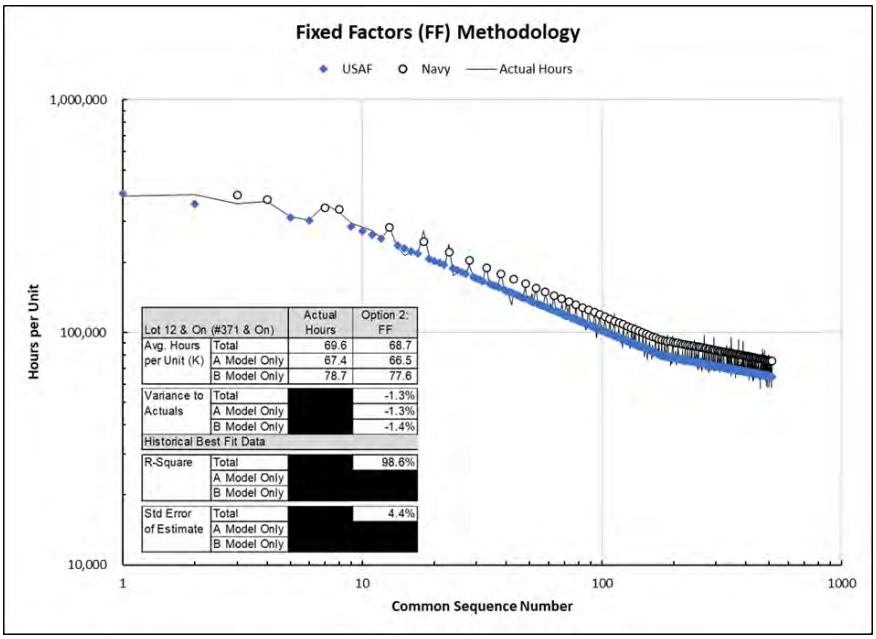


Figure 8. Learning Curve Best Fit & Forecast, FF Methodology

After exponentiation, the value becomes 1.164, which means that all else equal, the hours for the B variant are 16.4% higher than the A model.

Figure 8 shows the derived learning curve projected through the end of Lot 14. As expected, the FF forecast provides an improved fit against Lot 12 and on actual hours both in total and at the variant level. It also visually displays that our use of a dummy variable for the USN model has essentially established two parallel learning curves with identical slopes – one for the A model and one for the B.

The FF methodology does make the implicit assumption that the cost relationship between the models is relatively stable over time. If we were forecasting hours without the benefit of prior actual hour history, the relationship between the variants could be established by other means, such as the use of Industrial Engineering standards.

Total Separation (TS)

The Total Separation (TS) methodology is the opposite of the ID methodology. Instead of assuming that all variants are common and that 100% learning transfer will take place between the variants, the TS methodology treats each variant as unique and assumes that *no* learning transfer will take place.

The TS methodology is best suited for build environments where the individual models are produced in different locations. An excellent example is the Eurofighter Typhoon, which has four separate assembly lines (Germany, Italy, Spain, and the United Kingdom) where each participating country is responsible for final assembly of its national aircraft. (*Eurofighter Typhoon*, n.d.) It is also well suited for situations where unique work is performed, e.g., the cockpit for a one-seat fighter versus a two-seat trainer version. It might also make sense if each variant has a dedicated build crew, and there is no cycling of mechanics between models.

To prepare our data for regression analysis, the effective sequence numbers will be modified to count only the cumulative builds for each model, as shown in Figure 9. Note that the change in effective sequence methodology required us to adjust our breakpoints for the multi-leg curve, assuming we want to retain our previous breaks at the end of EMD (common sequence #8) and mid-Lot 7 (common sequence #150). In addition, we will separate our data into two distinct regression models, one for USAF and one for USN.

The results from the regression are shown in Figure 10 (USAF model only) and Figure 11 (USN model only).

Common Sequence Number	Effective Sequence Number	Model	HPU	Curve Breakpoints		Dependent Variable	Independent Variables				
				T_1	T_2	LN(HPU)	β1	α_2	β2	α_3	β3
1	1	A	384,354	5	119	12.86	-	-	-	-	-
2	2	A	392,722	5	119	12.88	0.69	-	-	-	-
3	1	B	359,041	5	33	12.79	-	-	-	-	-
4	2	B	366,820	5	33	12.81	0.69	-	-	-	-
5	3	A	316,530	5	119	12.67	1.10	-	-	-	-
6	4	A	303,031	5	119	12.62	1.39	-	-	-	-
7	3	B	355,896	5	33	12.78	1.10	-	-	-	-
8	4	B	329,786	5	33	12.71	1.39	-	-	-	-
9	5	A	294,270	5	119	12.59	-	1	1.61	-	-
10	6	A	283,824	5	119	12.56	-	1	1.79	-	-
					⋮			⋮			
149	117	A	87,845	5	119	11.38	-	1	4.76	-	-
150	118	A	79,812	5	119	11.29	-	1	4.77	-	-
151	119	A	78,318	5	119	11.27	-	-	-	1	4.78
152	120	A	81,745	5	119	11.31	-	-	-	1	4.79
153	33	B	94,523	5	33	11.46	-	-	-	1	3.50
154	121	A	86,816	5	119	11.37	-	-	-	1	4.80
					⋮			⋮			
366	291	A	66,039	5	119	11.10	-	-	-	1	5.67
367	292	A	66,241	5	119	11.10	-	-	-	1	5.68
368	76	B	78,852	5	33	11.28	-	-	-	1	4.33
369	293	A	72,902	5	119	11.20	-	-	-	1	5.68
370	294	A	71,358	5	119	11.18	-	-	-	1	5.68

Figure 9. Subset of Notional Data Set Up for Regression (TS Methodology)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	294
Equation in Unit Space:	LN_HRS = 12.9 + (-0.1862) * BETA1 + (-0.3977) * BETA2 + (-0.209) * BETA3 + 0.3635 * ALPHA2 + (-0.5838) * ALPHA3

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.9047	0.0358		360.3412	0.0000	1.0000
BETA1	-0.1862	0.0377	-0.0589	-4.9395	0.0000	1.0000
BETA2	-0.3977	0.0049	-2.2356	-80.9902	0.0000	1.0000
BETA3	-0.2090	0.0115	-1.5626	-18.1867	0.0000	1.0000
ALPHA2	0.3635	0.0408	0.5084	8.9133	0.0000	1.0000
ALPHA3	-0.5838	0.0707	-0.8213	-8.2568	0.0000	1.0000

TFU - Leg 1	402,196
Slope - Leg 1	87.9%
Slope - Leg 2	75.9%
Slope - Leg 3	86.5%
TFU - Leg 2	578,528
TFU - Leg 3	224,336

Figure 10. Best Fit Regression – TS Methodology (USAF Only)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	76
Equation in Unit Space:	LN_HRS = 12.81 + (-0.05086) * BETA1 + (-0.5459) * BETA2 + (-0.1468) * BETA3 + 0.5661 * ALPHA2 + (-0.8621) * ALPHA3

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.8135	0.0579		221.4691	0.0000	1.0000
BETA1	-0.0509	0.0609	-0.0272	-0.8351	0.4065	0.5935
BETA2	-0.5459	0.0231	-1.8901	-23.6604	0.0000	1.0000
BETA3	-0.1468	0.0394	-0.7215	-3.7246	0.0004	0.9996
ALPHA2	0.5661	0.0875	0.6818	6.4670	0.0000	1.0000
ALPHA3	-0.8621	0.1671	-1.0629	-5.1601	0.0000	1.0000

TFU - Leg 1	367,146
Slope - Leg 1	96.5%
Slope - Leg 2	68.5%
Slope - Leg 3	90.3%
TFU - Leg 2	646,678
TFU - Leg 3	155,033

Figure 11. Best Fit Regression – TS Methodology (USN Only)

Comparison of the derived learning curve slopes shows a substantive difference between the two variants now. The B model EMD slope is 96.5% versus 87.9% for the A model, while the middle leg has a 68.5% slope for the B model versus 75.9% for the A model. The slope for the middle leg is so steep for the B model because the same reduction in B model hours is now calculated over B peculiar units 5 thru 32, not common units 9 thru 150 as in the previous examples. If, in fact, there is significant cross-variant learning benefit occurring, the TS methodology has, in this instance, overstated the true rate of B model learning.

Conversely, Jones (2019) argues that over longer production runs, the TS methodology is likely to produce *inflated* values. Notice, for instance, the significant difference between slopes for the third leg – 90.3% for B model versus 86.5% for A model – if this was extended out another several hundred units, we might see such an overstatement.

Figure 12 shows the derived learning curve projected through the end of Lot 14. The TS forecast provides the tightest fit to Lot 12 and on actual hours. This result is more a function of luck than skill, though, because there is a wider gap at the variant level, particularly for the B model.

Regardless of fit statistics, the TS methodology would only be appropriate, if we are reasonably certain there is limited or no transfer of learning between variants. To use it otherwise would risk the potential understatement or overstatement of future hours.

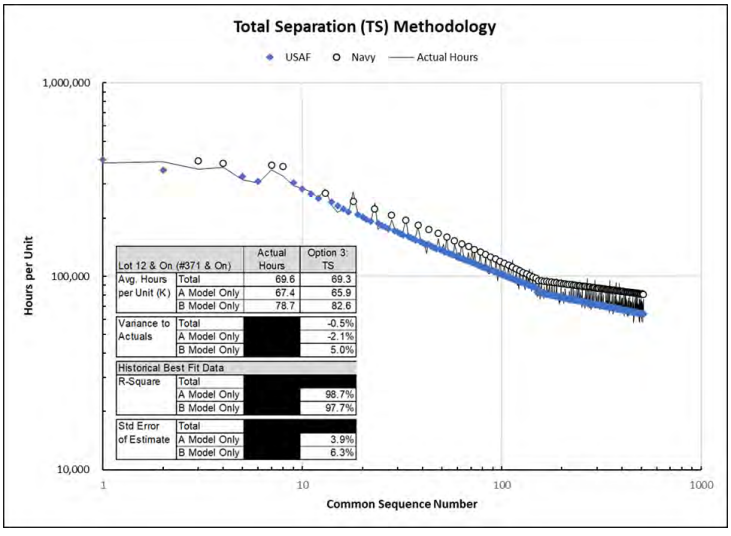


Figure 12. Learning Curve Best Fit & Forecast, TS Methodology

Partial Separation (PS)

The Partial Separation (PS) methodology is a hybrid methodology. Like the TS approach, it calculates different rates of learning for each variant. Like FF, it assumes there is learning transfer between variants.

The only differences in the setup of our data compared to the TS methodology is that we will revert to making the effective sequence number equal to the common sequence number and reset our curve breakpoints accordingly, before splitting our data into separate USAF and USN runs, as seen in Figure 13. The results from the regression are shown in Figure 14 (USAF model only) and Figure 15 (USN model only).

Common Sequence Number	Effective Sequence Number	Model	HPU	Curve Breakpoints		Dependent Variable	Independent Variables				
				T_1	T_2	LN(HPU)	β1	α_2	β2	α_3	β3
1	1	A	384,354	9	151	12.86	-	-	-	-	-
2	2	A	392,722	9	151	12.88	0.69	-	-	-	-
3	3	B	359,041	9	151	12.79	1.10	-	-	-	-
4	4	B	366,820	9	151	12.81	1.39	-	-	-	-
5	5	A	316,530	9	151	12.67	1.61	-	-	-	-
6	6	A	303,031	9	151	12.62	1.79	-	-	-	-
7	7	B	355,896	9	151	12.78	1.95	-	-	-	-
8	8	B	329,786	9	151	12.71	2.08	-	-	-	-
9	9	A	294,270	9	151	12.59	-	1	2.20	-	-
10	10	A	283,824	9	151	12.56	-	1	2.30	-	-
					⋮			⋮			
149	149	A	87,845	9	151	11.38	-	1	5.00	-	-
150	150	A	79,812	9	151	11.29	-	1	5.01	-	-
151	151	A	78,318	9	151	11.27	-	-	-	1	5.02
152	152	A	81,745	9	151	11.31	-	-	-	1	5.02
153	153	B	94,523	9	151	11.46	-	-	-	1	5.03
154	154	A	86,816	9	151	11.37	-	-	-	1	5.04
					⋮			⋮			
366	366	A	66,039	9	151	11.10	-	-	-	1	5.90
367	367	A	66,241	9	151	11.10	-	-	-	1	5.91
368	368	B	78,852	9	151	11.28	-	-	-	1	5.91
369	369	A	72,902	9	151	11.20	-	-	-	1	5.91
370	370	A	71,358	9	151	11.18	-	-	-	1	5.91

Figure 13. Subset of Notional Data Set Up for Regression (PS Methodology)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	294
Equation in Unit Space:	LN_HRS = 12.91 + (-0.1452) * BETA1 + (-0.4264) * BETA2 + (-0.211) * BETA3 + 0.5924 * ALPHA2 + (-0.5247) * ALPHA3

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.9053	0.0335		384.8929	0.0000	1.0000
BETA1	-0.1452	0.0268	-0.0606	-5.4261	0.0000	1.0000
BETA2	-0.4264	0.0052	-2.5485	-82.2160	0.0000	1.0000
BETA3	-0.2110	0.0114	-1.6463	-18.4469	0.0000	1.0000
ALPHA2	0.5924	0.0401	0.8284	14.7761	0.0000	1.0000
ALPHA3	-0.5247	0.0717	-0.7383	-7.3223	0.0000	1.0000

TFU - Leg 1	402,459
Slope - Leg 1	90.4%
Slope - Leg 2	74.4%
Slope - Leg 3	86.4%
TFU - Leg 2	727,760
TFU - Leg 3	238,136

Figure 14. Best Fit Regression – PS Methodology (USAF Only)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	76
Equation in Unit Space:	LN_HRS = 12.89 + (-0.07207) * BETA1 + (-0.437) * BETA2 + (-0.1398) * BETA3 + 0.8028 * ALPHA2 + (-0.7484) * ALPHA3

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.8904	0.1263		102.0407	0.0000	1.0000
BETA1	-0.0721	0.0754	-0.0675	-0.9563	0.3422	0.6578
BETA2	-0.4370	0.0175	-2.2601	-24.9591	0.0000	1.0000
BETA3	-0.1398	0.0358	-0.9557	-3.9088	0.0002	0.9998
ALPHA2	0.8028	0.1467	0.9670	5.4710	0.0000	1.0000
ALPHA3	-0.7484	0.2349	-0.9227	-3.1861	0.0021	0.9979

TFU - Leg 1	396,487
Slope - Leg 1	95.1%
Slope - Leg 2	73.9%
Slope - Leg 3	90.8%
TFU - Leg 2	884,908
TFU - Leg 3	187,584

Figure 15. Best Fit Regression – PS Methodology (USN Only)

Comparing the regression results from our PS and TS runs reveals some interesting contrasts. In particular, the learning slopes for the PS case are all shallower than in the TS case. This contrast is most evident for the USN model, particularly noticeable for its second leg which has flattened from 68.5% to 73.9% and which is within range of the USAF slope (74.4%) over the same range. The USAF slopes are flatter as well, although these changes are on a smaller scale.

Figure 16 shows the derived learning curve projected through the end of Lot 14. The PS method provides a tighter forecast for the B model than does the TS model, while the A model forecast shows the same variance as observed before.

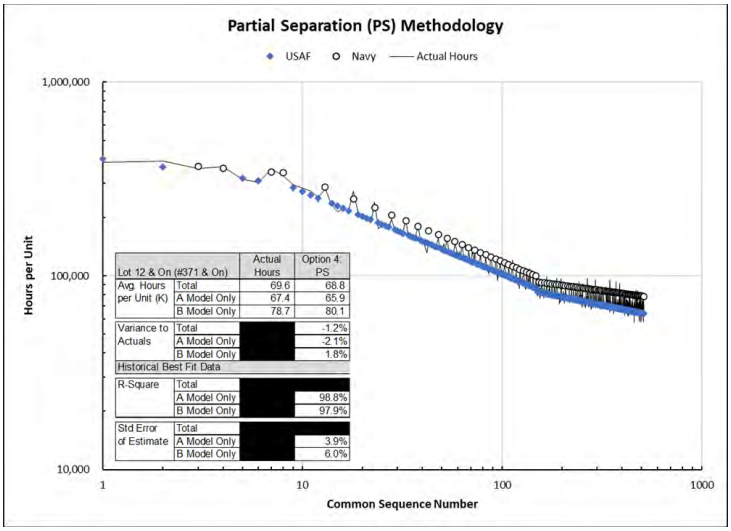


Figure 16. Learning Curve Best Fit & Forecast, PS Methodology

Proportional Representation (PR)

The proportional representation methodology provides a different methodology for calculating learning curve cumulative units. In lieu of using all units or only the units of a singular variant, the PR methodology derives a weighted calculation based on the ratio of common to unique work.

We will briefly take a break from the two-variant notional program we have been examining to explore some of the complexities for a three (or more) variant program. It is for such situations that the PR approach was initially developed. (Garg, 1961)

If we have three variants (A, B and C models), there are seven (7) possible combinations of common and unique work. (During the early days of F-35 development, we joked with our Northrop Grumman and BAE Systems counterparts about the "seven flavors of commonality" -- hence, the title of this paper.) We can have ABC common – that is, work that is common to all three variants. Likewise, we can have AB common – work common to the A and B variants, but which is not common to the C model. Similarly, there is AC common and BC common work. Lastly, there is work that is peculiar to a single variant and has no equivalent (A unique, B unique, C unique).

In general, we can calculate the number of combinations using the formula $2^x - 1$ where x is the number of variants, as illustrated in Figure 17.

Variants	Number
2	3
3	7
4	15
5	31
6	63

Figure 17. Number of Possible Combinations

For a given aircraft or component, how do we calculate the percentage of work that is common and unique? There is no consensus how to do this. Zhang (2019) suggests no less than seven different measures of commonality suggested by previous studies. In the author's experience, he has seen the following approaches suggested:

- Count number of common vs unique engineering drawings (Garg, 1961).
- Count number of common vs unique parts.
- Sum the empty weight of common vs unique parts.
- Sum the Industrial Engineering standard hours of common vs unique parts.
- Engineering judgment based on the similarity or uniqueness of assembly processes and tooling.

An additional complication is that a given part may be highly similar between two or more variants, but not identical. It follows parts or components that are similar should have some degree of learning transfer between variants. To address this, the JAST program, which eventually evolved into the F-35, created the following definitions:

- Common: Physically identical and interchangeable
- Cousin: Same material, function, and interfaces – similar internal geometry, e.g., bulkheads made of identical material, same external dimensions, yet different web thickness and number of penetrations). Made using common fabrication or assembly tooling.
- Unique: Single variant application. (*JAST Commonality Study*, 1996)

Using the empty weight methodology outlined above, the weight of a cousin part was allocated 85% to the common category and 15% to unique. This allocation was done on the assumption that a cousin part would retain most, but not all, of the cost advantages of a common part. (*JAST Commonality Study*, 1996) Imagine two composite skins made from the same graphite material, laid up on a common tool, and sharing the same outer mold line, but with different ply buildups to account for different load patterns. The common characteristics of these parts were judged to be sufficient to allow a high degree of learning transfer between the two versions of the skin.

In the end, the goal is to construct a table like Figure 18, which shows a theoretical example for a product with three variants:

We can also construct a unit sequence table identifying the cumulative number of units which would be built under each commonality "flavor," as shown in Figure 19.

We can now apply a learning curve to each of these commonality combinations utilizing a single learning curve slope but using different theoretical first unit values for each variant (Jones, 2019; Garg, 1961). The results will be something like Figure 20.

	Percent Common to Each Model							
Model	ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique	Total
A	50%	15%	10%		25%			100%
B	50%	15%		5%		30%		100%
C	50%		10%	5%			35%	100%

Figure 18. Notional Commonality Matrix

Model	Common/Unique Build Sequence Number						
	ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique
A	1	1	1		1		
A	2	2	2		2		
B	3	3		1		1	
C	4		3	2			1
A	5	4	4		3		
A	6	5	5		4		
B	7	6		3		2	
C	8		6	4			2
A	9	7	7		5		
A	10	8	8		6		
B	11	9		5		3	
C	12		9	6			3

Figure 19. Cumulative Build Sequences (PR Methodology)

		ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique	Variant Total
Work Content Split	A	50%	15%	10%		25%			100%
	B	50%	15%		5%		30%		100%
	C	50%		10%	5%			35%	100%
T-1 Hours	A	17,500	5,250	3,500	-	8,750	-	-	35,000
	B	22,500	6,750	-	2,250	-	13,500	-	45,000
	C	25,000	-	5,000	2,500	-	-	17,500	50,000

Learning Curve Slope	85%
Learning Beta	-0.23447

Model	Common/Unique Build Sequence Number							Hours per Unit							
	ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique	ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique	Totals
A	1	1	1		1			17,500	5,250	3,500		8,750			35,000
A	2	2	2		2			14,875	4,463	2,975		7,438			29,750
B	3	3		1		1		17,391	5,217		2,250		13,500		38,358
C	4		3	2			1	18,063		3,865	2,125			17,500	41,552
A	5	4	4		3			11,999	3,793	2,529		6,763			25,084
A	6	5	5		4			11,497	3,600	2,400		6,322			23,819
B	7	6		3		2		14,257	4,435		1,739		11,475		31,906
C	8		6	4			2	15,353		3,285	1,806			14,875	35,319
A	9	7	7		5			10,454	3,327	2,218		6,000			21,999
A	10	8	8		6			10,199	3,224	2,149		5,749			21,322
B	11	9		5		3		12,824	4,032		1,543		10,434		28,833
C	12		9	6			3	13,961		2,987	1,642			13,526	32,116

Figure 20. Calculation of Three-Variant Model Using PR Methodology.

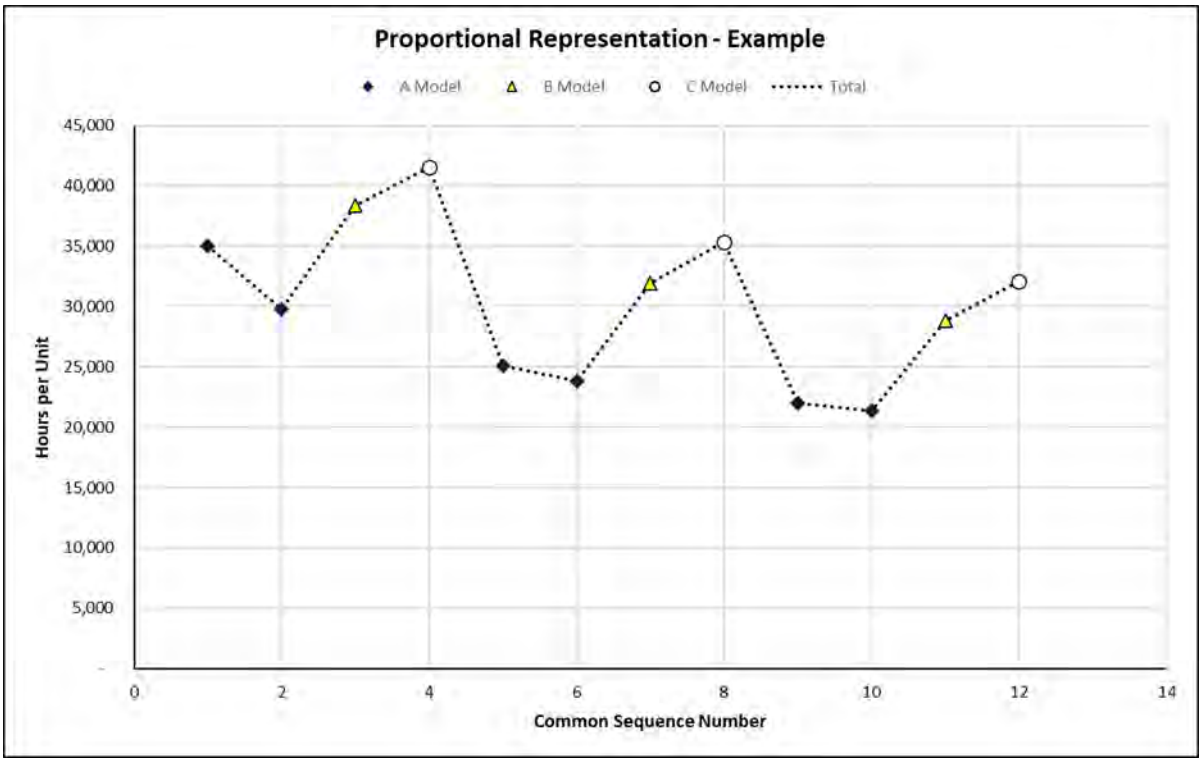


Figure 21. Results of the Three-Variant Model (PR Methodology)

The results of such a calculation can be shown in Figure 21, which shows the sawtooth pattern we have observed from our other commonality methodologies.

The calculations, even in this simple three-variant version, can become quite involved. In addition, it is difficult to derive historical learning curve slopes from actual hours using this method. Jones (2019) suggests that Microsoft Excel Solver can be used to calculate theoretical first unit hours by variant and learning curve slope subject to certain constraints. However, as with any non-linear model, it is possible that Solver will not converge to a solution. In addition, using Solver means that conventional regression statistics such as R-square or p values are not available. The calculations become even more complex when a multi-leg learning curve is introduced.

Fortunately, a simpler approach can be used to develop estimates as well as calculate historical performance[2]. It uses the percentages of common and unique work to calculate an effective sequence number, which will vary depending on which variant is being built. This approach allows the different variants to appear at separate points on the learning curve such that the hours per unit for more unique variants are calculated on earlier segments of the learning curve, resulting in higher hours.

Figure 22 shows the approach. Our commonality matrix shown in Figure 18 is combined such that, for each variant, the percentage learning credit incurred from the other models is calculated. This allows us to say that for each A model, not only does it earn a full unit of learning for each A model built, but it also receives 0.65 unit credit for each B model and 0.60 unit credit for each C model built to that point. The specific amount of credit will depend on the ratio of common to unique work – as the percentage of unique work increases, the learning credit for building the other variants will decrease.

2. Acknowledgments to Kevin N. Curtis, retired Industrial Engineer and Lockheed Martin Fellow Emeritus, who showed me this approach some years ago

Model	Percent Common to Each Model							
	ABC Common	AB Common	AC Common	BC Common	A Unique	B Unique	C Unique	Total
A	50%	15%	10%		25%			100%
B	50%	15%		5%		30%		100%
C	50%		10%	5%			35%	100%

Commonality Matrix			
Model	A Credit	B Credit	C Credit
A	100%	65%	60%
B	65%	100%	55%
C	60%	55%	100%

For Every A Model Built:
Each A Model: A Receives 100% Learning Credit
Each B Model: ABC Common + AB Common = 50% + 15% = A Receives 65% Credit
Each C Model: ABC Common + AC Common = 50% + 10% = A Receives 60% Credit

Model	Cumulative Unit Count			Effective Unit
	A	B	C	
A	1	0	0	1.00
A	2	0	0	2.00
B	2	1	0	2.30
C	2	1	1	2.75
A	3	1	1	4.25
A	4	1	1	5.25
B	4	2	1	5.15
C	4	2	2	5.50
A	5	2	2	7.50
A	6	2	2	8.50
B	6	3	2	8.00
C	6	3	3	8.25

(3 A's x 100%) + (1 B x 65%) + (1 C x 60%) = Effective Unit 4.25

Figure 22. Alternate Methodology for Calculating Effective Unit Sequences

Common Sequence Number	Effective Sequence Number	Model	HPU	Curve Breakpoints		Dependent Variable	Independent Variables					
				T_1	T_2	LN(HPU)	β1	α_2	β2	α_3	β3	B Model Dummy
1	1.0	A	384,354	7.6	139.8	12.86	-	-	-	-	-	-
2	2.0	A	392,722	7.6	139.8	12.88	0.69	-	-	-	-	-
3	2.3	B	359,041	7.6	111.0	12.79	0.83	-	-	-	-	1
4	3.3	B	366,820	7.6	111.0	12.81	1.19	-	-	-	-	1
5	4.3	A	316,530	7.6	139.8	12.67	1.46	-	-	-	-	-
6	5.3	A	303,031	7.6	139.8	12.62	1.67	-	-	-	-	-
7	5.6	B	355,896	7.6	111.0	12.78	1.72	-	-	-	-	1
8	6.6	B	329,786	7.6	111.0	12.71	1.89	-	-	-	-	1
9	7.6	A	294,270	7.6	139.8	12.59	-	1	2.03	-	-	-
10	8.6	A	283,824	7.6	139.8	12.56	-	1	2.15	-	-	-
⋮												
149	137.8	A	87,845	7.6	139.8	11.38	-	1	4.93	-	-	-
150	138.8	A	79,812	7.6	139.8	11.29	-	1	4.93	-	-	-
151	139.8	A	78,318	7.6	139.8	11.27	-	-	-	1	4.94	-
152	140.8	A	81,745	7.6	139.8	11.31	-	-	-	1	4.95	-
153	111.0	B	94,523	7.6	111.0	11.46	-	-	-	1	4.71	1
154	142.5	A	86,816	7.6	139.8	11.37	-	-	-	1	4.96	-
⋮												
366	339.8	A	66,039	7.6	139.8	11.10	-	-	-	1	5.83	-
367	340.8	A	66,241	7.6	139.8	11.10	-	-	-	1	5.83	-
368	265.8	B	78,852	7.6	111.0	11.28	-	-	-	1	5.58	1
369	342.4	A	72,902	7.6	139.8	11.20	-	-	-	1	5.84	-
370	343.4	A	71,358	7.6	139.8	11.18	-	-	-	1	5.84	-

Figure 23. Subset of Notional Data Set Up for Regression (PR Methodology)

I. Model Form and Equation Table

Model Form:	Unweighted Linear model
Number of Observations Used:	370
Equation in Unit Space:	LN_HRS = 12.88 + (-0.1342) * BETA1 + (-0.4215) * BETA2 + (-0.2076) * BETA3 + 0.554 * ALPHA2 + (-0.5313) * ALPHA3 + 0.08574 * B_MODEI

II. Fit Measures (in Fit Space)

Coefficient Statistics Summary

Variable	Coefficient	Std Dev of Coef	Beta Value	T-Statistic (Coef/SD)	P-Value	Prob Not Zero
Intercept	12.8807	0.0355		363.2449	0.0000	1.0000
BETA1	-0.1342	0.0268	-0.0704	-5.0024	0.0000	1.0000
BETA2	-0.4215	0.0054	-2.3133	-77.6938	0.0000	1.0000
BETA3	-0.2076	0.0116	-1.5020	-17.8864	0.0000	1.0000
ALPHA2	0.5540	0.0419	0.7322	13.2168	0.0000	1.0000
ALPHA3	-0.5313	0.0722	-0.7089	-7.3544	0.0000	1.0000
B_MODEL	0.0857	0.0061	0.0941	13.9869	0.0000	1.0000

TFU - Leg 1	392,667
Slope - Leg 1	91.1%
Slope - Leg 2	74.7%
Slope - Leg 3	86.6%
TFU - Leg 2	683,295
TFU - Leg 3	230,835
B Model Factor	1.090

Figure 24. Best Fit Regression – PR Methodology

This methodology has the substantial advantage that it allows the historical learning curve slope to be easily calculated using a single effective unit value for each aircraft and permitting the use of conventional linear regression tools.

Back to our two-variant notional program. In this case, there are only three possible combinations (AB common, A unique, B unique). We will presume that 65% of the aircraft is common between the USAF and USN versions and the remaining 35% is variant unique. Thus, we can develop our effective sequence table in Figure 23. Note as well that we have reintroduced our B model dummy variable that we used in the FF approach. We still need to account for the greater work content of the USN model independent of the learning curve impacts.

Figure 25 shows the derived learning curve projected through the end of Lot 14. The PR method provides a reasonable forecast of Lot 12 and on with a slightly larger variance for the B model. But in this case the variance to actual hours is not quite as good as the FF method. It also has the disadvantage that, even with the simplified calculation of effective sequence numbers, of being the most computationally complex method.

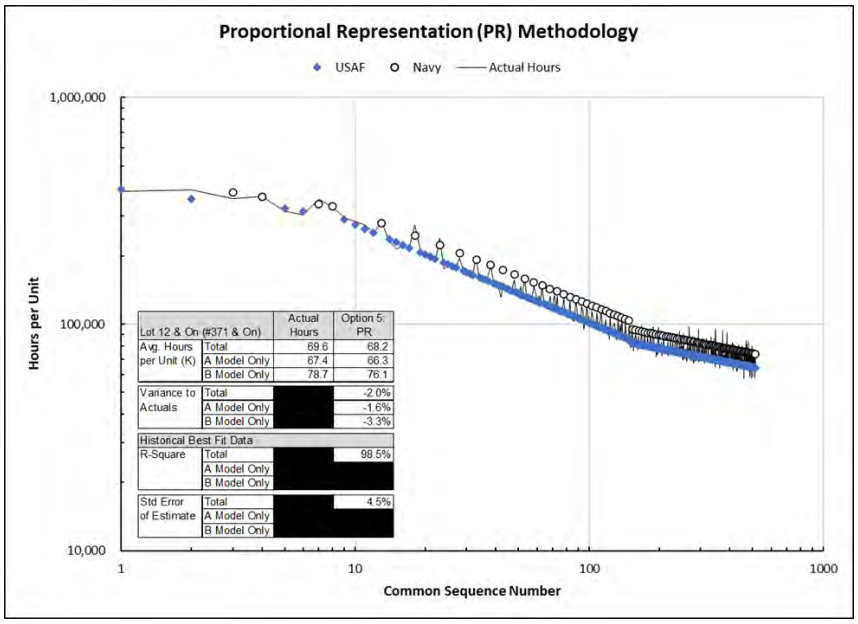


Figure 25. Learning Curve Best Fit & Forecast, PR Methodology

Lot 12 & On (#371 & On)		Actual Hours	Option 1: ID	Option 2: FF	Option 3: TS	Option 4: PS	Option 5: PR
Avg. Hours per Unit (K)	Total	69.6	68.5	68.7	69.3	68.8	68.2
	A Model Only	67.4	68.5	66.5	65.9	65.9	66.3
	B Model Only	78.7	68.5	77.6	82.6	80.1	76.1
Variance to Actuals	Total		-1.6%	-1.3%	-0.5%	-1.2%	-2.0%
	A Model Only		1.8%	-1.3%	-2.1%	-2.1%	-1.6%
	B Model Only		-12.9%	-1.4%	5.0%	1.8%	-3.3%
Historical Best Fit Data							
R-Square	Total		95.9%	98.6%			98.5%
	A Model Only				98.7%	98.8%	
	B Model Only				97.7%	97.9%	
Std Error of Estimate	Total		7.6%	4.4%			4.5%
	A Model Only				3.9%	3.9%	
	B Model Only				6.3%	6.0%	

Figure 26. Comparison of Forecasted Hours

Conclusions

Figure 26 summarizes the five approaches to estimating commonality and shows how different the forward projections can be.

It is worth nothing that from a best fit perspective, four of the five methods have a R-square value of greater than 98% with standard errors around 4%. From this perspective, only the ID method can be clearly rejected.

From the perspective of forecast accuracy, the differences between the methodologies become more apparent. Overall, the TS, FF and PS methodologies show the greatest accuracy at the top level, ranging from -0.5% to -1.3% error. However, the TS methodology shows a higher B model variance (+5.0%) than we would probably like. Of the remaining two methods (FF and PS), the FF option provides the forecast closer to the true actual hours for Lots 12 and on at the individual variant level (-1.3% versus -2.0% for A model and -1.4% versus +1.8% for B model).

This difference is hardly surprising because we "rigged" the notional data to make it so – the hours per unit were generated using an FF approach before introducing a random error to provide a realistic spread of values. Had we generated the data using a different set of premises, another method would probably have produced the best forecast.

The purpose of this demonstration was not to provide proof that one method is always superior to the others. Its purpose is to pilot each method, and to show that the particulars of a program and its build circumstances will dictate which method is the preferred approach.

We might summarize the cases where each methodology might prove more appropriate in Figure 27.

As Figure 27 demonstrates, the appropriateness of using one methodology versus another is highly dependent on the program's peculiar circumstances.

In the introduction, we asked five questions:

1. How common are the airframe engineering designs between the different variants?
2. Do they use a common set of mission and vehicle systems?
3. Will the different variants be built on a common production line, or will they be built on separate production lines, possibly even by different companies?

Methodology	More Appropriate If:	Less Appropriate If:
Ignore Differences (ID)	There is little or no cost difference between variants.	Significant differences in work content exist between variants.
Fixed Factors (FF)	Significant amount of work is common or similar and the probability of learning transfer between variants is high. The cost variance between models is expected to be a fixed ratio in the future, e.g., B models are 10% more costly than A models.	If component or subcomponent is variant-unique (TS may be more appropriate for that item).
Total Separation (TS)	Individual models are produced in different locations or on unique production lines, and the probability of learning transfer between variants is low. A component or subcomponent is variant-unique (FF, PS or PR may be used for the other, more common build areas).	Models are built in the same location and/or same production line with work crews being cycled between models.
Partial Separation (PS)	Significant degree of common or similar work, but reason to believe each variant has a unique rate of learning.	If the elements of learning that are common or similar between variants are high contributors to cost improvement, causing the rate of learning between variants to be roughly equal.
Proportional Representation (PR)	Significant amount of work is common or similar and the probability of learning transfer between variants is high. A fixed cost ratio between models cannot be established from actual cost history, or the relationship of one variant to another is expected to be different in the future.	No suitable *a priori* methodology exists for determining the percentage of common vs unique work.

Figure 27. Comparison of Estimating Methodologies

4. To what extent will the different variants be built using common tooling or manufacturing processes?
5. Will each variant be built using dedicated crews of assemblers? Or will crews be cycled between models as aircraft move down the production line?

It is the answers to these questions – in addition to the traditional best fit regression statistics --that will provide the best guide to the estimator how to proceed.

References

Aboulafia, R. (1996). *Janes Civil Aircraft.* Glasgow, UK: HarperCollins.

About the Boeing 737 MAX (n.d.). Retrieved from https://www.boeing.com/commercial/737max/.

Benkard, C. L. (2000). Learning and Forgetting: The Dynamics of Aircraft Production. *American Economic Review,* 90(4), 1034-1054.

Cochran, E. B. (1960, July-August). New Concepts of the Learning Curve. *The Journal of Industrial Engineering*, 317-327.

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

Engwall, R. L. (2001). Learning Curves. In K.B. Zandin (Ed.), *Manyard's Handbook of Industrial Engineering* (pp. 17.83-17.104). New York, NY: McGraw Hill.

Eurofighter Typhoon. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Eurofighter_Typhoon.

Garg, A., & Milliman, P. (1961, January-February). The Aircraft Progress Curve – Modified for Design Changes. *The Journal of Industrial Engineering*, 7(1), 23-28.

Jefferson, P. (1981, May) Productivity Comparisons with the USA – Where Do We Differ? *Aeronautical Journal*, 85(844), 179-184.

Johnstone, B.M. (2022, November). Projecting Future Costs with Improvement Curves: Perils and Pitfalls. *Journal of Cost Analysis and Parametrics*, 10(3), 64-82.

Johnstone, B. M. (2023, May). *Trouble with the Curve: Engineering Changes and Manufacturing Learning Curves.* 2023 International Cost Estimating and Analysis Association (ICEAA) Professional Development & Training Workshop, San Antonio, TX.

Joint Advanced Strike Technology (JAST) Commonality Study Final Report. (1996, August). Arlington, VA: Joint Advanced Strike Technology Program Office.

Jones, A. R. (2019). *Learning, Unlearning and Re-learning Curves*. London, UK: Routledge.

Jones, A.R. (2001). Case Study: Applying Learning Curves in Aircraft Production – Procedures and Experiences. In K.B. Zandin (Ed.), *Manyard's Handbook of Industrial Engineering* (pp. 17.197-17.213). New York, NY: McGraw Hill.

Kennedy, P. A. (1992). *A Guide to Econometrics* (3rd ed). Cambridge, MA: The MIT Press.

Lorell, M. A., Kennedy, M., Leonard, R. S., Munson, K., Abramson, S., An, R. A., Guffey, R. A. (2013) *Do Joint Fighter Programs Save Money?* MG-1225-AF. Santa Monica, CA: RAND.

Zhang, Y., Yang, Z., Ma, X., Dong, W., Dong, D., Tan, Z., Zhang, S. (2019) Exploration and Implementation of Commonality Valuation Method in Commercial Aircraft Family Design. *Chinese Journal of Aeronautics*, 32(8), 1828-1846. Retrieved from https://www.sciencedirect.com/science/ article/pii/ S100093611930216X.

Brent Johnstone *is a Lockheed Martin Fellow and production air vehicle cost estimator at Lockheed Martin Aeronautics Company in Fort Worth, Texas. He has 38 years' experience in the military aircraft industry, including 35 years as a cost estimator. He has worked on the F-16 program and Advanced Development Programs and has been for 29 years the lead Production Operations cost estimator for the F-35 program. He has a Master of Science from Texas A&M University and a Bachelor of Arts from the University of Texas at Austin.*

My appreciation to Bernie Belley, retired Director of F-35 Estimating & Pricing, who encouraged me to write and submit a paper on commonality.

Analyzing Development Phase NRE/REC Costs in Defense Acquisition Programs

Jason Aristizabal
Jonathan D. Ritschel
Edward D. White
Michael J. Brown
Brandon M. Lucas
Shawn M. Valentine

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Accurately estimating Nonrecurring Engineering (NRE) and Recurring Engineering (REC) costs in defense acquisition programs remains a critical challenge, particularly during the development phase. Although some prior research examines production-phase costs, limited attention has been given to understanding NRE/REC ratios in development efforts. This paper provides novel insights into the development phase by analyzing empirical NRE/REC cost ratios across major Work Breakdown Structure (WBS) elements, commodity types, and time periods.

The findings indicate significant variability in NRE/REC ratios, with System Level, Prime Mission Equipment (PME), and System Test & Evaluation (ST&E) WBS elements exhibiting unique trends. These insights challenge the assumption of a uniform 1:1 NRE/REC ratio, demonstrating that cost structures can shift considerably over time. By highlighting the dynamic nature of NRE/REC ratios, this paper emphasizes the need for cost analysts to move beyond static benchmarks and adopt adaptive methodologies. Tailoring estimates to reflect historical trends and program-specific factors can lead to more accurate and effective resource planning for defense programs.

Introduction

There are multiple cost type classifications defense analysts use when developing a cost estimate. One of the most critical is the delineation of Nonrecurring Engineering (NRE) costs and Recurring Engineering (REC) costs. NRE costs refer to one-time engineering expenses incurred in a program, including research and development, design, testing, and other preparatory activities that do not repeat once production begins (Defense Acquisition University [DAU], 2024a). Conversely, REC costs account for expenses that occur each time a new unit is produced, making them a continuous factor throughout the program's life cycle (DAU, 2024b). To generate reliable cost estimates, cost analysts must delineate these two cost components, as misallocating resources between them can lead to financial inefficiencies, program delays, or funding shortfalls.

This paper has two primary objectives, both centered on improving cost estimation during the development phase of military programs. The first objective is to establish empirically derived NRE-to-REC cost ratios using historical data. A data-driven benchmark for NRE/REC costs, as presented in this paper, could enhance the rigor of early cost estimate validation practices within defense cost estimating organizations. Although cost analysts employ standard-practice estimation techniques, the resulting NRE/REC ratios can vary significantly depending on the available data and chosen methodology. By identifying historical trends, this paper seeks to provide defense organizations with an empirical robustness check to evaluate whether an estimated ratio is plausible within the broader cost landscape of military acquisitions.

The second objective is to enable cost analysts to estimate NRE costs when REC costs are known.

Estimating NRE costs is particularly challenging due to high levels of uncertainty during the early stages of a program (Bradbery, Newnes, & Mileham, 2008). This uncertainty stems from factors such as the introduction of new technologies, limited historical data, and evolving program requirements. Moreover, it is well established that early-stage cost estimates tend to be overly optimistic, often increasing as additional program needs are identified (Prince, 2023). While certain cost differentiators provide insight, initial NRE estimates remain difficult to validate due to these inherent uncertainties.

To the best of our knowledge, a widely accepted, empirically validated NRE/REC ratio does not currently exist within defense cost literature. Some cost practitioners anecdotally observe a 1:1 ratio in their programs, but this assertion lacks systematic validation (S. Valentine, personal communication, February 20, 2024). Interestingly, a similar 1:1 ratio phenomenon appears in various engineering contexts, where system costs are often split evenly between hardware and installation or integration expenses (Lemmens, 2016). For this paper's purposes, the 1:1 ratio serves as an unverified assumption rather than an established rule, offering a reference point for evaluating the empirical ratios derived from historical data.

By systematically analyzing past NRE and REC cost patterns, this paper aims to contribute to the field of defense cost estimation by improving the reliability of early-stage estimates, reducing uncertainty, and providing analysts with a practical tool for evaluating program cost feasibility.

Literature Review

Given the complexity and scale of military programs, accurately estimating NRE and REC costs is essential for effective cost management. NRE costs are influenced by several factors, including system complexity, technological innovation, testing requirements, and program-specific constraints (Lemmens, 2016). These NRE costs are concentrated in the development phase, with the bulk of expenditures occurring before production. While some NRE continues during production for modifications and upgrades, the rate of spending is substantially reduced. Key components of NRE costs include:

- **Research and Development (R&D)** – Activities focused on fundamental research, feasibility studies, and applied engineering (Daley et al., 2021).
- **Initial Design and Prototyping** – The creation and refinement of initial system designs, including physical prototypes (Bradbery, Newness, & Mileham, 2008).
- **Testing and Validation** – Performance evaluations, system certifications, and regulatory compliance testing (Schank, Bodily, & Barbour, 1987).
- **Tooling and Pre-Production Setup** – Costs associated with establishing production facilities and specialized equipment (Hackbart & Covert, 2013).

In contrast, REC costs are incurred repeatedly throughout the production phase and, to a lesser extent, the later stages of development (DAU, 2024). These costs are primarily driven by production scale, operational demands, and technological complexity. Key components of REC costs include:

- **Manufacturing and Assembly** – Material procurement, labor, and direct production costs per unit (Howarth, 2018).
- **Maintenance and Support Costs** – Long-term sustainment activities, including depot maintenance and spare parts (Schank, Bodilly, & Pei, 1986).
- **Operational Costs** – Resource consumption during the system's lifecycle, such as fuel, training, and logistics (Howarth, 2018).
- **Engineering Overhead** – Indirect expenses related to production not specifically tied to a unit of output such as depreciation of equipment or salaries of supervisory engineering personnel (Schank, Bodilly, & Pei, 1986).

Estimating NRE and REC costs accurately has grown increasingly complex due to advancements in defense technologies and evolving acquisition methodologies (Schank, Bodilly, & Pei, 1986). These challenges are compounded further by inconsistencies in how contractors categorize costs, which can distort financial projections. For instance, Galorath Federal (2020) highlights cases where contractors force REC and NRE costs into a single cost element, leading to misclassification and unreliable estimates. This misalignment can obscure

true cost drivers and complicate budget forecasting.

A fundamental aspect of cost estimation is recognizing the interdependence between NRE and REC costs. While NRE expenses are largely front-loaded in a project, they exert a lasting influence on REC costs. Bradbery et al. (2008) differentiate between these two cost categories by emphasizing that REC costs are primarily dictated by production volume, whereas NRE costs stem from initial system design and setup. However, these two cost types are not independent; well-allocated NRE spending can directly reduce REC costs by streamlining production and minimizing inefficiencies (Whitehead et al., 2024). Additionally, strategic NRE investments, such as optimizing tooling and improving system integration, can result in substantial lifecycle cost savings (Bradbery et al., 2008).

A compelling example of effective NRE investment is the U.S. Navy's CVN-21 aircraft carrier program. This initiative allocated substantial NRE funding to pioneering technologies such as an advanced nuclear reactor, electromagnetic launch system, and automated maintenance features (Defense Industry Daily, 2005). These upfront investments were designed to yield long-term benefits, with projections indicating that the CVN-21 program would reduce operating costs by approximately $5 billion over the vessel's 50-year lifecycle. This case illustrates how calculated NRE expenditures can drive significant REC savings, reinforcing the importance of strategic planning in defense acquisitions.

Conversely, the F-111 fighter-bomber program underscores the risks of inadequate NRE planning. The program encountered significant technical difficulties due to the premature integration of the Mark II digital avionics system, an untested technology at the time (Richey, 2005). The decision to implement this system without sufficient validation resulted in a two-year delay and a fourfold increase in NRE costs. Moreover, fundamental design flaws, such as inlet-engine incompatibility and wing structure deficiencies, necessitated extensive redesigns, inflating both REC and long-term sustainment costs (Bradbery, Newnes, & Mileham, 2008). The F-111 program highlights the compounding impact of unresolved NRE challenges on production and maintenance expenditures, demonstrating how insufficient NRE oversight can amplify downstream costs.

The literature underscores that NRE and REC costs are deeply intertwined, with well-managed NRE expenditures leading to more predictable REC costs. Historical case studies demonstrate that effective NRE planning can yield substantial long-term cost reductions, while inadequate NRE allocation can create significant financial liabilities. These insights serve as a foundation for the present paper, which seeks to quantify the relationship between NRE and REC costs across various military programs.

Data

The data comes from Cost Data Summary Reports (i.e. DD Form 1921s) for Acquisition Category (ACAT) 1 programs and from Cost Performance Reports or Contract Funds Status Reports for ACAT 2 and 3 programs. All data was compiled and provided by the Air Force Life Cycle Management Center (AFLCMC), Wright-Patterson Air Force Base, Ohio. The dataset provided by AFLCMC includes detailed information on various development programs, encompassing both NRE and REC costs along with metadata such as program

Category	Contracts	Remaining	Data Points Remaining
All Contracts in Database		368	82574
Non-Final Contract Data	81	287	61311
Cost Data Unavailable	67	220	38769
WBS Information Relevant	5	215	9930
Data Abnormality	68	147	1444

Table 1: Data Inclusion/Exclusion Criteria

type, commodity, branch of service, contractor type, program timelines, and WBS elements. The full dataset consists of 368 different contracts, of which 147 met our criteria for inclusion. See Table 1.

The initial dataset consisted of 368 contracts solely from the development phase, which is the focus of this paper. The first exclusion consisted of 81 contracts with only initial or interim reports (i.e. non-final reports). The second exclusion consisted of 67 contracts that lacked the necessary NRE or REC actual cost data. However, there is one caveat to this exclusion. We kept any programs that lacked the actual cost data if it included the Estimate at Complete (EAC) for NRE and REC costs. This decision was based on the research of White and Tracey (2011), who showed the EAC after the 92.5% completion point was statistically equivalent to the actual costs. As a robustness check, we replicated White and Tracy's (2011) methods with our data and also found statistical equivalency. To summarize, we omitted only those 67 contracts without actual cost data or EAC data past the 92.5% completion point.

The third exclusion consisted of five contracts that did not have the requisite WBS information necessary to complete the analysis, as well as all lines of data that are not relevant to the WBS information we are examining in this paper. The last exclusion consisted of 68 contracts that had data abnormalities. For example, this included data where costs (either NRE or REC) were negative. We also investigated any cost data that recorded a zero for either NRE or REC data. A known issue in collecting NRE vs REC data is that contractors sometimes misclassify data by lumping all data into one category or the other (Galorath Federal, 2020). After applying these exclusion criteria, 1,444 data points remained.

Data Characteristics

The final dataset covers a diverse range of military programs across multiple commodity types, service branches, development classifications, and contractor types. Table 2 displays a breakdown of the data. Note that contractor names are withheld to ensure anonymity.

Methods

We first calculate NRE/REC ratios across various military development programs to assess cost

Commodity Type	**#**	**% of Data**
Aircraft	672	46.5%
Avionics/Radar Systems	6	0.4%
Electronic/Automated Software	285	19.7%
Engine	32	2.2%
Missile	37	2.6%
Rotary Wing	234	16.2%
Space	73	5.1%
UAV	105	7.3%
Branch		
Air Force	571	39.5%
Army	230	15.9%
Joint	15	1.0%
Navy	628	43.5%
Type of Development		
Commercial Derivative	27	1.9%
Modification	240	16.6%
New Design	418	28.9%
Prototype/Experimental	29	2.0%
Subsystem	390	27.0%
Variant	130	9.0%
Unassigned/Not Applicable	210	14.5%
Contractor (Top 5)		
KTR A	362	25.1%
KTR B	300	20.8%
KTR C	269	18.6%
KTR D	105	7.3%
KTR E	52	3.6%
Prime or Subcontractor		
Prime Contractor only	774	53.6%
Prime and subcontractor	670	46.4%
Aircraft Type		
Bomber	43	6.4%
Cargo	76	11.3%
Electronic Warfare	69	10.3%
Fighter	209	31.1%
Recon	239	35.6%
Tanker	19	2.8%
Trainer	17	2.5%

Figure 8: Distribution of NASA TechPort technology types by taxonomy.

distributions and identify significant patterns. We conduct the analysis at four Work Breakdown Structure (WBS) levels: System Level (i.e. total cost), Prime Mission Equipment (PME), System Test & Evaluation (ST&E), and Systems Engineering/Program Management (SEPM). We compute the NRE/REC ratios simply by dividing NRE costs by REC costs. These ratios provide insight into how one-time engineering expenditures compare to recurring costs within the development phase.

Next, we provide measures of cost variation through descriptive statistics. More specifically, we calculate the median, minimum, maximum, mean, standard deviation, and coefficient of variation for each commodity type. This provides a preliminary assessment of how NRE/REC ratios differ across programs.

Our second set of analyses identifies differences in NRE/REC ratios across multiple categorical variables to include: commodity type, branch of service, development type, contractor, prime vs subcontractor classification, and aircraft types (for aircraft programs specifically). Given the potential for non-normal distributions in cost data, we employ non-parametric statistical tests to ensure robust comparisons. Specifically, we use the Kruskal-Wallis (KW) test and the Steel-Dwass (SD) multiple comparison test. The KW test determines whether significant differences exist among multiple categories by comparing NRE/REC ratio distributions. The SD test further identifies which specific groups differ statistically within each categorical comparison, refining insights into cost variability across program attributes.

The final set of analyses investigates how NRE/REC ratios have evolved over time. We use Locally Estimated Scatterplot Smoothing (LOESS) to explore the nonlinear patterns observed in the NRE/REC ratios over time. This non-parametric regression technique fits localized polynomial models to data subsets, enabling flexible trend identification without assuming a predefined functional form. LOESS is particularly useful for detecting subtle shifts that linear models may overlook. By incorporating LOESS models, this paper provides visual confirmation of trends observed in traditional statistical analyses, offering clearer insights into cost evolution across commodity types and WBS elements. For brevity, the focus of the LOESS results is solely on the Aircraft commodity type.

Results

Descriptive Statistics

The descriptive statistics for NRE/REC ratios across WBS elements are presented in Table 3. The values in the minimum and maximum columns indicate a dataset containing a multitude of extreme values. As a result, we suggest the median as the more reliable measure of central tendency due to its' robustness to outliers. The key finding from Table 3 is that the commonly cited 1:1 NRE/REC ratio does not hold universally, though PME is often the closest.

Among the Aircraft related WBS elements, System Level exhibits a median ratio of 2.36, while PME has a lower ratio of 1.6, and ST&E is significantly higher at 5.81. The PME ratio of 1.6 is closest to the 1:1 heuristic noted by practitioners in the field. But the large dispersion in the data shown by the Coefficient of Variation of 13.4 provides a cautionary note.

Beyond Aircraft, the other categories (Electronic/Automated Software, Missiles, Rotary Wing, Space, and UAVs) all show varying trends in NRE/REC ratios. Some of these commodity types demonstrated higher median values than others, which reflects their greater NRE cost burdens. Notably, Missiles exhibit the highest median NRE/REC ratio (9.22 at the System Level), highlighting the uniquely high NRE costs of missile programs relative to REC costs.

Comparative Analyses

We performed KW and SD tests at an alpha of 0.05 to assess the statistical significance of NRE/REC ratio differences across various categorical delineators. The results, summarized in Table 4, reveal key differentiators that significantly impact NRE/REC ratios across System Level, PME, and ST&E WBS elements. Using branch as an example, the numbers in Table 4 are interpreted as follows: a zero for PME indicates no differences between branches; a one for System Level indicates the Air Force and Army exhibit a statistically significant difference from each other; a two for ST&E

WBS Element	N	Median	Min	Max	Mean	Std. Dev.	CV
Aircraft							
System Level	154	2.36	0.00904	4878	36.5	393	10.8
PME	198	1.6	0.00106	108813	575	7735	13.4
ST&E	56	5.81	0.0103	162339	4292	22582	5.26
SEPM	135	3.79	.00511	1827	33.1	170	5.12
Electronic/Automated Software							
System Level	77	2.87	0.0545	12878	175	1467	8.39
PME	91	2.01	0.00349	231	8.82	27.8	3.15
ST&E	22	4.25	0.0281	497	56.7	127	2.23
SEPM	75	4.12	0.101	2821	95.4	380	3.98
Missile							
System Level	9	9.22	0.00146	17.8	8.89	6.44	0.72
PME	10	6.71	0.373	38.3	9.57	10.8	1.13
SEPM	14	23.9	0.0155	380	75.2	108	1.43
Rotary Wing							
System Level	61	2.54	0.131	256	8.66	33.7	3.89
PME	64	1.93	0.119	221	10.8	34.5	3.2
ST&E	41	1.48	0.298	340	15.9	56.5	3.56
SEPM	44	6.22	0.182	557	48.8	102	2.09
Space							
System Level	14	2.79	0.463	39.1	5.93	9.89	1.67
PME	24	1.18	0.0103	16.4	1.9	3.22	1.69
ST&E	5	1.69	0.191	4592	954	2035	2.13
SEPM	24	0.713	0.283	0.0113	4.08	10.6	2.58
UAV							
System Level	15	1.06	0.000009	15	3.24	4.76	1.47
PME	36	0.904	0.00198	97.4	7.09	20.5	2.89
ST&E	11	3.28	0.579	286	40.8	88.2	2.16
SEPM	27	2.34	0.0009	511	29.9	98.4	3.29
Note: Only four WBS elements are shown, which results in sample size differences from Table 1.							

Table 3: NRE/REC Descriptive Statistics

indicates that the Air Force is statistically different from both the Army and the Navy.

The results in Table 4 show two of the six differentiators (prime versus subcontractor and development type) have significant differences in only one WBS element, indicating that they are not frequent delineators. Notably, aircraft type is the only differentiator with significant differences in all three WBS elements. Analyzing across WBS elements, ST&E shows the most differences overall, reflecting its highly variable cost structure across programs, followed by PME. System Level, while having the fewest differences, still demonstrates statistical significance for branch of service and aircraft types, reinforcing the importance of these differentiators in cost analysis.

Given the importance of contractor selection in military procurement, this paper specifically

	Commodity	Branch	Development Type	Contractors	Prime/Sub	Aircraft Types
System	0	1	0	0	0	2
PME	2	0	0	0	1	2
ST&E	1	2	3	0	0	1

Table 4: Non-Parametric Testing Delineators

examined how NRE/REC ratios vary across different prime contractors. We conducted the KW and SD tests on the five largest contractors in the dataset, as these firms represent the primary entities responsible for executing major defense contracts. The focus was to determine whether these dominant contractors significantly influenced cost structures relative to one another. Additionally, to ensure a comprehensive evaluation, we conducted a secondary test in which all smaller contractors were grouped into a single category and compared against the five major firms. This approach aimed to assess whether smaller firms collectively exhibited statistical differences in NRE/REC ratios compared to the dominant contractors. However, results from both tests indicated no significant findings, suggesting that the selected contractor does not introduce substantial variation in NRE/REC ratios.

The presence of these statistical differences underscores the importance of tailoring cost estimation approaches to specific program attributes. A one-size-fits-all method risks overlooking the substantial cost variability introduced by service branch, development type, and type of commodity. These findings highlight the necessity for cost analysts to account for program-specific delineators when developing NRE/REC ratio-based cost estimates.

Time Analysis

The final analysis seeks to understand how NRE/REC ratios have evolved over time. The LOESS regression models (Figures 1 & 2) provide insights into how cost structures have fluctuated over decades, capturing nonlinear trends that would be missed by traditional statistical tests.

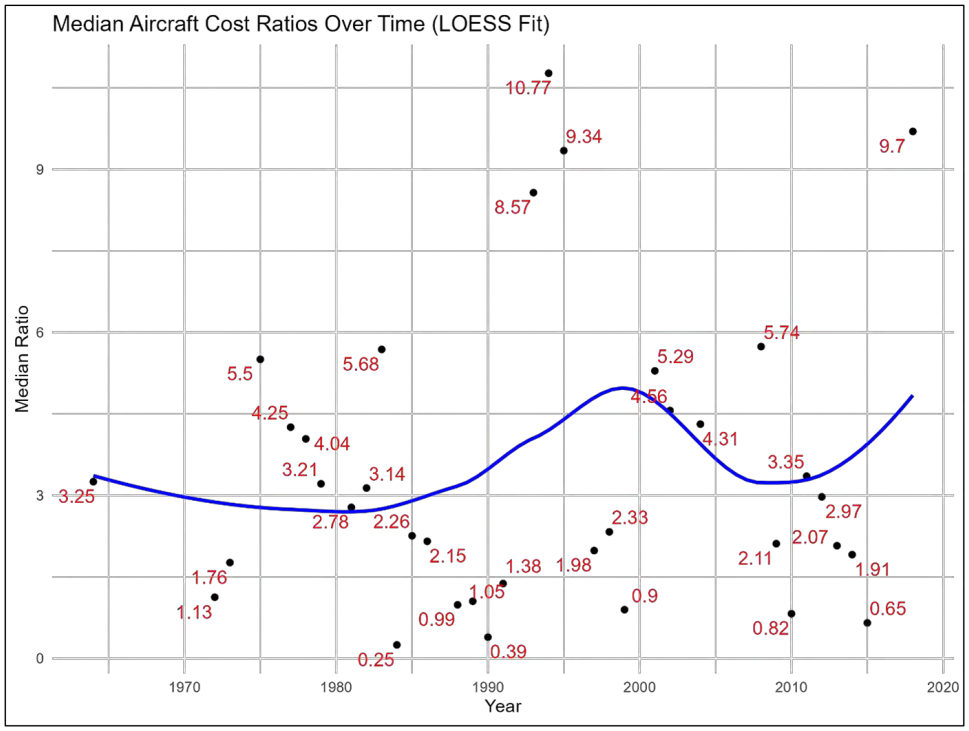


Figure 1: LOESS Model Aircraft System Level

Figure 1 reveals a distinct oscillation pattern in Aircraft System Level NRE/REC ratios over time. From the 1970s to the early 1980s, the ratio remains relatively stable with a slight downward trend. However, the late 1980s to 2000 saw a significant increase in relative NRE costs, potentially linked to technological advancements, integration complexities, and policy shifts. After peaking near 2000, the trend reverses, suggesting a decline in relative NRE burdens until the late 2010s, at which point a new upward shift emerges**.** To further explore how these trends extend to PME costs, Figure 2 examines Aircraft PME NRE/REC ratios over time.

While PME follows a similar oscillatory pattern, the fluctuations are more gradual. The 1970s to mid-1980s show relative stability, followed by a slight downturn that reverses in the early 1990s. Unlike System Level costs, PME ratios do not exhibit sharp spikes or declines but instead maintain a slow upward trend over time.

The lack of dramatic shifts in PME ratios may reflect greater consistency in program management and engineering costs, potentially due to the standardization of engineering methodologies and the gradual adoption of Model-Based Systems Engineering (MBSE). Although MBSE impacts were not explicitly confirmed by statistical tests, the LOESS curve suggests a potential influence, particularly in the 2000s and beyond, where PME costs appear to stabilize.

These temporal insights emphasize that NRE/REC cost structures are not static. They evolve in response to technological, policy, and programmatic changes. Understanding these shifts can provide cost analysts with critical insight when forecasting future program costs and assessing the viability of early cost estimates.

Discussion

The findings of this paper have implications for cost estimation practices, offering practical insights that enhance the accuracy and robustness of defense program cost analyses. This research served a twofold purpose: (1) establishing empirically based NRE/REC ratios to provide a robustness check for validating cost estimates, and (2) equipping cost analysts with a structured approach to estimating NRE costs when REC costs are known.

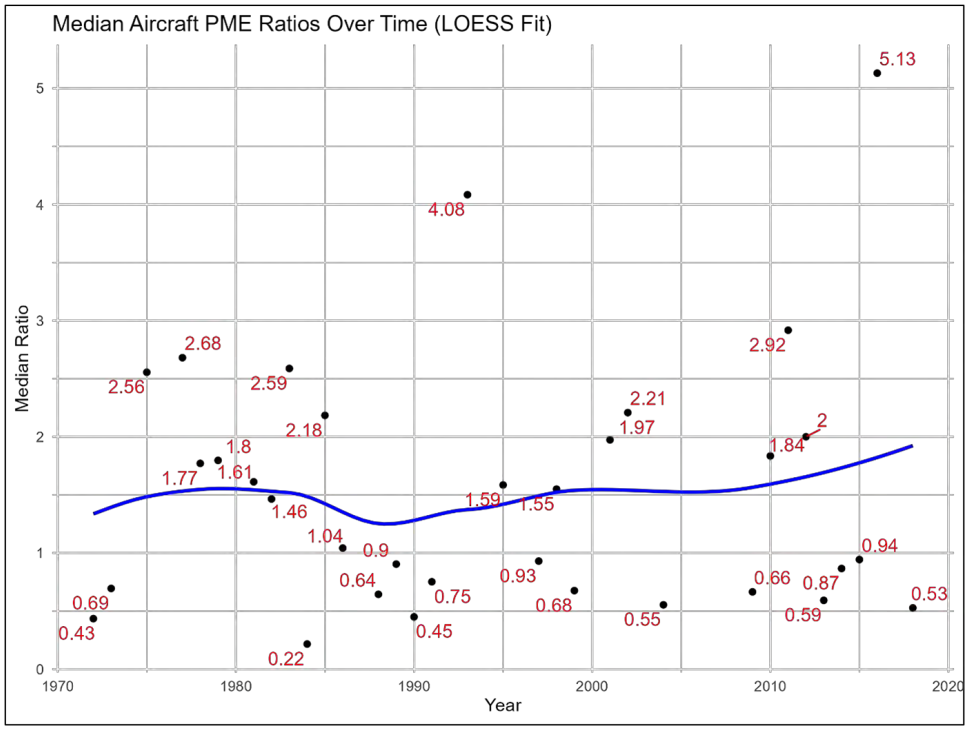


Figure 2: LOESS Model Aircraft PME

During the course of our research, we found a widely referenced assumption of a 1:1 NRE/REC ratio by defense cost analysts. The analysts primarily referenced the 1:1 assumption for PME, suggesting for every dollar spent on recurring costs, an equal dollar is spent on nonrecurring costs. While anecdotal observations suggest this ratio has appeared in some defense programs, and similar trends exist in other engineering and construction contexts, this paper's findings challenge the assumption of a universal ratio.

Across all WBS elements, the empirical data reveals notable deviations from the 1:1 expectation, demonstrating significant variation in NRE/REC ratios based on program type, commodity, and timeframe. The results indicate no consistent trend aligning with a 1:1 ratio, suggesting that relying on this assumption may lead to inaccurate cost projections. Instead, this paper emphasizes the necessity for cost analysts to develop program-specific estimates, using historical trends and data-driven insights rather than broad, generalized assumptions.

Although not an initial objective of this paper, we found an unexpected issue in the data that provides an opportunity for improved data classification/ reporting. The issue is the prevalence of REC cost entries within the dataset recorded as zero cost. A total of 2,734 entries reported a REC cost of zero, raising concerns about potential data misclassification. To investigate this issue, we conducted an in-depth audit of 673 aircraft-related entries, revealing that 529 of these entries had associated quantities. To have a quantity, but zero REC costs is illogical. It suggests they were likely misclassified, with REC costs incorrectly reported as NRE costs.

This finding underscores the importance of data validation in cost estimation. While we only conducted this verification process for aircraft-related entries, the broader dataset excluded all zero REC entries across other commodities, potentially removing data points that should have been reclassified rather than omitted. Moving forward, cost analysts should exercise caution when interpreting zero REC values, as these could reflect reporting inconsistencies rather than genuine cost structures. Addressing these discrepancies will require improved guidance for contractors on correctly categorizing NRE and REC costs, along with standardized data validation protocols.

In conclusion, this paper provides novel insights through empirically derived NRE/REC development phase ratios. By testing across various delineators, our findings reinforce the importance of tailored cost estimation approaches, demonstrating that NRE and REC cost relationships are not static but instead shift based on program characteristics or external factors such as technological advancements and procurement strategies. Additionally, this paper demonstrates how NRE and REC cost relationships vary across key differentiators, such as commodity type, service branch, development phase, and WBS elements. Lastly, we provide an exploratory examination of the ratios over time. While we were unable to definitively ascertain the impact of changes like MBSE, we recommend that future researchers explore the impact of these phenomena as more data becomes available.

References

Arnold, D. (2020). *Missile Systems Sufficiency Review Handbook Update.* Tecolote Research, Inc.

Bradbery, R., Newnes, L., & Mileham, A. (2008). Estimation of non-recurring costs in manufacturing. *SCEA-ISPA Joint Annual Conference and Training Workshop.* University of Bath, Department of Mechanical Engineering. https://www.iceaaonline.com/wp-content/uploads/2017/09/MM11.pdf

Defense Acquisition University. (2024a). Retrieved 2024, from Non-Recurring Engineering Cost: https://www.dau.edu/glossary/non-recurring-costs

Defense Acquisition University. (2024b). Retrieved 2024, from Recurring Engineering Cost: https://www.dau.edu/glossary/recurring-cost

Daily, D. I. (2005). *Costing the CVN-21: A DID Primer*. Defense Industry Daily. https://www.defenseindustrydaily.com/costing-the-cvn21-a-did-primer-01624/

Daley, C. S., Southwell, A., Gayatri, R., Biersdorff, S., Toepfer, C., Özen, G., & Nicholas, W. (2021). Non-Recurring Engineering (NRE) Best Practices: A Case Study with the NERSC/NVIDIA Open MP Contract. Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. https://dl.acm.org/doi/pdf/10.1145/3458817.3476213

Federal, G. (2020). *Wheeled and Tracked Vehicle Cost Estimating Bluebook.* Alexandria: Galorath Federal.

Hackbarth, L., & Covert, R. (2013). A Novel Non-Recurring Production CER Methodology. *ICEAA Professional Development & Training Workshop.* https://www.iceaaonline.com/2013-m106/

Howarth, D. K. (2018). Demand, Recurring Costs, and Profitability. *ICEAA Professional Development & Training Workshop.* Denver: International Cost Estimating and Analysis Association (ICEAA). https://www.iceaaonline.com/wp-content/uploads/2018/07/EA12-Demand-Recurring-Costs-and-Profitability-Howarth.pdf

Lemmens, S. (2016). Cost engineering techniques and their applicability for cost estimation of organic Rankine cycle systems. *Energies*, 9(7): 485. DOI: https://doi.org/10.3390/en9070485

Prince, A. (2023). The Psychology of Cost Estimating. *Journal of Cost Analysis and Parametrics*, 11(1): 5-23.

Richey, K. G. (2005). *F-111 Systems Engineering Case Study.* Air Force Center for Systems Engineering, Dayton OH: Air Force Institute of Technology. https://scholar.afit.edu/docs/38/

Schank, J. F., Bodilly, S. J., & Pei, R. Y. (1986). *Unit Cost Analysis-Annual Recurring Operating and Support Cost Methodology.* Santa Monica, CA: RAND Corporation. https://www.rand.org/pubs/reports/R3210.html

White, T., & Tracy, S. (2011). *Estimating the Final Cost of a DoD Acquisition Contract.* Journal of Public Procurement, 11(2): 190-205. DOI: https://doi.org/10.1108/JOPP-11-02-2011-B002

Whitehead, P. N., Luna, A., Mignano, J., & Light, T. (2024). *Framework for Assessing the Costs and Benefits of Digital Engineering: A Systems Approach*. Santa Monica, CA: RAND Corporation. https://www.rand.org/pubs/research_reports/RRA2418-1.html

---

*Capt **Jason Aristizabal** is currently assigned as a cost analyst at the Life Cycle Management Center/T-7 at Wright-Patterson Air Force Base (AFB), OH. Capt Aristizabal is a recent graduate of the Air Force Institute of Technology (AFIT) where he earned an MS in Cost Analysis.*

*Dr. **Jonathan D. Ritschel** is a Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, MS in Cost Analysis from AFIT, and PhD in Economics from George Mason University. His research interests include public choice, defense cost economics, cost analysis and economic institutional analysis.*

*Dr. **Edward D. White** is a Professor of Statistics in the Department of Mathematics and Statistics at AFIT. He received his BS in Mathematics from the University of Tampa, MAS from The Ohio State University, and PhD in Statistics from Texas A&M University. His primary research interests include statistical modeling, simulation, and data analytics.*

*Major **Michael J. Brown** is an Assistant Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BS in Economics from the United States Air Force Academy, a MS in Cost Analysis from AFIT, and a MA and PhD in Economics from the University of Tennessee. Major Brown's research interests include recruitment and retention, econometric analysis, and governmental acquisitions.*

Dr. ***Brandon M. Lucas*** *is an Assistant Professor of Systems Integration and Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BA in History from the University of Texas at Austin, an MS in Cost Analysis from AFIT, and a PhD in Economics from George Mason University. Lucas's research interests include profit analysis, cost and economic analyses, and incentive structures.*

Mr. ***Shawn M. Valentine*** *is an Operations Research Analyst at AFLCMC/FZCE at Wright Patterson AFB, Ohio. Mr. Valentine graduated from Ohio University with a Bachelor of Science (BS) in Actuarial Science and a Master of Science (MS) in Financial Economics.*

The Impact of Advancing Technology on Fire Control Radar Costs

Dustin M. Brewer
Michael J. Brown
Jonathan D. Ritschel
Robert D. Fass

Radar systems are integral components of fighter aircraft, gaining heightened importance with the emergence of fifth-generation variants. Among these systems, fire control radar plays a crucial role by providing pilots with unparalleled long-range sensing capabilities vital for navigation and target identification. The rapid evolution of radar technology is crucial for aligning with changing mission requirements and is expected to significantly influence radar system costs. We investigate the influence of advancing technology on fire control radar development and production costs. To accomplish this objective, we categorize radars into generations based on existing literature and conduct a comparative analysis between active electronically scanned array (AESA) and non -AESA radars, as well as between modification and non-modification radar programs.

This research shows that costs have remained consistent over time for both development and production. Moreover, no statistical difference is observed between AESA and non-AESA radars. Modification programs are found to be less expensive to develop than non-modification programs, although no significant difference is found in production costs. This research serves as a valuable resource for project managers and cost analysts, offering insights to enhance cost estimates for fire control radar programs.

Introduction

Radar systems are indispensable components of fighter aircraft, gaining heightened significance with the advent of fifth-generation variants. Fire control radar assumes a pivotal role by providing pilots with long-range sensing capabilities crucial for navigation and target identification. Modern aerial combat objectives extend beyond defeating adversaries; fifth -generation fighters aim to outmatch and conquer enemy aircraft while safeguarding allies through advanced technologies. Key technical features of fifth-generation fighters include enhanced stealth, extended combat range, heightened situational awareness, and groundbreaking airborne weapons. These advancements impose greater demands on fire control radar systems, necessitating rapid technological progress to align with mission requirements, significantly impacting radar system costs (Stern, Dryden, & Balakrishnan, 2008; Xiao et al., 2021).

As a result, a non-trivial portion of the defense budget is allocated towards designing, manufacturing, upgrading, and maintaining these systems (Grossman et al., 2019). This study investigates the influence of evolving technology on fire control radar development and production costs. Radars are categorized into generations based on existing literature, and a comparative analysis is conducted between active electronically scanned array (AESA) and non-AESA radars, as well as modification and non-modification radar programs.

This study aims to investigate potential differences in development and production costs between AESA and non-AESA radars to determine whether advancing technology and capability comes at growing cost. Additionally, this study analyzes cost differences in development and production between new and modified radar programs. The findings of this study will aid program managers and cost analysts in the development of appropriate, defendable program baselines by identifying the relationship between technology evolution and that of cost growth. This currently unsettled question introduces uncertainty and potential bias into radar program baselines which undermines the

effectiveness and efficiency in the overall weapon system portfolio.

BACKGROUND

History

Ever since WWII, radio, detection, and ranging (radar) has played a crucial role in modern warfare. The German Luftwaffe first demonstrated the ability to use radar in aircraft, cementing its place as a critical air superiority enabler (Foley, 2011). Since that proof of concept nearly 80 years ago, airborne radar technology has evolved significantly, enhancing situational awareness and enabling precise target acquisition, tracking, and weapon deployment in both air-to-air and air-to-surface attack scenarios (Younus & Manarvi, 2010).

The 1960s introduced a leap in radar technology with phased arrays and electronically directed beams (Grossman, et al., 2019; Strong, 2005; Xiao, et al., 2021). The 1970s continued this evolution with the introduction of synthetic aperture radars which enabled near real-time processing of airborne images. The 1970s also saw hardware enhancements that led to radar with greater power, more frequencies, and reduced sizes (Grossman, et al., 2019). Within 30 years of the Luftwaffe's first use of aerial radars in WWII, the US fielded the AN/AWG-9 in the early 1970s. This marked the first practical airborne pulse-Doppler fire control radar which debuted on the F-14 aircraft (Xiao, et al., 2021).

In the 1990s, solid-state transmitters like gallium arsenide (GaAs) semiconductor devices further revolutionized radar operations by integrating these transmitters behind individual radiating elements in phased-array antennas, rendering high-powered transmitters no longer necessary (Grossman, et al., 2019). Instead, numerous lower-powered transmit/receive (T/R) modules were used, allowing radar beams to change direction and waveform between pulses. This innovation gave rise to active electronically scanned array (AESA) radars (Stern, Dryden, & Balakrishnan, 2008, Grossman, et al., 2019).

AESA technology has continued to evolve, particularly with the introduction of gallium nitride (GaN) T/R modules. This has significantly increased output power and efficiency. Enhanced computational capabilities have expanded the applications of AESA radars significantly as well. Modern AESA radars now boast abilities such as the generation of expansive synthetic aperture radar (SAR) images, concurrent SAR and ground moving target indicator (GMTI) modes, multiple target tracking, and dynamic space-time adaptive processing (Grossman et al., 2019). AESA radars stand out for their extended operational range, multifunctional versatility, heightened reliability, and reduced interception risk (Mishra, 2018). This evolution has seamlessly integrated AESA into fifth-generation and modern aircraft designs, enabled the retrofitting of older aircraft, enhanced radar capabilities for precision weapon delivery, minimized collateral damage, and improved overall weapon system effectiveness (Stern, Dryden, & Balakrishnan, 2008; Strong, 2005).

As the DoD has achieved new milestones with fire control radar, the continuous enhancement of radar technology has remained a priority as adversaries continue to innovate, exemplified by Russia's Sukhoi Su-57 and China's Chengdu J-20. The rapid progress in electromagnetic countermeasures technology has negatively influenced radar detection and tracking performance, posing a formidable threat to the survivability of aircraft. Moreover, the ongoing advancement of aircraft stealth technology has elevated the requirements for fire control radars, demanding superior levels of detection accuracy, resolution, and recognition capabilities (Xiao, et al., 2021).

Technology Growth—Cost Savings or Growth?

Experts widely accept that the United States must continue developing more powerful and sophisticated radars. What is unclear, however, is how the technological advances will change costs, if at all. Renaud (2013) voices concerns regarding the escalating research and development (R&D) spending, noting its tendency to yield only incremental innovations. Similarly, Kirkpatrick (2004) emphasizes the growing dominance of fixed costs within weapon programs. This trend can be attributed to three key factors: As the unit purchase costs escalate, the fixed cost components of

successive projects increase proportionally. Secondly, modern weapon systems rely heavily on complex software, necessitating extensive development and testing efforts. Lastly, the increasing integration of weapon systems necessitates comprehensive development and testing to ensure seamless operation alongside other legacy systems in a nation's inventory. Consequently, these three factors contribute significantly to the rising trend of fixed/variable cost within projects (Kirkpatrick, 2004).

Greer Jr. & Moses (1992) conducted a study involving 18 USAF satellite programs, yielding insights that underscore the role of product technology as a substantial cost driver. The authors find that complexity, when coupled with time as an intervening factor, effectively explains development costs. This observation aligns with the perspective of those who categorize product complexity as a primary determinant of cost.

In contrast, common theories assert that advancing technology naturally reduces costs. For example, the ongoing refinement of computers consistently makes them more affordable and efficient with each iteration. Gordon Moore (1975), a co-founder of Intel, projected that the number of transistors per integrated circuit would roughly double every 18 months, a concept later known as Moore's Law (Jovanovic & Rousseau, 2002). This perspective argues that increased efficiency often results from the cumulative expertise manufacturers gain during the production and sales cycle of a product (Moore, 1975). Empirically, investments in R&D enhance both firms' profitability and overall efficiency (Jovanovic & Rousseau, 2002).

Since their inception, integrated circuits' complexity has witnessed an approximate doubling on an annual basis. Concurrently, the cost per function has undergone a remarkable reduction, decreasing by several orders of magnitude, all while manifesting substantial advancements in both system performance and reliability (Moore, 1975). This pattern is echoed across various technologies, which show exponential progress in line with Moore's Law but at a more measured pace, as seen in industries like beer production, offshore gas pipelines, and aviation (Nagy, Farmer, Bui, & Trancik, 2013; Koh and Magee, 2006).

The semiconductor industry stands as a prime example of the profound impact of technology on cost reduction. Driven by the principles of Moore's Law, the semiconductor industry has exhibited unparalleled expansion by delivering enhanced capabilities at reduced costs (Chien, et al., 2007). Through continuous innovation and refinement of manufacturing processes, semiconductor manufacturers have been able to pack more transistors onto integrated circuits, resulting in improved performance while maintaining or lowering costs. Furthermore, ongoing developments in the private and commercial semiconductor sector, as explored by Grossman, et al. (2019) and initiatives by the DoD, highlight the determination to drive down costs while simultaneously enhancing performance in this dynamic industry. These efforts are particularly significant for major radar acquisition programs, given the intrinsic link between the performance and costs of AESA radars and the capability of T/R semiconductor modules.

However, the trend of capability boons at lower costs may have peaked. Kim, et al., (2020) affirms that the integration and downscaling of semiconductor transistors in accordance with Moore's law have catalyzed profound enhancements in information technology over the past five decades. Nevertheless, the escalating power consumption has pushed Moore's law to its boundaries, thereby inhibiting the further downsizing of transistors. This could impact the costs of AESA radars and future major radar acquisition programs. To enable the sustained advancement of circuit technologies, substantial reductions in both standby and switching power consumption levels are imperative. This notion is further supported by the initiatives of the Defense Advanced Research Projects Agency (DARPA) and DoD, as they actively pursue enhancements in AESA technology by utilizing GaN monolithic microwave integrated circuits (MMICs) to achieve higher output power (Grossman, et al., 2019).

As this brief review of the literature shows, whether cost growth will accompany future technological advancement is ambiguous. Advocates exist on each side of the debate and each side is arguably justified in their position. Unfortunately, this ambiguity does not benefit program management and cost professionals in their attempt to establish an accurate

program baseline. This ambiguity undermines portfolio management accuracy and leads to inefficient tradeoffs between capability, cost, and schedule. These tradeoffs ultimately result in late or subpar products for the warfighter or increased costs to the US Taxpayer. (US GAO, 2021). Thus, a comprehensive empirical examination is needed to better inform future radar estimates.

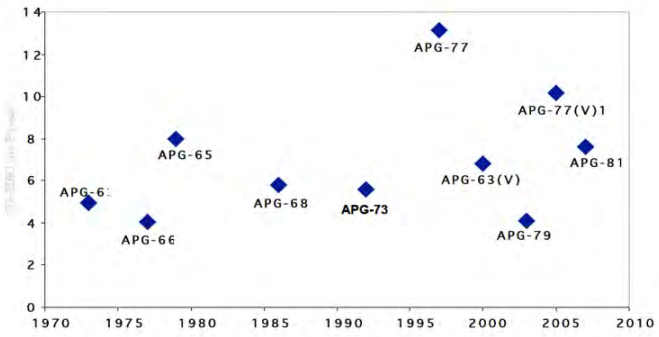


Figure 1. Fire-Control Radar 100th-Unit Cost per Pound vs. First Production Year. Adapted from "A Cost, Technical, and Industrial-Base Review of Select Airborne Radars" by A. Stern, J. Dryden, & S. Balakrishnan, 2008, RAND Corporation

Previous Radar Cost Studies

In their studies for the RAND Corporation, Stern, Dryden, & Balakrishnan (2008) and Grossman et al. (2019) assert that estimating costs for new radar acquisition programs has become increasingly challenging because of the constant evolution of fire control radars. These challenges primarily arise from the lack of comprehensive data covering developmental complexity and associated costs. Despite the data difficulties, Stern, Dryden, & Balakrishnan (2008) examined the influence of technological evolution on the fundamental characteristics of radar costs. The findings are illustrated in Figure 1, which presents the fire-control-radar cost per pound at the 100th unit against the first production year. Notably, the cost per pound exhibited a discernible upward trend, albeit with increased variability. It is important to note that this analysis excluded the APG-79 due to its weight and the APG-77 and APG-(77)V1 due to low production rate. The overarching inference drawn from this study indicates a relatively consistent trend, suggesting that, on average, technological advancements have not fundamentally reshaped the cost structure of production. This observation is noteworthy, considering Moore's Law would imply cost reductions with technological advancements and increased capability. However, it is worth acknowledging that this study presents certain limitations, including a limited sample size comprising only seven fire control radars, with the excluded radar systems taken into consideration. Moreover, employing the cost per pound metric raises concerns about the applicability of the findings, given that both the 2008 and 2019 RAND studies concluded that weight serves as an inadequate predictor of production costs (Stern, Dryden, & Balakrishnan, 2008; Grossman, et al., 2019).

Adding to the complexity of the analysis, most fire control radar programs incorporate elements from previous projects. Modified radar programs, which leverage more technology from prior endeavors, offer potential cost savings and more accurate cost estimates. For example, the APG-63(V)3 program retains largely unchanged or only marginally modified back-end components compared to the APG-63(V)1 or APG-63(V)2 models, with primary enhancements focused on the radar's front end (Stern, Dryden, & Balakrishnan, 2008; Grossman et al., 2019). This suggests that modified programs can be more cost-effective in development and production compared to entirely new programs.

While the previous RAND studies provide valuable information regarding the evolution of radar production costs, the small production data set and lack of development data restrict the ability to draw firm conclusions. This study builds upon the previous RAND studies by expanding the production data set, introducing development analysis, and exploring the costs of modified programs versus new programs in both development and production. This study constitutes a novel and comprehensive analysis on the cost growth of radar systems throughout the last several decades and provides evidence regarding the trends that program managers and cost professionals should expect for future programs.

METHODS

Data

The Air Force Research Laboratory (AFRL) and the Air Force Life Cycle Management Center (AFLCMC) supplied the data for this analysis. They provided technical and cost information for 64 radars. The information was sourced from DD Form 1921s and the AFLCMC/FZC Cost Library. All data were normalized in Constant Price (CP) 2022 and Then Year (TY) dollars by AFLCMC. All dollar figures reported in this study are in CP$22.

Initially, the dataset contained 64 radars of various types, including Surveillance, Air & Missile Defense, Airborne Fire Control, Airborne Multipurpose/Special, Airborne Countermeasures Multi-purpose, and Airborne Surveillance. We excluded 31 observations to focus solely on Airborne Fire Control radars. We then reviewed each remaining observation for completeness in developmental costs, removing seven additional data points. Three more radars are excluded due to missing total weight when the analysis requires weight as an independent variable. Consequently, the final developmental cost dataset comprised 23 or 26 radars depending on the type of analysis. Table 1 outlines the exclusion criteria and the corresponding number of data points used for the developmental analysis.

Category	Number Removed	Number Remaining: Cost Trend Analysis	Number Remaining: AESA/Mod Delta
Initial Data Set	0	64	64
Not Fire Control Radar	31	33	33
Missing Development Costs	7	26	26
Missing Total Weight (for cost trend analysis only)	*3*	23	26

Table 1. Development Data Set Exclusions

For the production data set, we removed all non-airborne fire control radars (31 data points). We then reviewed the remaining 33 fire control radar data points for completeness, excluding 19 due to missing production cost information. One radar is excluded for missing total weight when the analysis requires weight as an independent variable. Finally, one data point was excluded as an outlier. Upon investigation, this specific program functions as an all-in-one advanced electronic countermeasure system rather than a traditional fire control radar significantly increasing both cost and weight. As this system is unlike all other systems, it should not be considered a "fire control radar" even though this function is included with the system. This resulted in a final production dataset of 11 or 12 radars depending on analysis. Table 2 outlines the exclusion criteria and the corresponding number of data points used for this research.

Category	Number Removed	Number Remaining: Cost Trend Analysis	Number Remaining: AESA/Mod Delta
Initial Data Set	0	64	64
Not Fire Control Radar	31	33	33
Missing Production Costs	19	14	14
Missing Total Weight (for cost trend analysis only)	1	12	13
Verified Outlier	1	11	12

Table 2. Production Data Set Exclusions

Data Characteristics

A key focus of this study is our evaluation of the impact of technology, AESA, and modified radars on both development and production costs. To achieve this, we concentrate on three identifiable

variables within the data. The first variable, termed "generations," plays a central role in our analysis, serving as a proxy for technological progression over time. This variable captures the evolutionary advancements made in radar technology. Each generation represents a period of ten years. The period selection is supported by radar programs consistently leveraging previous efforts and the previous advancements such as those noted earlier in the history section. The second variable in our study represents radars equipped with an AESA. This variable serves to identify advanced radar systems, highlighting their electronically steered beam agility and heightened tracking precision. The third variable focuses on modification radar programs and identifies them explicitly. These programs are differentiated by distinct version names within the same radar program, signifying tailored adaptations and enhancements. Tables 3, 4, and 5 provide a detailed breakdown of the three variables used in the development cost analysis for this study.

Generations Variable	Development Year Range	Number in Dataset	Radar in Dataset
0	1957-1966	3	AN/APQ-72, AN/APQ-120, AN/APQ-130
1	1970-1979	5	AN/APG-63, AN/AWG-9, AN/APG-66, AN/APG-65, AN/ALQ-165
2	1980-1989	7	AN/APG-78, AN/APG-68, AN/APQ-181, AN/ALQ-161A, AN/APG-71, AN/APQ-173, AN/APG-73
3	1990-1999	5	AN/APY-2, AN/APG-77, AN/APG-63 (V1), AN/APG-63 (V2), AN/APG-68 (V9)
4	2000-2009	6	AN/APG-79, ZPY-2, AN/APG-81, AN/APQ-164-Mod, AN/APQ-181 RMP, AN/APG-82

Table 3. Generation Variable: Development

AESA Variable	Number in Dataset	Radar in Dataset
AESA (1)	9	AN/APG-63 (V2), AN/APG-77, AN/APG-79, AN/APG-81, AN/APG-82, AN/APQ-173, AN/APQ-181, AN/APQ-181 RMP, ZPY-2
Non-AESA (0)	17	AN/APG-68 (V9), AN/APG-78, AN/ALQ-161A, AN/ALQ-165, AN/APG-63, AN/APG-63 (V1), AN/APG-71, AN/APQ-120, AN/APQ-130, AN/APQ-164-Mod, AN/APQ-72, AN/APY-2, AN/AWG-9, AN/APG-65, AN/APG-66, AN/APG-68, AN/APG-73

Table 4. AESA Variable: Development

Modification Variable	Number in Dataset	Radar in Dataset
Modification (1)	6	AN/APG-63 (V1), AN/APG-63 (V2), AN/APG-68 (V9), AN/APQ-164-Mod, AN/APQ-181 RMP, , AN/APG-82 (V1)
Non-Modification (0)	20	AN/APQ-72, AN/APQ-120, AN/APQ-130, AN/APG-63, AN/AWG-9, AN/APG-66, AN/APG-65, AN/ALQ-165, AN/APG-78, AN/APG-68, AN/APQ-181, AN/ALQ-161A, AN/APG-71, AN/APQ-173, AN/APG-73, AN/APY-2, AN/APG-77, AN/APG-79, ZPY-2, AN/APG-81

Table 5. Modification Variable: Development

In the production analysis, the three variables are adapted from the development cost variables. Within the "generations" variable, existing radars were retained in their development generation classification for consistency, while one additional radar, previously lacking development cost information was incorporated due to the availability of production cost data. It is important to highlight that the production data contained only a single modification data point for analysis. The AESA and modification variables for the production analysis adhere to the same criteria applied in the development analysis. Tables 6, 7, and 8 outline the three variables employed in the production analysis.

Generations Variable	Development Year Range	Number in Dataset	Radar in Dataset
0	1957-1966	4	AN/APG-63, AN/AWG-9, AN/APG-66, AN/APG-65
1	1970-1979	5	AN/APG-78, AN/APG-70, AN/ALQ-161A, AN/APG-71, AN/APG-73
2	1980-1989	1	AN/APG-77
3	1990-1999	3	AN/APG-79, AN/APG-81, AN/APG-82 (V1)

Table 6. Generation Variable: Production

AESA Variable	Number in Dataset	Radar in Dataset
AESA (1)	4	AN/APG-77, AN/APG-79, AN/APG-81, AN/APG-82 (V1)
Non-AESA (0)	9	AN/APG-78, AN/ALQ-161A, AN/APG-63, AN/APG-71, AN/AWG-9, AN/APG-65, AN/APG-66, AN/APG-73, AN/APG-70

Table 7. AESA Variable: Production

Modification Variable	Number in Dataset	Radar in Dataset
Modification (1)	1	AN/APG-82 (V1)
Non-Modification (0)	12	AN/APG-63, AN/AWG-9, AN/APG-66, AN/APG-65, AN/APG-78, AN/APG-70, AN/ALQ-161A, AN/APG-71, AN/APG-73, AN/APG-77, AN/APG-79, AN/APG-81

Table 8. Modification Variable: Production

Statistical Analysis

All statistical analyses in this study are performed at a 5% level of significance. Because the distribution of all dependent variables were deemed sufficiently normal via visual inspection, Anderson-Darling and Kolmogorov-Smirnov tests for normality, inferential tests consist of parametric t-tests to explore differences in means between AESA/non-AESA and modification/non-modification programs. AESA/non-AESA comparisons are also analyzed via the Kruskal-Wallace rank-order test for robustness, as the AESA data was determined normally distributed via the Kolmogorov-Smirnov test only, and a noticeable difference between the mean and median of the distribution warranted the non-parametric robustness check. Statistically significant findings in the t-tests/Kruskal-Wallace tests are further analyzed via ordinary least squares (OLS) regression. Differences across generations were evaluated

through analysis of variance (ANOVA) as more than two categories exist. We also investigate the potentiality for cost growth across time via OLS regression. In an effort to replicate previous studies with expanded data while simultaneously examining effects of the different generations, we analyze the following regression models:

$$Dev\ Cost_i = \alpha_0 + \sum_{n=1}^{4} \beta_n Generations_{ni} + \beta_5 Weight_i + \varepsilon_i \quad (1)$$

Where:

Dev Cost$_i$: Total Developmental Cost (CP$K) of program *i*

Generations$_{ni}$: Generation *n* (1-4) of program *i*

Weight$_i$: Total Weight of program *i*

ε_i: Idiosyncratic error term

α_0: Model Intercept

$$Prod\ Cost_i = \alpha_0 + \sum_{n=1}^{3} \beta_n Generations_{ni} + \beta_5 Weight_i + \varepsilon_i \quad (2)$$

Where:

Prod Cost$_i$: $T100 Production Cost (CP$K) of program *i*

Generations$_{ni}$: Generation *n* (1-3) of program *i*

Weight$_i$: Total Weight of program *i*

ε_i: Idiosyncratic error term

α_0: Model Intercept

Table 9 displays the independent variables analyzed in the regressions and the expected impact of each.

Developmental and Production Variables	
Variable Name	**Expected Impact**
Generations Dummy Variables	Unknown
Total Weight	+

Table 9: Independent Variables for Regression Analysis

Following the regression analysis, we perform a series of diagnostic tests to validate regression results and verify that the regressions are not affected by multicollinearity, non-constant variance, non-normality in the residuals, outliers, and overly influential data points. Table 10 summarizes the diagnostic tests performed as well as their intended function. Unless otherwise noted in this paper, all diagnostic tests revealed satisfactory results. Specific results are available upon request.

Test	Description
Anderson-Darling Test	Tests that the residuals follow a normal distribution.
Variance Inflation Factor (VIF)	Diagnostic for the detection of multicollinearity that could impact the accuracy of model standard errors. A VIF score of 5 or lower is typically considered acceptable.
Breusch-Pagan Test	Employed to examine whether constant variance is observed in the regression residuals (homoskedasticity)
Cook's Distance	Used to identify overly influential datapoints in the regression. A Cook's Distance greater than 4/n signifies heightened leverage which can skew the results.
Studentized Residuals	Residuals transformed to standard normal to determine potential outliers. Datapoints exceeding 3 standard deviations warrant additional scrutiny

Table 10: Diagnostic Tests for Regression Analysis

RESULTS

Descriptive Statistics

The descriptive statistics, including mean, median, observation count, standard deviation, and inner-quartile range of both the development and production datasets are presented in Tables 11 and 12 respectively. We present the overall descriptive statistics of the data as well as the conditional descriptive statistics based on radar generation, AESA/non-AESA, and modification/non-modification. In subcategories where N = 1, the data is masked to protect proprietary information.

Variable	N	Mean	Median	Std Dev	IQR
Total CP$K	26	$614,554	$505,488	$480,263	$596,619
Generation (0)	3	$297,833	$196,757	$227,869	$420,770
Generation (1)	5	$757,541	$654,466	$441,048	$751,999
Generation (2)	7	$673,918	$484,406	$476,002	$692,565
Generation (3)	5	$588,562	$526,570	$402,420	$758,175
Generation (4)	6	$606,160	$323,527	$699,359	$932,878
AESA	8	$636,427	$352,325	$637,841	$887,521
Non-AESA	18	$604,832	$535,194	$413,941	$566,389
Modification	6	$299,332	$224,973	$151,598	$282,700
Non-Modification	20	$709,120	$606,611	$506,607	$701,044

Table 11: Development Cost Descriptive Statistics (CP22 $K)

Variable	N	Mean	Median	Std Dev	IQR
T100 CP$K	12	$3,217	$2,838	$1,805	$2,191
Generation (0)	4	$3,588	$2,775	$2,456	$4,390
Generation (1)	4	$2,292	$1,911	$1,367	$2,503
Generation (2)	1	MASKED			
Generation (3)	3	$3,025	$3,043	$383	$766
Generation (4)	0	-	-	-	-
AESA	4	$3,770	$3,221	$1,522	$2,618
Non-AESA	8	$2,940	$2,262	$1,966	$2,559
Modification	1	MASKED			
Non-Modification	11	$3,232	$2,633	$1,893	$2,429

Table 12: Production Cost Descriptive Statistics (CP22 $K)

Generation Analysis

The development generations were categorized as 0 – 4, where generation 0 serves as the baseline and 1, 2, 3, and 4 represent adjustments to this baseline. Each generation represents a 10-year period. Development costs were regressed on each generation and weight as described in equation 1. Notably, the model did not yield any statistically significant results, indicating that neither weight nor any of the generation variables yielded meaningful explanatory evidence. This is further shown with the model's R-squared value of 0.1 and adjusted R-squared value of -0.17, meaning the explanatory penalty for the number of independent variables exceed the additional explanation of variation. While the model itself is unremarkable, this is a positive result for the DoD as we can infer that technological advancements do not strictly increase cost. In fact, if we were able to quantify capability, the results may indicate that capability per dollar spent is actually increasing over time—a welcome result for any acquisition professional. Table 13 summarizes the development regression results.

Variable	Estimate ($K) *(standard error)*	P-Value	VIF	R2/Adj R2
Intercept	286563.17 *(291303.66)*	0.34	-	0.1/-0.17
Total Weight	20.33 *(63.17)*	0.75	1.107	
Generation (1)	286978.51 *(382580.17)*	0.46	1.928	
Generation (2)	432016.76 *(358513.61)*	0.25	2.272	
Generation (3)	261666.3 *(376783.38)*	0.5	2.214	
Generation (4)	389729.62 *(366178.15)*	0.3	2.091	

Table 13: Development Regression Results

Variable	Estimate ($K) *(standard error)*	P-Value	VIF	R2/Adj R2
Intercept	2,396.94 *(1362.57)*	0.0585	-	0.45/0.089
Total Weight	1.95 *(1.67)*	0.2876	1.067	
Generation (1)	-1,435.12 *(1282.34)*	0.8003	1.284	
Generation (2)	2,694.41 *(2032.59)*	0.1206	1.152	
Generation (3)	-299.9 *(1580.89)*	0.1381	1.254	

Table 14: Production Regression Results

The production generations were categorized as 0 – 3, where generation 0 serves as the baseline and 1, 2, and 3 represent adjustments to this baseline. While these generations correspond to the same generations as the development dataset, no production data was available for generation 4. $T100 production costs were regressed on each generation and weight as described in equation 2. Similar to what was observed in the development data, the model did not yield any statistically significant results, indicating that neither weight nor any of the generation variables yielded meaningful explanatory evidence. This is further shown with the model's R-squared value of 0.45 and adjusted R-squared value of 0.089 indicating a much stronger penalty for adding independent variables than additional variation explained. Again, this null result is a positive finding—technological advancements do not appear to yield higher production costs over the years indicating that capability per dollar spent may be increasing over time—the DoD is getting "more bang for the buck". Table 14 summarizes the production regression results.

While the regression analysis shows no meaningful cost growth for the generations against the base case (generation 0), the regressions do not examine whether the generations are statistically different from each other. The regressions give a strong indication that the DoD is not experiencing cost growth in the long run, but they do not definitely reveal whether generations differ from each other. For example, the regressions cannot indicate whether generation 4 differs from generation 2 or 3, only whether generation 4 differs from generation 0. This remains true for all generations other than generation 0. To further explore the potential differences in cost across generations, we conduct ANOVA on the development and production data across the generation subsets. ANOVA analyzes the ratio of the Mean Square between groups to the Mean Square within groups. The resultant ratio, known as the F statistic, quantifies the amount of variation between the different groups relative to the variation within each group. For example, if the ratio is less than one, this indicates that there is more variation within a single group than variation across groups, indicating that the groups are likely statistically equivalent. This ratio is formally tested against the F-distribution to derive a p-value indicating the likelihood that the differences in the groups are an artifact of random chance. A p-value above 0.05 indicates that we lack sufficient evidence to say the groups are different. For both development and production datasets, the F-Statics were well under 1 and p-values far exceeded 0.05 indicating that no generation was statistically different than another. Table 15 summarizes the ANOVA results for both development and production datasets.

	F Statistic	p-value
Development	0.42	0.79
Production	0.33	0.53

Table 15: Development and Production ANOVA Results

	T-test p-value	Kruskal-Wallis p-value
Development	0.55	0.66
Production	0.78	-

Table 16: AESA Parametric and Non-Parametric Tests

AESA vs Non-AESA Analysis

AESA technology marks a significant evolutionary advancement in fire control radar capability, spanning multiple generations. Due to its presence across various generations, traditional generational analysis cannot adequately determine whether AESA technology alone significantly contributes to radar cost growth. To explore AESA as a potential cost driver, we conducted t-tests to assess differences in mean costs between AESA and non-AESA radars for both development and $T100 production costs. The t-tests yielded p-values of 0.55 and 0.78 respectively, resulting in a failure to reject the null hypothesis and indicating no significant difference in means for both development and production. Given the weak justification for the normality of the development AESA data, we also utilized the non-parametric Kruskal-Wallis rank sum test for robustness. This test produced a p-value of 0.66, leading to a similar conclusion of no significant difference in medians of the development costs. The Kruskal-Wallis rank sum test was not implemented for the production data, as the Anderson Darling test for normality provided strong evidence that production data followed a normal distribution. Since the parametric and non-parametric tests did not yield significant results, further analysis using OLS regressions was not deemed necessary. Table 16 summarizes the results.

	T Statistic	p-value	Confidence Interval Analysis
Develop-ment	-2.75	0.006	-
Production	-	-	Within AESA/non-AESA 95% Confidence Intervals

Table 17: Modification Parametric Tests/Confidence Interval Analysis

Modification vs New Radar Analysis

To assess differences in developmental costs between new and modified radar programs, we conducted parametric t-tests. The results indicated that modification programs were less expensive to develop than new radar programs, with a statistically significant t-statistic of -2.75 and a p-value of 0.006. However, data limitations precluded a traditional t-test analysis for production costs, as modification production data was limited to a single data point, as shown in Table 8. To explore production costs between modified and new radars, we constructed a set of confidence intervals based on the production costs of both AESA and non-AESA new radars and evaluated whether the cost of the one modification radar fell within these 95% confidence intervals. Although definitive statistical conclusions cannot be drawn, the modification program's cost was situated well within both sets of confidence intervals, suggesting that production costs for modified radars may not significantly differ from those of new radars, regardless of whether the radar is AESA or non-AESA. However, more data will be needed to determine this conclusively. Table 17 summarizes these results.

To quantify the cost difference between modified and new radar development programs, we adjusted the original OLS regression model by adding a control variable for modification programs in Equation 3:

$$Dev\ Cost_i = \alpha_0 + \sum_{n=1}^{4} \beta_n Generations_{ni} + \beta_5 Weight_i + \beta_6 Mod_i + \varepsilon_i \quad (3)$$

Where:

Dev Cost$_i$: Total Developmental Cost (CP$K) of program *i*

Generations$_{ni}$: Generation *n* (1-4) of program *i*

Weight$_i$: Total Weight of program *i*

Mod$_i$: Dummy variable = 1 if program *i* is a modification program

ε_i: Idiosyncratic error term

α_0: Model Intercept

The coefficient on the modification variable in equation 3 illustrates the expected cost difference between the development of modified and new radars. The results show that, on average, modified radars are nearly $671,000 less expensive to develop than new radar programs. It is also worth noting that Generations 3 and 4 become more statistically significant as their p-values drop below 0.1. However, we still cannot confidently conclude that these radar generations are more costly than previous ones, as their confidence intervals include zero. With additional data, further investigation into this trend would be justified. All post-regression diagnostic tests were conducted and showed no significant issues, indicating that the findings do not suffer from multicollinearity or bias. Table 18 summarizes the regression results.

Variable	Estimate ($K) *(standard error)*	P-Value	VIF
Intercept	309,576.11 *(264052.39)*	0.2582	-
Total Weight	-21.18 *(60.32)*	0.73	1.231
Generation (1)	289,280.15 *(346512.72)*	0.4161	1.928
Generation (2)	468,516.42 *(325147.52)*	0.1689	2.278
Generation (3)	723,599.93 *(402038.9)*	0.0908	3.073
Generation (4)	669,035.39 *(355684.67)*	0.0783	2.405
Modification	-670,976.19 *(308732.96)*	0.0451*	1.812

Table 18: Modification Development Cost Regression Results

CONCLUSION AND TAKEAWAY

This study examined the evolving relationship between technological advancements and the development and production costs of fire control radar systems, with a particular focus on AESA technology. Through analysis of an expanded dataset and the application of various statistical methods, our findings suggest that while radar capabilities have significantly advanced, there has not been a corresponding dramatic increase in development or production costs. Notably, AESA technology—despite its complexity and the expanding use of T/R modules—did not emerge as a statistically significant driver of cost growth.

These findings provide valuable insights for program offices, particularly in the ongoing debate over whether the increasing adoption of AESA radars will inevitably break the cost curve due to their growing reliance on sophisticated T/R modules. Our analysis suggests that this concern may be overstated, as no significant evidence was found to indicate that AESA radars lead to markedly higher costs. In fact, the results hint at a scenario where capability growth is achieved without breaking the cost curve, providing reassurance that AESA systems may continue to offer enhanced capabilities at a sustainable cost.

However, the study does reveal some nuances. While the regression results did not yield statistically significant cost growth across generations, the p-value for the fourth generation dropped below 0.1, providing some credence to concerns about potential cost increases associated with newer technologies. This suggests that while the data does not yet definitively support the notion of dramatic cost growth, there may be subtle indications of rising costs as we progress into the later generations of radar systems. These findings, although

inconclusive, hint at the possibility of cost growth occurring—but perhaps at a more measured and manageable pace than some have feared. Future research should be conducted as more data becomes available to better understand these potential trends.

Furthermore, the cost-saving potential of modified radar programs compared to entirely new systems was clearly supported, with development costs for modified systems found to be approximately $671,000 lower on average. This reinforces the value of leveraging existing technologies to contain costs while maintaining or enhancing system capabilities. Still, our analysis of production costs for modified radars remains incomplete due to data limitations, and further investigation is necessary.

In sum, this research offers promising evidence that advancing radar capabilities, including the use of AESA technology, does not necessarily translate into unsustainable cost increases. However, the slight movement in p-values for the later generations should serve as a reminder that the landscape of radar technology is complex and evolving. Continued data collection and analysis will be essential to deepen our understanding of these trends and to provide program managers and cost analysts with even more refined tools for future radar acquisition programs. While the current evidence suggests that drastic cost growth may not be imminent, vigilance is required to ensure that future technological milestones do not result in unexpected cost escalation.

References

Chien, C.-F., Wang, J., Chang, T.-C., & Wu, W.-C. (2007). Economic analysis of 450mm wafer migration. International Symposium on Semiconductor Manufacturing, 1-4.

Foley, S. (2011). World War II technology that changed warfare - Radar and bombsights. Academic Symposium of Undergraduate Scholarship.

Greer Jr., W. R., & Moses, O. D. (1992). Extending technology: The estimation and control of costs. Journal of Accounting and Public Policy, 43-66.

Grossman, S. A., Eisman, M., Bohman, A., Price, C. C., Roshan, P., Dryden, J., ... & O'Donoughue, N. A. (2019). A review of radar technology trends and historical cost data. RAND Corporation.

Jovanovic, B., & Rousseau, P. L. (2002). Moore's law and learning-by-doing. National Bureau of Economic Research, 1-28.

Kim, S., Myeong, G., Park, J., Watanabe, K., Taniguchi, T., & Cho, S. (2020). Monolayer hexagonal boron nitride tunnel barrier contact for low-power black phosphorus heterojunction tunnel field-effect transistors. NANO Letters, 3963-3969.

Kirkpatrick, D. L. (2004). Trends in the costs of weapon systems and the consequences. Defence and Peace Economics, 259-273.

Koh, H., & Magee, C. L. (2006). A functional approach for studying technological progress: Application to information technology. Technological Forecasting and Social Change, 1061-1083.

Moore, G. E. (1975). Progress in digital integrated electronics. Technical Digest, 11-13.

Nagy, B., Farmer, J. D., Bui, Q. M., & Trancik, J. E. (2013). Statistical basis for predicting technological progress. PLOS ONE.

Renaud, B. (2013). Technology and the defense industry: Real threats, bad habits, or new (market) opportunities? Journal of Innovation Economics & Management, 59-78.

Stern, D. E., Dryden, J., & Balakrishnan, A. (2008). A cost, technical, and industrial-base review of select airborne radars. RAND Corporation.

Strong, R. (2005). Radar: The evolution since World War II. IEE, A&E Systems Magazine, 19-24.

U.S. Government Accountability Office. (2021). DOD acquisition reform: Increased focus on knowledge needed to achieve intended performance and innovation outcomes (GAO-21-511T, p. 6). Retrieved from https://www.gao.gov/products/gao-21-511t

Xiao, B., Wang, R., Xiang, J., Yang, Y., & Deng, Y. (2021). The development review of airborne fire control radar technology. China Automation Congress.

Younus, B., & Manarvi, I. A. (2011). Defect trend analysis of airborne fire control radar using maintenance history. Aerospace Conference, 1-15.

---

*Capt **Dustin Brewer** is currently assigned as a cost analyst at the Life Cycle Management Center/F-15 at Wright-Patterson Air Force Base (AFB), OH. Capt Brewer is a recent graduate of the Air Force Institute of Technology (AFIT) where he earned an MS in Cost Analysis.*

*Major **Michael J. Brown** is an Assistant Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BS in Economics from the United States Air Force Academy, a MS in Cost Analysis from AFIT, and a MA and PhD in Economics from the University of Tennessee. Major Brown's research interests include recruitment and retention, econometric analysis, and governmental acquisitions.*

*Dr. **Jonathan D. Ritschel** is a Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, MS in Cost Analysis from AFIT, and PhD in Economics from George Mason University. His research interests include public choice, defense cost economics, cost analysis and economic institutional analysis.*

*Dr. **Robert D. Fass** is an Assistant Professor in the Department of Systems Engineering and Management at AFIT. He holds a BA in Economics, an MBA from the University of New Mexico, and a PhD in Business Administration and Management from New Mexico State University. Dr. Fass's research interests include cost analysis, decision analysis, risk analysis, behavioral economics, organizational behavior, organizational change, and government acquisition policy.*

Navigating Defense Software Costs through Multinominal Logistic Regression

Stephen D. Chatterton
Edward D. White
Jonathan D. Ritschel
Brandon M. Lucas
Robert D. Fass
Shawn M. Valentine

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

Using historical data from the Naval Air Systems Command and the Air Force Life Cycle Management Center, this study develops two models for categorizing non-recurring software costs in defense projects. Both models employ Multinomial Logistic Regression (MLR) and provide a probabilistic framework to classify costs into low, medium, or high categories. Results indicate similar predictive performance, with significant cost drivers including code size, experience levels, and hours. One key finding, yet unexpected, is the role of software sizing metrics. The results suggest that ESLOC and SLOC offer complementary insights to estimate software costs. This novel result clashes with conventional wisdom in extant defense cost models and has implications for possible improved decision-making in defense project planning.

Background

Software cost estimation is a critical component of project planning, especially for large-scale defense projects. Traditional cost estimation models, such as Cost Estimating Relationships (CERs), often rely on continuous variables like Source Lines of Code (SLOC) or Equivalent Source Lines of Code (ESLOC) to predict costs as single-point estimates, typically accompanied by a margin of error. However, as Rosenberg (1997) notes, these models face limitations when applied across diverse project environments with varying complexities and codebases. Boehm (1981) highlighted the importance of structured methods for software cost estimation, particularly early in program development when detailed data are limited, and broader classifications are useful for preliminary planning. The high stakes of cost estimation errors are evident in numerous defense software projects, where underestimations of software complexity have led to significant cost overruns, schedule delays, and even project cancellations (Charette, 2006). These challenges highlight the value of approaches that segment projects into cost levels, offering a structured way to address uncertainty before precise estimates are available (Jørgensen & Shepperd, 2007).

Recent trends in software cost estimation include probabilistic and categorical methods, offering flexibility in handling uncertainty (Mittas & Angelis, 2010; MacDonell & Shepperd, 2013). Advances in digital transformation and data-driven methodologies have introduced adaptive models that integrate real-time data, dynamically adjusting to new inputs and enhancing estimation accuracy. For example, a digital cost estimation system leveraging meta-learning algorithms and real-time market data significantly improved the estimation process in complex environments (Nguyen et al., 2023).

Categorizing software projects into distinct cost levels may provide valuable insights for decision-makers. Ricci et al. (2013) demonstrated the utility of categorical models in managing estimation uncertainty, a benefit particularly relevant in defense

projects where effective resource allocation and budgeting are paramount (Bayaga, 2010). By segmenting projects based on estimated cost categories, organizations can better anticipate financial requirements and plan for contingencies, thereby reducing the likelihood of cost overruns. This structured approach aids in strategic planning, allowing resources to be allocated according to projected cost levels, which in turn enhances the overall effectiveness of decision-making processes.

Hybrid machine learning approaches have been explored in the literature to improve software cost estimation accuracy. As highlighted by Moser et al. (2008) leveraging metrics in predictive modeling can enhance estimation accuracy, a trend further supported by Derya et al. (2024) who demonstrated the use of optimization techniques like particle swarm optimization and genetic algorithms to select optimal feature sets. However, despite their predictive power, these models often face challenges in high-stakes decision-making environments, such as defense projects, due to their limited interpretability. Machine learning models like artificial neural networks, while sophisticated, often require extensive tuning and tend to have a "black box" nature, making them less transparent compared to regression-based models (de Barcelos Tronto et al., 2008; Papatheocharous & Andreou, 2012).

In defense projects, where transparency and explainability are crucial, more interpretable methods are often preferred. Because of this preference, this study employs Multinomial Logistic Regression (MLR) to categorize software costs into discrete cost levels. Unlike traditional regression models, which provide a single point estimate of costs, MLR predicts the probability of a project falling into each cost category, offering a nuanced understanding of cost outcomes (Mittas, 2011). Balancing predictive accuracy with interpretability, MLR is a practical choice in contexts requiring clear, actionable insights (Bayaga, 2010; Mittas & Angelis, 2010).

Variability in software size measurement complicates accurate cost estimation. Tokdemir and Cagiltay (2019) have highlighted the relationship between SLOC and logical database metrics, underscoring the challenges of early estimation. Despite efforts to standardize practices through guidelines from the Software Engineering Institute and IEEE, inconsistencies in SLOC measurement persist (Iqbal et al., 2017). These discrepancies pose challenges when comparing SLOC metrics across organizations, potentially leading to inaccuracies in cost estimation (Nguyen et al., 2007). Researchers advocate using ESLOC to account for factors like code complexity, language differences, and code reuse, providing a more standardized metric for estimation (Software Technology Support Center, 2008). For instance, Nguyen et al. (2007) highlight the role of ESLOC in addressing inconsistencies inherent in SLOC measurement.

Given the challenge of accurately classifying software costs, this study employs MLR to categorize costs into discrete "buckets." Two distinct methods are used to generate the dependent variable: a tertile-based approach and a data-driven segmentation approach utilizing decision tree logic, each tailored to align with the underlying data patterns. This dual segmentation framework provides a robust foundation for probabilistic cost classification, incorporating explanatory variables that capture key cost drivers. We next discuss the data and methodology adopted in our analysis.

Database and Methods

This study's methodology involves multiple stages, including data preparation, feature selection, model development, and validation. The dataset for this analysis initially included 395 records, each representing a unique software development project, obtained from two primary sources: the Naval Air Systems Command (NAVAIR) Software Resources Data Report (SRDR) database and the Air Force Life Cycle Management Center (AFLCMC) repository. These databases provide information on software development projects, including work breakdown structure elements, contract details, application types, different forms of SLOC (including ESLOC), hours, experience levels, staffing numbers, and software costs. During data cleaning, 43 records were removed due to missing software cost values or the absence of SLOC/ESLOC data, which resulted in a final dataset of 352 completed software contracts representing 43 different DoD programs. Table 1 presents the Mission Design Series (MDS) corresponding to these DoD programs.

AGM-88E	AH-1Z & UH-1Y	AH-64E	AIM-9X	AMDR
AMF JTRS	C-130 AMP	C-130J	CH-53K	CNS/ATM
Cobra Judy	E-2D	E-8C	EA-18G	EPS
F/A-18E/F	F-16 Blk40/50	F-22A	F-35A/B/C	FAB-T
FBCB2	G/ATOR	GBU-53/B	GPS - OCX	GPS-IIIA
IAMD	JAGM	JATAS	JLENS	JMPS
JTRS	JTRS NED - MUOS	MH-60R	MQ-4C	NVST GPS - MUE
NMT	P-8A	RQ-8A	SM-6	TSAT
UH-1N	UH-60M	WIN-T		

Table 1: Mission Design Series of Programs Analyzed

Missing values for specific types of SLOC (e.g., New Code, Modified Code, Autocode) were imputed with zeros, indicating the absence of such lines of code. Records flagged by NAVAIR as having unreliable SLOC or hours data were retained in the dataset, but the flagged variables were left blank to ensure potentially inaccurate figures were not considered in the analysis. The entire dataset was normalized to constant price (CP) 2024 dollars using the Producer Price Index (PPI) 3364 from the Bureau of Labor Statistics, providing consistency across different fiscal years and enabling meaningful comparisons.

To address the increasing variance observed with higher software costs, consistent with the multiplicative nature of models like COCOMO and the analyses by Rosa et al. (2014), a natural logarithmic transformation was applied. Figure 1 illustrates this flaring pattern. To minimize this heterogeneity effect, the variables ESLOC, SLOC, total programming hours, and the dependent variable (software costs) were all transformed to their natural logarithms (i.e., Ln_ESLOC, Ln_SLOC, Ln_Hours, and Ln_Software Costs). These transformations help stabilize variance across the dataset, ensuring more reliable statistical analysis and homogeneity. Figure 2 demonstrates the effectiveness of these transformations in resolving the increasing variance issue.

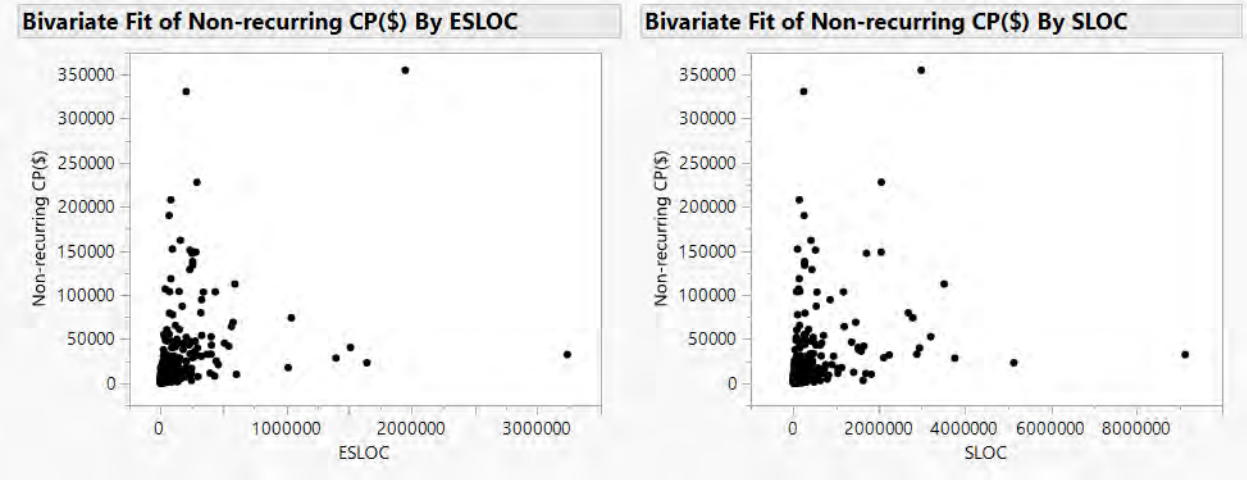


Figure 1. Bivariate Graph of Software Cost versus ESLOC and SLOC

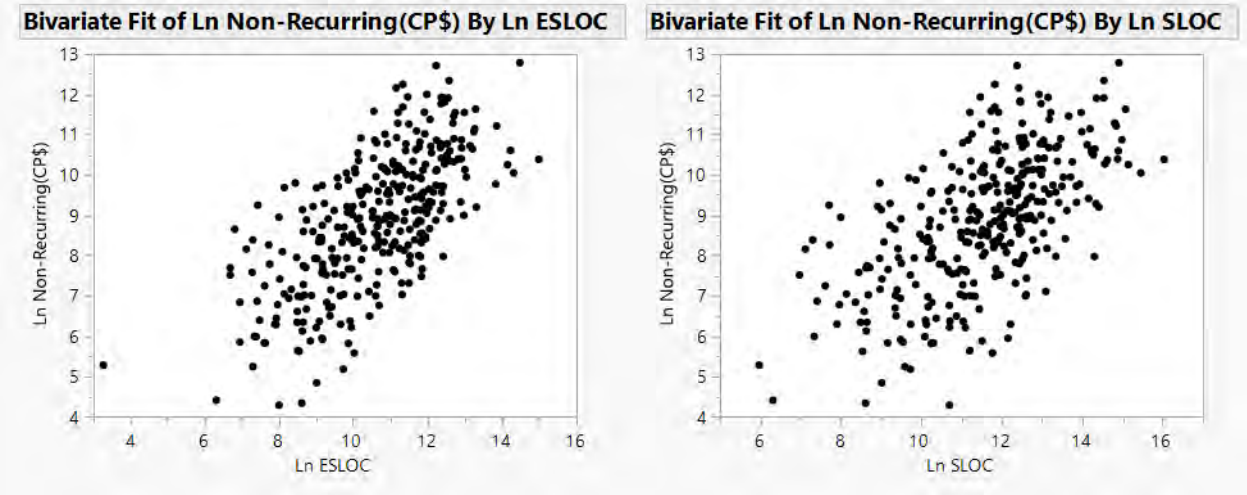


Figure 2. Bivariate Graph of Natural Log of Software Cost versus Natural Log of ESLOC and Natural Log of SLOC

The descriptive and inferential analysis documented in this article was conducted using JMP® Pro 15. A 5% level of significance (Type I error rate) was applied for all statistical tests. The dependent variable, software costs, was categorized into three discrete buckets: low, medium, and high. Two methods were employed to create these categories, tailored to the characteristics of each model. For Model 1, a tertiles method was used, dividing the software cost data into three equal groups, each containing approximately one-third of the contracts. This method was selected for its simplicity and even distribution of cost categories, facilitating straightforward comparisons across projects.

For Model 2, a decision tree-based method was utilized to identify natural breakpoints in the data, resulting in an uneven distribution of cost categories that better aligned with observed patterns. This approach was implemented using JMP's methodology, which selects breakpoints by minimizing within-node variance and maximizing separation between groups. The variables used to determine splits in the decision tree method were identified in a prior OLS regression analysis by Chatterton, et al. (2026), where Ln_SLOC, Ln_ESLOC, Ln_Hours, Electronic/Automated Software, and Fixed Price were found to be significant predictors of software costs.

Following the determination of cost buckets, feature selection was conducted to identify the most influential predictors for the MLR model. This study employed a mixed stepwise regression approach, combining forward selection and backward elimination techniques. This method iteratively evaluates predictor variables, including those that significantly improve model fit and excluding those with minimal impact. The significance level for including or excluding a variable was set at a p-value of 0.05, with a Bonferroni correction applied to account for multiple comparisons.

The primary predictors considered during feature selection included ESLOC, SLOC, contract type, commodity type, mission design series, hours (as a covariate), and experience levels of programmers. The selected predictors were then used to develop the MLR model, which categorizes software costs into three risk buckets (low, medium, high). The model coefficients were estimated using maximum likelihood estimation, a method that provides efficient and unbiased parameter estimates by maximizing the likelihood of observing the given data.

Once the MLR model was built its performance was assessed using classification diagnostics. Receiver Operating Characteristic (ROC) curves and the corresponding Area Under the Curve (AUC) values were used to evaluate the model's ability to distinguish between the three cost categories (low, medium, high). The ROC curve helps visualize the model's classification performance by plotting the trade-off between true positives and false positives across different thresholds. The AUC provides a single metric to summarize this performance, where values closer to 1 indicate stronger model

discrimination. A higher AUC value reflects better classification performance, indicating the model's effectiveness in predicting the correct cost bucket for each software project. For instance, an AUC of 0.90 suggests that the model has a 90% chance of correctly ranking a randomly chosen high-cost project higher than a low-cost project. To enhance the robustness of the AUC estimates, a bootstrapping procedure with 2,500 iterations was employed. This technique resamples the data multiple times to estimate the variability in the AUC, providing confidence intervals that offer a more reliable measure of the model's predictive performance (Efron & Tibshirani, 1993).

Following the model evaluation, an exploratory post -hoc Pareto analysis was conducted to rank predictor variables based on their relative importance using their associated chi-squared test statistics. This analysis provided a clear indication of the variables that contributed most significantly to the model's predictive power, focusing the interpretation on the 'vital few' predictors (Juran, 1988).

The analysis provides log-odds equations that allow users to estimate the likelihood of a software project falling into one of the three cost categories (Low, Medium, High). These equations use the coefficients derived from the MLR model, reflecting the influence of predictor variables on the log-odds of each category. To convert the estimated log-odds into probabilities, practitioners can apply the softmax function (shown in Equation 1), which calculates probabilities based on the specific characteristics of the contract being analyzed.

$$P(Y = k) = \frac{e^{log-odds_k}}{\sum_{j=1}^{K} e^{log-odds_j}} \qquad (1)$$

Here, $P(Y = k)$ is the probability of the project falling into category k, and log-odds_k represents the log-odds equation for category k. The softmax function exponentiates the log-odds and normalizes the probabilities, ensuring that they sum to one across all categories. This allows users to interpret the model output and determine the most likely cost category for any given project. Additionally, we provide 95% confidence intervals for the median cost for each category. Next, we present the results of the analysis, detailing the performance of the MLR model and the insights gained from the evaluation of predictor variables.

Results

We break this section into four parts. First, we present descriptive statistics for the dependent and independent variables, including odds ratios and confidence intervals, to display univariate relationships between cost drivers and categories. Following this, we display Model 1 and Model 2. Finally, we compare both models, evaluating their performance across cost categories and identifying practical implications for defense software cost estimation.

To facilitate cost prediction across discrete categories, software costs were categorized differently for each model. Model 1 uses a tertile approach to divide costs into low, medium, and high categories, ensuring an even distribution across these categories. Model 2 applies a decision tree to identify natural breakpoints, resulting in an uneven distribution that better aligns with observed data patterns. Throughout this analysis, a natural logarithmic transformation (Ln) was applied to all continuous variables, including both dependent and independent variables, to minimize heterogeneous variance. Figures 3 and 4 display histograms illustrating the cost distributions within each model's defined categories with respect to the dependent variable.

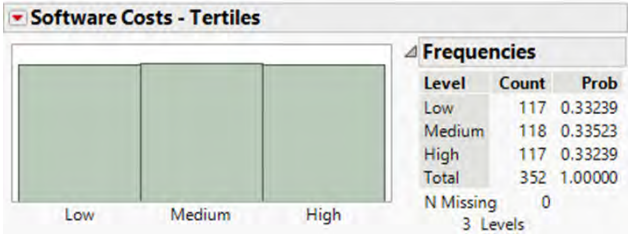


Figure 3. Distribution of Software Costs Using Tertile-Based Method in Model 1

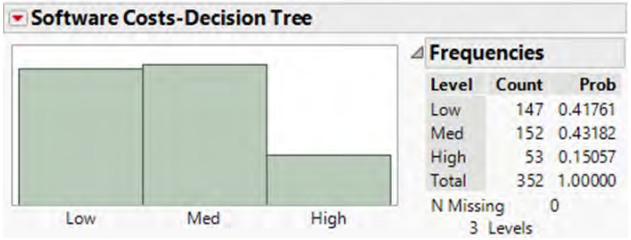


Figure 4. Distribution of Software Costs Using Decision Tree-Based Method in Model 2

	Tertile Method	Decision Tree Method
Low Cost Range	$3K - $3,570K (117 contracts)	Total Hours < 21,590 (147 contracts), Mean = 1,901
Medium Cost Range	$3,635K - $15,480K (118 contracts)	Total Hours ≥ 21,591 and ESLOC < 204,843 (152 contracts), Mean = 11,849
High Cost Range	$15,856K - $354,675K (117 contracts)	Total Hours ≥ 21,591 and ESLOC ≥ 204,844 (53 contracts), Mean = 43,478
Splitting Criteria	Cost Percentiles (1/3, 1/3, 1/3)	Decision tree based on Ln_Hrs and Ln_ESLOC breakpoints
* Ranges shown are in original, non-transformed cost values for clarity.		

Table 2. Descriptive Statistics for Dependent Variables in Study

Table 2 expounds on Figures 3 and 4 with respect to showing the quantitative ranges of the three cost categories. As a reminder, even though Table 2 displays the actual, non-transformed cost ranges for clarity, all statistical analyses and model estimations were conducted using the transformed (Ln) values.

Table 3 displays the independent variables in this analysis. Independent variables (IVs) were tailored to meet analysis needs, with some naturally dichotomous (e.g., Electronic/Automated Software, Modification) and others transformed into dichotomous variables using decision tree methods or thresholds. For instance, Ln_ESLOC and Total Ln_Hours were categorized into low, medium, and high groups, then recoded as dichotomous variables for the MLR models (e.g., Ln_ESLOC Low coded as 1 for contracts in the lower range, 0 otherwise). The categorized ranges shown in the tables provide context and statistical relevance for software development projects.

Model 1

The first model was developed using software costs as the dependent variable, categorized into low, medium, and high-cost tertiles, each containing

Variable	Type	Categories or Range	Number of Con-
Ln_SLOC	Continuous (Decision Tree)	Low: 392–69,564, Medium: 70,262–392,385, High: 396,329–9,156,733	Low: 143, Medium: 141, High: 68
Ln_ESLOC	Continuous (Decision Tree)	Low: 26–20,952, Medium: 21,162–202,805, High: 204,843–3,236,490	Low: 111, Medium: 161, High: 56
Total Ln_Hours	Continuous (Decision Tree)	Low: 19–30,638, Medium: 30,946–137,310, High: 138,690–2,495,501	Low: 173, Medium: 97, High: 61
Electronic/ Automated Software	Dichotomous	1 = Electronic/Automated, 0 = Not Electronic/Automated	1: 136, 0: 216
Very High Experience >30%	Dichotomous	1 = More than 30% high experience, 0 = Less than 30%	1: 65, 0: 287
Entry-level >10%	Dichotomous	1 = More than 10% entry-level, 0 = Less than 10%	1: 79, 0: 273
High Requirements Analysis Hours	Categorical (Decision Tree)	None: 0 hours, Low: 1–1,737 hours, High: 1,958–220,721 hours	None: 56, Low: 136, High: 139
High Other Development Hours	Categorical (Decision Tree)	Low: 0–288 hours, Medium: 306–7,723 hours, High: 8,040–350,969 hours	Low: 110, Medium: 111, High: 139
Modification	Dichotomous	1 = Modification, 0 = Not a modification	1: 51, 0: 301
* For continuous variables, ranges shown are in original, non-transformed cost values for clarity.			

Table 3. Descriptive Statistics for Independent Variables in Study

Variable	Test Statistic (Chi-Square)	p-value
Ln_ESLOC Low	34.45	< 0.0001
Ln_ESLOC High	10.63	0.0049
Total Ln_Hours Low	34.23	< 0.0001
Total Ln_Hours High	16.75	0.0002
Very High Experience >30%	15.1	0.0005
Entry-level >10%	11.09	0.0039
Electronic/Automated Software	18.06	0.0001

Table 4. Model 1: Mixed Stepwise Regression Results

approximately one-third of the contracts. To identify significant predictors, a mixed stepwise regression approach was applied with an initial significance threshold of 0.05. After a Bonferroni correction for multiple comparisons adjusted the threshold to 0.007, only predictors with associated p-values below this level were retained in the final model. Because this model categorizes costs into three groups, the test statistics are compared to a chi-square distribution with 2 degrees of freedom. Table 4 displays the final significant predictors in Model 1, including their chi-square test statistics and associated p-values.

The results highlight key predictors with strong associations to software cost categorization based on the chi-square tests. Variables like Ln_ESLOC Low, Total Ln_Hours Low, Ln_ESLOC High, and Total Ln_Hours High show significant impact, reflecting the importance of code size and labor hours in

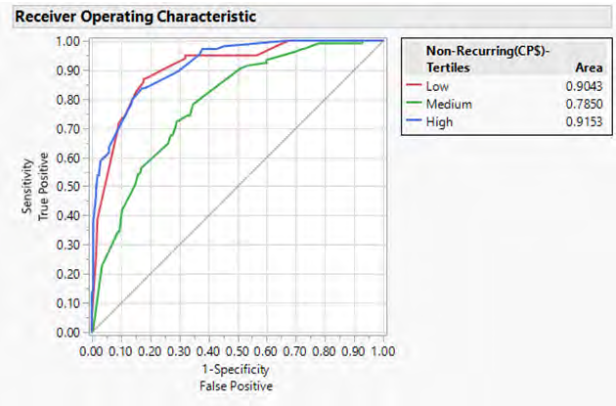


Figure 5. Model 1: Receiver Operating Characteristic Graph and Area Under the Curve Values

differentiating cost categories. Additional significant factors include Very High Experience >30%, Entry-level >10%, and Electronic/Automated Software, indicating that workforce composition and project type also contribute meaningfully. Further interpretation, including the direction of these effects, is addressed in the subsequent model equations. Figure 5 presents the ROC curves, demonstrating strong classification performance, particularly for low and high-cost categories.

The AUC values indicate that the model effectively distinguishes between low and high-cost contracts, with slightly reduced performance for medium-cost contracts. Overall, the AUC scores reflect predictive accuracy, particularly for cost extremes. To assess the reliability of these findings, 2,500 bootstrap iterations provided confidence intervals for the AUC values, with results displayed in Table 5.

Cost Category	Original AUC	95% Confidence Interval
Low-cost contracts	0.904	[0.858, 0.944]
Medium-cost contracts	0.785	[0.711, 0.821]
High-cost contracts	0.915	[0.874, 0.939]

Table 5. Model 1: Bootstrapped Results with Confidence Intervals

The bootstrapped intervals confirm that the AUC estimates are stable and reflect robust classification performance across cost categories. The model is especially reliable for low and high-cost contracts, with slightly wider confidence intervals for medium-cost contracts due to their lower AUC. The cost ranges for each category, shown in Table 6, clarify the thresholds defining low, medium, and high-cost contracts, derived directly from the actual cost distributions.

Cost Category	Minimum Cost ($K)	Maximum Cost ($K)	Median Cost ($K)
Low-Cost	4	3,570	1,142
Medium-Cost	3,636	15,479	7,443
High-Cost	15,857	354,675	38,151

Table 6. Model 1: Predicted Cost Ranges by Category

These ranges clarify the thresholds for each cost category, providing decision-makers with clear distinctions between low, medium, and high-cost projects. Figure 6 presents a Pareto analysis that summarizes the relative importance of each variable, with chi-square values identifying Ln_ESLOC Low and Total Ln_Hours Low as the most significant predictors of cost outcomes.

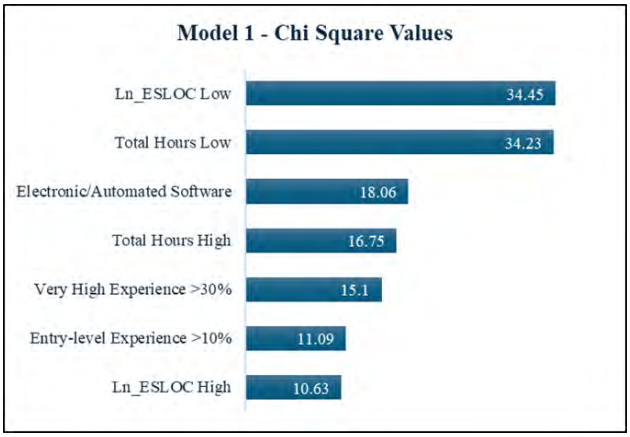


Figure 6. Model 1: Pareto Analysis of Variable Significance

The Pareto analysis shows software size (Ln_ESLOC) and programming hours (Ln_Hours) as the primary drivers of software costs, while programmer experience and electronic/automated software have secondary effects. The following equations represent the estimated regression models derived from Model 1, expressed in log-odds form. These equations enable analysts to estimate the likelihood of a project falling into one of three cost categories: Low, Medium, or High. Equation 2 provides the log-odds for a project being in the Low-cost category relative to the Medium-cost category, while Equation 3 provides the log-odds for the Medium-cost category relative to the High-cost category. These log-odds equations, derived directly from the MLR analysis coefficients, form the foundation for calculating category probabilities using the softmax function.

Each coefficient represents the impact of a variable on the log-odds of being in a specific cost category. For instance, a positive coefficient for Ln ESLOC Low [see Table 3 for low ESLOC range] in Equation 1 indicates that projects with lower ESLOC values are more likely to fall into the Low-cost category relative to Medium. To calculate the probability of a project belonging to each cost category, the log-odds results are exponentiated and normalized. As an example, suppose a project has variable values as shown in Table 7.

Variable	Value
Ln_ESLOC Low	1
Ln_ESLOC High	0
Total Ln_Hours Low	1
Total Ln_Hours High	0
Very High Experience >30%	0
Entry-level >10%	1
Electronic/Automated Software	1

Table 7. Example Calculation Variable Values

$$\log\left(\frac{P(Low)}{P(Medium)}\right) = -2.92 + 1.97 * Ln\ ESLOC\ Low + 2.09 * Total\ Ln\ Hours\ Low + 1.22 * Very\ High\ Experience\ > 30\% \quad (2)$$

$$\log\left(\frac{P(Medium)}{P(High)}\right) = -0.11 - 2.11 * Total\ Ln\ Hours\ High - 1.79 * Ln\ ESLOC\ High + 1.45 * Electronic\ Automated\ Software + 1.22 * Total\ Ln\ Hours\ Low + 1.69 * Very\ High\ Experience\ > 30\% - 1.21 * Entry\ level\ > 10\% \quad (3)$$

Step 1: Calculate the Log-Odds:

For Low vs. Medium (Equation 2):

$$log\left(\frac{P(Low)}{P(Medium)}\right) = -2.92 + (1.97 * 1) + (2.09 * 1) = 1.14$$

For Medium vs. High (Equation 3):

$$log\left(\frac{P(Medium)}{P(High)}\right) = -0.11 + (1.45 * 1) + (1.22 * 1) - (1.21 * 1) = 1.35$$

Step 2: Exponentiate the Log-Odds:

For Low: $e^{1.14} = 3.13$

For Medium: $e^{1.35} = 3.86$

For High (reference category): $e^{0} = 1$

Step 3: Apply the Softmax Function:

To find the probabilities for each category, normalize these values along with the odds for the reference (High) category, which is 1 by default:

$$P(Low) = \frac{3.13}{3.13 + 3.86 + 1} = \frac{3.13}{7.99} \approx 0.39$$

$$P(Medium) = \frac{3.86}{3.13 + 3.86 + 1} = \frac{3.86}{7.99} \approx 0.48$$

$$P(High) = \frac{1}{3.13 + 3.86 + 1} = \frac{1}{7.99} \approx 0.13$$

In this example, the probabilities for each cost category are approximately 39% for Low, 48% for Medium, and 13% for High. This calculation illustrates how the model distributes probabilities across categories based on the input values. The 95% confidence intervals for the median non-recurring cost (in CP$K) are ($1,217, $1,586), ($7,551, $8,778), and ($48,377, $70,078), respectively for the low, medium, and high tertile categories. Next, we examine our second model, which utilizes SLOC in lieu of ESLOC for our software cost classification.

Model 2

In Model 2, a decision tree approach was used to identify natural breakpoints in software costs, categorizing contracts into low, medium, and high-cost groups, resulting in 147 low-cost, 152 medium-cost, and 53 high-cost contracts. To generate the decision tree-based dependent variable for this model, the analysis initially included Ln_Hrs, Ln_SLOC, Ln_ESLOC, Electronic/Automated Software, and Fixed Cost as predictors, following their significance in preliminary OLS regression models (Chatterton, et al., 2026). Once the decision tree DV was established, a mixed stepwise regression approach was applied to refine the MLR

model, with an initial significance threshold of 0.05. After a Bonferroni correction adjusted the threshold to 0.0125, only predictors with p-values below this level were retained. The final model included High Requirements Analysis Hours, High Other Development Hours, Ln_SLOC High, and Modification as significant predictors, with test statistics compared to a chi-square distribution with 2 degrees of freedom. Table 8 summarizes these results.

Variable	Test Statistic (Chi-Square)	*p-value*
High Requirements Analysis Hours	102.66	< 0.0001
High Other Development Hours	92.39	< 0.0001
Ln_SLOC High	83.69	< 0.0001
Modification	9.25	0.0098

Table 8. Model 2: Mixed Stepwise Regression Results

The results show the key predictors for Model 2, with High Requirements Analysis Hours, High Other Development Hours, and Ln SLOC High demonstrating the strongest associations with software cost categories based on their chi-square values. These variables highlight the importance of project scope, extensive development activities, and code size in distinguishing costs. Modification is also significant, suggesting cost differences between new and modified projects. Figure 7 illustrates the model's classification performance through ROC curves and AUC values for each category.

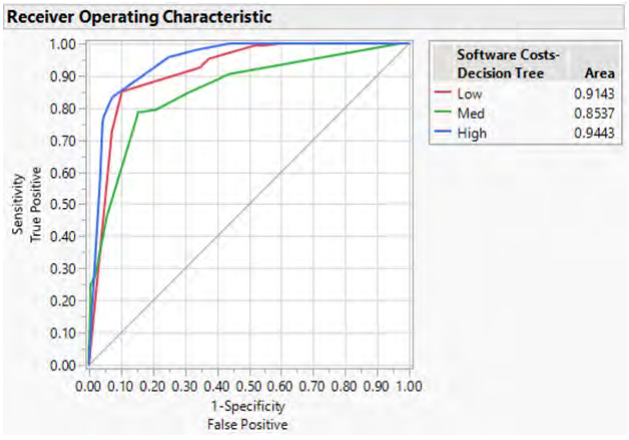


Figure 7. Model 2: Receiver Operating Characteristic Graph and Area Under the Curve Values

Cost Category	Original AUC	95% Confidence Interval
Low-cost contracts	0.914	[0.881, 0.941]
Medium-cost contracts	0.854	[0.809, 0.884]
High-cost contracts	0.944	[0.907, 0.968]

Table 9. Model 2: Bootstrapped Results with Confidence Intervals

The AUC values reveal that Model 2 performs well in differentiating cost categories, especially for low and high-cost contracts, with slightly reduced accuracy for medium-cost projects. This strong classification capability is further supported by confidence intervals for the AUC values, obtained through 2,500 bootstrap iterations, as presented in Table 9.

The bootstrapped confidence intervals support the stability of the AUC estimates for Model 2, with high and low-cost categories showing especially strong performance. The wider intervals for the medium-cost category reflect their slightly lower AUC. Table 10 displays the cost ranges derived from actual distributions, defining the thresholds for low, medium, and high-cost contracts.

Cost Category	Minimum Cost ($K)	Maximum Cost ($K)	Median Cost ($K)
Low-Cost	4	106,742	2,114
Medium-Cost	190	207,727	11,179
High-Cost	7,424	354,675	41,912

Table 10. Model 2: Predicted Cost Ranges by Category

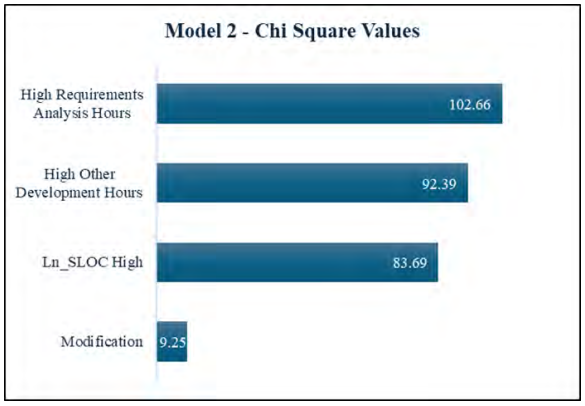


Figure 8. Model 2: Pareto Analysis Effect of Input

These ranges establish clear thresholds for each cost category, aiding decision-makers in distinguishing between low, medium, and high-cost projects. Figure 8 presents a Pareto analysis highlighting High Requirements Analysis Hours and High Other Development Hours as the most influential predictors, followed by Ln_SLOC High and Modification.

The following equations represent the estimated regression models in log-odds form. Equation 4 provides the log-odds for a project falling into the Low-cost category relative to the Medium-cost category, while Equation 5 provides the log-odds for the Medium-cost category relative to the High-cost category. These equations, developed from the MLR coefficients, support the calculation of category probabilities and highlight the relationships between predictor variables and cost classifications for Model 2.

$$log\left(\frac{P(Low)}{P(Medium)}\right) = 1.26 - 1.63 * High\ Requirements\ Analysis\ Hours - 1.86 * High\ Other\ Development\ Hours \quad (4)$$

$$log\left(\frac{P(Medium)}{P(High)}\right) = -1.65 - 2.05 * Ln\ SLOC\ High + 1.04 * Modification \quad (5)$$

These log-odds equations allow analysts to calculate the probability of each cost category by using the coefficients to interpret the influence of each predictor. For example, in Equation 3, the negative coefficients for High Requirements Analysis Hours and High Other Development Hours suggest that projects with greater requirements analysis and development hours are less likely to fall into the Low-cost category compared to Medium. In Equation 4, the negative coefficient for Ln SLOC High implies that larger software sizes increase the likelihood of being in the High-cost category relative to Medium, while the positive coefficient for Modification suggests that modification projects are more likely to be in the Medium-cost category than High. As outlined in the methodology section, these log-odds are converted to probabilities using the softmax function. The 95% confidence intervals for the median non-recurring cost (in CP$K) are ($3,402, $6,775), ($18,837, $30,635), and ($47,006, $86,923), respectively for the low, medium, and high decision tree cost categories. We now compare the performance and predictive insights of Model 1 and Model 2 to evaluate their effectiveness in classifying software cost categories.

Model Comparison

The primary distinction between Model 1 and Model 2 is the role of software size metrics. In Model 1, ESLOC was significant, whereas in Model 2, SLOC emerged as the significant predictor. This difference highlights the varying contexts in which each metric is most impactful for cost categorization. To distinguish them, we refer to Model 1 as the ESLOC -based model and Model 2 as the SLOC-based model, reflecting each model's primary software sizing metric. This naming highlights the impact of traditional versus complexity-adjusted code size on software cost categorization, emphasizing each metric's unique advantages for defense software projects.

Both models perform well across cost categories, achieving high AUC values in each range. While the SLOC-based model attains slightly higher AUC values overall, both models provide reliable cost categorization capabilities. The ESLOC-based model's primary drivers, lower software size (Ln_ESLOC Low) and reduced programming hours (Total Ln_Hours Low), make it well-suited to projects with streamlined, efficiency-focused scopes. Additional factors – such as Electronic/Automated Software and Very High Experience >30% --

enhance the model's specificity for projects where efficiency, team experience, and standardized processes play critical roles in moderating costs. This broader array of predictors enables the ESLOC-based model to effectively estimate costs in projects with structured workflows and targeted resource use.

In contrast, the SLOC-based model emphasizes factors associated with overall project scope and resource intensity. The inclusion of Ln SLOC High as a key predictor aligns this model with projects that involve larger codebases, where extensive coding needs can drive up costs. Additionally, variables like High Requirements Analysis Hours and High Other Development Hours capture the impact of both structured planning and more variable development activities, making Model 2 well-suited to projects that demand diverse and sometimes unstructured resource allocation. While Requirements Analysis hours signify the importance of detailed initial scoping, Other Development hours encompass tasks outside standard categories, adding flexibility for estimating costs in projects with complex development phases. Next, we turn to the broader implications of our findings and explore the practical applications, limitations, and future directions in the discussion and conclusion.

Discussion and Conclusion

This study developed two distinct models to categorize software project costs: the ESLOC-based model and the SLOC-based model. Both performed fairly well in distinguishing between low, medium, and high-cost contracts. The ESLOC-based model, driven by predictors such as Ln_ESLOC, Total Ln_Hours, and experience-related variables, may be particularly suited for projects emphasizing efficiency, standardized workflows, and experienced teams. The consistent association of Very High Experience (>30%) with lower costs suggests that seasoned teams enhance efficiency and mitigate risks, supported by findings of improved productivity, quality, and risk management in experienced teams (Krishnan, 1998; Melo et al., 2013; Boehm et al., 2014). Conversely, a higher proportion of Entry-level Experience (>10%) was linked to increased costs, likely due to additional training needs and lower execution efficiency, as noted by Boehm (1981) and Jørgensen and Sjøberg (2001). Electronic/Automated Software (E/AS) further reduced the likelihood of high-cost outcomes, aligning with potential cost savings from streamlined, automated processes.

For the SLOC-based model, significant predictors varied across comparisons. High Requirements Analysis and Other Development Hours pushed projects toward medium costs, reflecting the added scope of requirements analysis and varied tasks. This aligns with Jørgensen and Moløkken-Østvold (2004), who noted initial cost increases from thorough analysis, but also potential long-term savings from reduced rework – benefits not captured in our findings. Higher values of Ln SLOC pushed projects toward high-cost classifications, consistent with increased effort in larger codebases. The Modification variable increased the likelihood of medium-cost classification, suggesting updates may require less effort compared to new development.

While the SLOC-based model demonstrated a slight predictive advantage, both models offer valuable insights for categorical cost estimation. The ESLOC-based model's emphasis on efficiency, automation, and team experience makes it potentially more effective for projects with structured workflows and predictable processes. In contrast, the SLOC-based model captures the broader complexities of projects involving diverse activities, extensive requirements analysis, and modifications, making it better suited for larger, less predictable endeavors. Leveraging the strengths of each model based on project characteristics can enhance cost estimation accuracy, and employing both models together provides complementary perspectives, aiding in better resource allocation and strategic planning in defense software projects.

This categorical approach to software cost estimation offers practical advantages, particularly during the early stages of program development when detailed project data may be limited. The model's probability estimates for each cost category (low, medium, high) assist cost analysts in evaluating project risk and refining resource plans. Analysts can use these probabilities to assess the likelihood of a project falling into a specific cost category, enabling informed risk assessments and contingency planning. Additionally, the model's probabilistic outputs can feed into broader cost risk analysis frameworks, where likelihood percentages are used to fit statistical distributions and run

simulations. This process yields risk-adjusted estimates, providing an improved view of potential costs and complementing traditional point estimates by addressing uncertainty in software cost predictions.

This study's finding that SLOC-based and ESLOC-based models demonstrated comparable predictive performance challenges conventional wisdom in defense cost estimating, which often prioritizes complexity-adjusted metrics like ESLOC. The results here suggest that traditional metrics such as SLOC continue to provide valuable insights and merit further consideration in cost categorization efforts. These findings add nuance to the discussion around software cost estimation and encourage further exploration to determine under what conditions each metric provides the most reliable insights. As such, this study highlights the importance of revisiting established approaches with empirical results to refine and improve cost estimation practices.

Some limitations to the analysis should be acknowledged. The dataset, based on historical NAVAIR and AFLCMC records with 352 observations, may limit generalizability to other contexts, particularly commercial software projects. While predictors like Requirements Analysis Hours, Other Development Hours, and experience levels add depth, other potential cost drivers such as hardware dependencies were not included. Although bootstrapping was used for validation, testing on larger and more diverse datasets would enhance the robustness and external validity of the models.

Future research should consider expanding the dataset to include more recent and varied software projects, potentially incorporating commercial data to improve generalizability. Alternative modeling approaches like Random Forest or Gradient Boosting could offer greater flexibility in capturing non-linear relationships. Investigating additional cost drivers, including project management practices and software quality metrics, could further refine cost estimation accuracy. Validating the models across different contexts would help assess their robustness and applicability, providing deeper insights into their strengths and limitations.

References

Bayaga, A. (2010). Multinomial Logistic Regression: Usage and Application in Risk Analysis. *Applied Quantitative Methods*, *5*(2), 288-297.

Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall.

Boehm, B., Turner, R., & Lane, J. (2014). *Perspectives on Software Risk Management*. Springer.

Chatterton, S. D., White, E. D., Ritschel, J. D., Lucas, B. M., Fass, R. D., & Valentine, S. M. (2026). Predicting Software Costs: A Comparison of SLOC and ESLOC Models for DoD Programs. *Defense Acquisition Research Journal*, *33*(1), 30-50. https://doi.org/10.22594/dau.24-935.33.01

Charette, R. N. (2006, August 29). IEEE Spectrum. Retrieved from Why Software Fails: http://www.spectrum.ieee.org/print/1685

de Barcelos Tronto, I. F., da Silva, J. D., & Sant'Anna, N. (2008). An investigation of artificial neural networks based prediction systems in software project management. *Systems and Software*, *81*(3), 356-367. https://doi.org/10.1016/j.jss.2007.05.011

Derya, D., Derya, O. B., & Dokeroglu, T. (2024, July). Multi-Objective Software Project Cost Estimation Using Recent Machine Learning Approaches. *Researcher*, *4*(1), 1-14. doi: 10.55185/researcher.1350323

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall/CRC.

Iqbal, S. Z., Muhammad, I., Sana, A. B., & Khan, N. (2017). Comparative analysis of common software cost estimation modelling techniques. *Mathematical Modelling and Applications*, *2*(3), 33-39. https://doi.org/10.11648/j.mma.20170203.12

Jørgensen, M., & Moløkken-Østvold, K. (2004). How large are software cost overruns? A review of the 1994 CHAOS report. *Empirical Software Engineering*, *9*(4), 297-309.

Jørgensen, M., & Shepperd, M. J. (2007). Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, *33*, 33-53.

Jørgensen, M., & Sjøberg, D. I. (2001). Impact of experience on software project estimation accuracy. *Empirical Software Engineering*, *6*(4), 305-334.

Juran, J. M. (1988). *Juran's Quality Control Handbook* (4th ed.). New York: McGraw-Hill.

Krishnan, M. S. (1998). The role of team factors in software cost and quality: An empirical analysis. *Information Technology & People*, *11*(1), 20-35.

MacDonell, S., & Shepperd, M. (2013). Evaluating Prediction Systems in Software Project Estimation: Complementary Techniques and Benchmarks. *Empirical Software Engineering*, *18*(1), 1-14.

Melo, C. d., Cruzes, D. S., Kon, F., & Conradi, R. (2013). Interpretative case studies on agile team productivity and management. *Information and Software Technology*, *55*(2), 412-427.

Mittas, N. (2011). Evaluating the Performances of Software Cost Estimation Models through Prediction Intervals. *Journal of Engineering Science and Technology Review*, *4*(3), 266-270.

Mittas, N., & Angelis, L. (2010). Visual comparison of software cost estimation models by regression error characteristic analysis. *Journal of Systems and Software*, *4*(83), 621-637.

Moser, R., Pedrycz, W., & Succi, G. (2008). A Comparative Analysis of the Efficiency of Change Metrics and Static Code Attributes for Defect Prediction. 20th International Conference on Software Engineering & Knowledge Engineering (SEKE), (pp. 181-190).

Nguyen, T. H., Huang, P.-M., Chien, C.-F., & Chang, C.-K. (2023). Digital transformation for cost estimation system via meta-learning and an empirical study in aerospace industry. *Computers & Industrial Engineering*, *184*, 1-13. https://doi.org/10.1016/j.cie.2023.109558

Nguyen, V., Deeds-Rubin, S., Tan, T., & Boehm, B. W. (2007). A SLOC counting standard. *The 22nd International Annual Forum on COCOMO and Systems/Software Cost Modeling*.

Papatheocharous , E., & Andreou , A. S. (2012). Software Cost Modelling and Estimation Using Artificial Neural Networks Enhanced by Input Sensitivity Analysis. *Universal Computer Science*, *18*(14), 2041-2070.

Ricci, F., Scott, P. J., & Jiang, X. (2013). A categorical model for uncertainty and cost management within the Geometrical Product Specification (GPS) framework. *Precision Engineering*, *37*, 265-274.

Rosa, W., Madachy, R., Boehm, B., & Clark, B. (2014). Simple empirical software effort estimation model. *8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–4. https://doi.org/10.1145/2652524.2652558

Rosenberg, J. (1997). Some misconceptions about lines of code. *Proceedings Fourth International Software Metrics Symposium*, 137-142.

Software Technology Support Center. (2008). *Software development cost estimating handbook*. Naval Center for Cost Analysis and Air Force Cost Analysis Agency. Retrieved from https://www.dau.edu/sites/default/files/Migrated/CopDocuments/SW%20Cost%20Est%20Manual%20Vol%20I%20rev%2010.pdf

Tokdemir, G., & Cagiltay, N. (2019). Investigating the relationship between SLOC and logical database measures to improve the early estimation of software cost. *International Journal of Software Engineering and Knowledge Engineering*, *29*(3), 401-413. https://doi.org/10.1142/S0218194019500177

*Capt **Stephen Chatterton** is a Cost Analyst at the Air Force Life Cycle Management Center/Financial Management and Comptroller Support Directorate at Wright Patterson Air Force Base, OH. He received a BS in Business Administration from the University of Maryland University College and an MS in Cost Analysis from the Air Force Institute of Technology.*

*Dr. **Edward D. White** is a Professor of Statistics in the Department of Mathematics and Statistics at AFIT. He received his BS in Mathematics from the University of Tampa, MAS from The Ohio State University, and PhD in Statistics from Texas A&M University. His primary research interests include statistical modeling, simulation, and data analytics.*

*Dr. **Jonathan D. Ritschel** is a Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, MS in Cost Analysis from AFIT, and PhD in Economics from George Mason University. His research interests include public choice, defense cost economics, cost analysis and economic institutional analysis.*

*Dr. **Brandon M. Lucas** is an Assistant Professor of Systems Integration and Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BA in History from the University of Texas at Austin, an MS in Cost Analysis from AFIT, and a PhD in Economics from George Mason University. Lucas's research interests include profit analysis, cost and economic analyses, and incentive structures.*

*Dr. **Robert D. Fass** is an Assistant Professor in the Department of Systems Engineering and Management at AFIT. He holds a BA in Economics, an MBA from the University of New Mexico, and a PhD in Business Administration and Management from New Mexico State University. Dr. Fass's research interests include cost analysis, decision analysis, risk analysis, behavioral economics, organizational behavior, organizational change, and government acquisition policy.*

*Mr. **Shawn M. Valentine** is an Operations Research Analyst at AFLCMC/FZCE at Wright Patterson AFB, Ohio. Mr. Valentine graduated from Ohio University with a Bachelor of Science (BS) in Actuarial Science and a Master of Science (MS) in Financial Economics.*

Assessing Cost Growth Correlations between Work Breakdown Structures and System Test and Evaluation

Kyle P. Marquis
Edward D. White
Brandon M. Lucas
Robert D. Fass
Jonathan D. Ritschel
Shawn M. Valentine

In support of two knowledge areas of the Program Management Body of Knowledge, project cost management and project risk management, we analyze primarily Level 2 Work Breakdown Structure (WBS) elements of approximately 25 historical United States Air Force (USAF) Acquisition Category I Research Development Test and Evaluation (RDT&E) aircraft programs. Specifically, we employ ordinary least squares regression analysis to examine how the cost growth trends of various Level 2 WBS elements impact overall program cost growth. This analysis primarily focuses on the System Test and Evaluation (ST&E) WBS element, which has been identified as commonly neglected by decision makers yet significantly influential on program cost growth in prior research (Rosado, 2011). This analysis continues Rosado's work with a more comprehensive data set, a focus on USAF RDT&E aircraft programs, and an analysis of other Level 2 WBS elements. The results appear to confirm Rosado's exploratory conclusions. Further, our findings establish ST&E as the only second level WBS element that significantly correlates to program cost growth and that is influenced by improved cost and risk planning.

Background

Program Managers (PMs) require accurate cost forecasts to effectively manage their projects. The ability to generate reliable cost estimates is a critical function for federal agencies and is necessary to support the Office of Management and Budget's capital programming process (Government Accountability Office [GAO], 2020). Without this ability, agencies are at risk of experiencing cost overruns, missed deadlines, and performance shortfalls—all recurring problems that program assessments too often reveal. For a cost forecast, PMs generally rely on the Estimate at Completion (EAC) to forecast each component of their program. An EAC is normally derived from updating EACs at the control account level and summarizing up the Work Breakdown Structure (WBS) allowing for an EAC at any level (Defense Acquisition University [DAU], 2023).

EACs are generated for the total program as well as for different WBS elements in the WBS hierarchy (Department of Defense [DoD], 2022). EACs play a key role in establishing the funding required for a DoD acquisition program. The most difficult but also the most crucial EACs are produced at the beginning of the program. The initial EAC becomes the baseline for which the rest of the program's life cycle is compared. Faced with the greatest degree of cost risk, this initial EAC is likely subject to the greatest degree of inaccuracy (Smith et al., 2012).

Baselines from program offices tend to represent idealized conditions and fail to consider the complexity of developing new technology and the difficulty of integrating that technology into new systems (GAO, 2020). The System Test and Evaluation (ST&E) WBS element encompasses many tests that a system undergoes during its lifecycle from development to deployment. These

tests verify that the system is performing as designed and validate that the design meets the needs of the stakeholders (DoD, 2022). There are two major components of testing for DoD programs, Developmental Test and Evaluation (DT&E) and Operational Test and Evaluation. For this article, we only investigate the ST&E WBS element encompassing DT&E.

The contractor developing the system usually conducts the DT&E throughout the development phase of a system's design process. The testing entails the use of models, simulations, test beds, and prototypes or full-scale engineering development models of the system to verify that technical performance metrics are being met. Tests of this kind are also used whenever design changes are made to ensure the system in its entirety still meets specifications. The test data provides valuable information for PMs to make decisions on cost, schedule, and performance tradeoffs (DoD, 2022).

In 2016, the Deputy Secretary of Defense directed the Defense Business Board to form a task group to evaluate the DoD's management and use of the test and evaluation enterprise. The task group interviewed officials and experts from within DoD, other government agencies, and the private sector. They reviewed applicable statutes, DoD policies, and DoD directives. The task group also reviewed strategic documents, reports, studies, and briefing papers from both the DoD and industry (Phillips, 2016). The task group concluded that accurate tracking of ST&E costs is generally not viewed as a priority by the DoD, leading to a lack of incentive to control costs. They also found that ST&E garners unnecessary cost and schedule pressure due to too many approval authorities, inconsistent accountability for risk trade-off decisions, and lack of resources for additional testing when required (Phillips, 2016).

These factors all point towards ST&E as a low priority within the DoD in terms of cost control and risk management. This low priority of ST&E may not be optimal for assessing and perhaps controlling cost overruns in a program given the preliminary findings of Rosado (2011). Rosado's analysis suggested a positive correlation between changes in the ST&E WBS element and the overall program's EAC. Although significant, this finding was subject to data limitations.

The data for Rosado's thesis originated from the Earned Value Management Central Repository (EVM-CR). At the time, extracting data from the EVM-CR required a substantial effort as the data in the repository was stored in many forms, including Adobe Reader (.pdf), Power Point (.ppt), Excel (.xls), Extensible Markup Language (.xml), and Deltek wInsight (.wsa). Duplicated programs, issues with reporting, and missing files from the EVM-CR forced Rosado to pivot his work from a USAF focus to a DoD wide focus to have sufficient data points for regression analysis (Rosado, 2011).

This lack of data also prevented Rosado from omitting programs when the reports did not capture most of each program's duration, thereby potentially skewing the findings of his work. Figure 1 shows the reported program life for each program in Rosado's thesis. The coverage of programs in the reported files varied greatly, and rarely covered more than 70% of each program's duration. Despite this limitation, Rosado was able to model a statistically significant relationship between ST&E and EAC % cost growth, which is the same definition of cost growth that we identify later in this article. His findings in later years prompted the *Defense Acquisition Research Journal* (DARJ, 2019) to direct their readers to investigate this unpublished thesis.

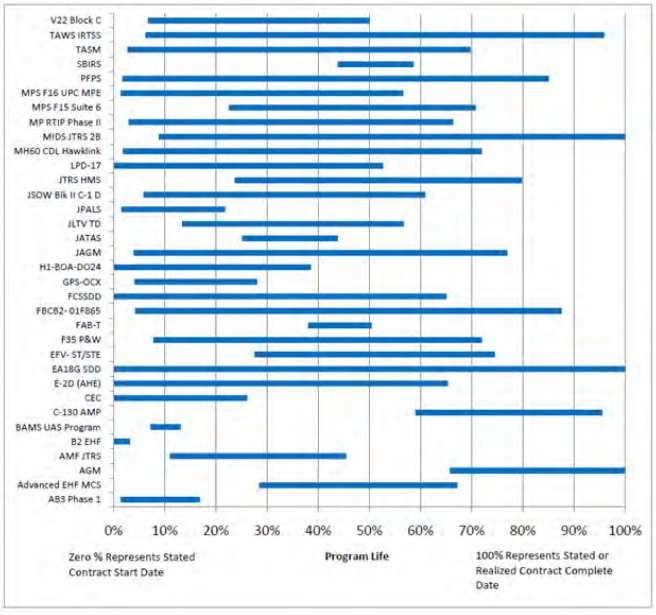


Figure 1. Completion coverages for the studied programs in Rosado (2011).

In this paper, we further investigate Rosado's findings that suggest ST&E influences total program cost growth. More importantly, we have more current and granular data. Due to the work of the Air Force Life Cycle Management Center (AFLCMC)'s Financial Management directorate, a large collection of aircraft EAC data has been compiled into a standardized database called Project Ginger II. Project Ginger II is an effort to standardize the data from the Cost Assessment Data Enterprise (CADE) database into a singular Microsoft Access File. This standardization allows a larger snapshot of a program's historical cost growth. With this data, we can not only focus on how cost growth of the EAC for ST&E affects the overall cost growth of a system's total EAC but also how ST&E's influence compares to the influence of six other common WBS elements: Data, Fee, Prime Mission Equipment (PME), Support Equipment (SE), System Engineering and Program Management (SEPM), and Training.

Data and Methods

The data for this analysis was originally collected from the CADE database and incorporated into Project Ginger II. CADE is the USAF's centralized repository of comprehensive cost, schedule, software and technical data. CADE contains Contractor Cost Data Reports, also known as DD 1921s. These are the primary means within the DoD to systematically collect actual data on the development and production costs incurred by contractors performing DoD acquisition contracts (CADE, 2023). DD 1921s provide a contract level overview of cost data by WBS element, to date and at completion. The 1921s report costs from WBS Level 1 all the way down to a WBS Level 5 or 6. Cost, Fee, General and Administrative, Price, and Miscellaneous are also reported.

As of July 2023, our initial data pull consisted of 59 Acquisition Category I (ACAT I) research, development, test, and evaluation (RDT&E) aircraft programs separated by contract number and then by WBS element. These aircraft programs include such commodities as aircraft, rotary wing, and engines. A Major Defense Acquisition Program is an ACAT I program if it has (a) total expenditure of RDT&E costs greater than $525 million (fiscal year [FY] 2020 constant dollars); (b) total expenditure of procurement costs greater than $3.065 billion (FY 2020 constant dollars); or (c) is specifically designated by the milestone decision authority as special interest (DoD, 2020).

Each program's DD 1921 captures the total EAC at the time of the report (Block J) as well as the current total expended amount for each WBS element (Block F). The percent complete for each report is calculated by dividing the current expended amount by the final reported EAC (Block F / Block J). These EACs are not always reported at the beginning and at the end of the program for each WBS. Some initial EACs report as late as 60% complete, and some final EACs report as early as 40% complete. To minimize the chance of skewing the results from relatively incomplete programs, we establish initial and ending thresholds for each reported WBS element for program inclusion in the analysis.

For this, we turn to published literature for these thresholds. Christensen and Heise (1993) report that cumulative Cost Performance Index stability has been found for USAF contracts at or after 20% of a program's completion. Even though recent work by Kim et al. (2019) suggests this percentage might be later, we accept this 20% as the maximum initial threshold percentage for a reported WBS element. For the ending threshold, we turn to Tracy and White (2011). Their analysis suggests that EACs at 92.5% or more complete are not statistically different from the final program cost. Consequently, we require the final EAC report to reach at least this threshold of 92.5% completion.

The cost growth for each WBS element in each program is intended to capture the ratio of the final cost to the initial projected cost calculated at the beginning or prior to the start of the program. To capture this information, EAC data is used. We investigate all the 1921s of the 59 ACAT I RDT&E programs that we pulled from Project Ginger II. For each WBS element, we only include those whose EACs report at 20% complete or earlier and those final EACs report at 92.5% complete or later. Cost growth is calculated as the latest EAC divided by the earliest EAC. All EAC dollar amounts are normalized to Constant Price (CP) 2023 dollars using the Aerospace Producer Price Index 3364 (USAF, n.d.). Equation 1 is used to calculate the cost growth for each element or WBS of a program.

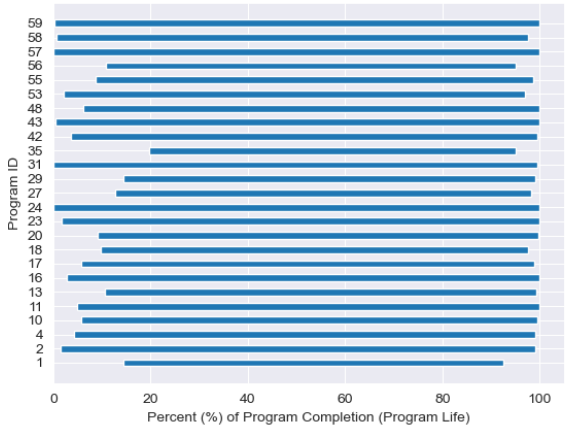


Figure 2. Completion coverages for the studied aircraft programs in this article.

$$\text{Cost Growth} = \frac{EAC_{Final}}{EAC_{Initial}} \quad (1)$$

In total, only 25 out of the 59 programs report 1921s that span from at or close to program inception to its conclusion. Figure 2 displays those coverages for the aircraft programs in our study's database. For confidential purposes, we do not link any program to any specific metric in the article, although Table 1 does list the programs that we used in our analysis. Note that Figure 2 better encompasses the development lifespan of a program with respect to reporting EACs than Figure 1.

Besides investigating Cost and Fee associated with a program [Note: We don't investigate Price since Cost plus Fee constitutes Price.], we also focus on the common Level 2 WBS of Data, Prime Mission Equipment (PME), Support Equipment (SE; sum of Peculiar and Common Support Equipment), Systems Engineering and Program Management (SEPM), System Test and Evaluation (ST&E), and Training. Due to lack of available data, we do not conduct any cost growth analysis on the other common Level 2 WBS elements associated with Initial Spares, Site Activation, or Industrial Facilities. This lack of data for site activation and initial spares is also noted by Markman et al. (2021).

For programs in Project Ginger II where multiple contractors have efforts on a project, each contractor has its own set of reported WBS elements. This segmentation includes total cost for that contractor's work. Contractors are identified as "primes" or "subs" for each program. Prime contractors are ultimately responsible for product delivery, integrating components developed by their subcontractors, and other system engineering type work. Subcontractors are responsible for different components of the project and work directly with the primes.

A-10A Full Scale Development	F135 Engine
AC-130U Full Scale Development	F136 Prototype Engine
AH-64E Apache (Formerly AB3)	F-22A Full Scale Development
AN/APQ-181 - Mod	HH-60W Combat Rescue Helicopter (CRH)
ARH-70A System Development & Demonstration (SDD)	JSTARS Radar Subsystem (E-8C)
B-2 DMS: Defensive Management System & EHF SATCOM and COMPUTER	RAH-66 Full Scale Development
C-130J Aircraft System Block 6.0	UH-1H Composite Main Rotor Blade Development
C-17A Full Scale Development	USMC H-1 Upgrade for AH-1Z (4BW)
C-5 Reliability Enhancement and Re-Engining (RERP)	USMC H-1 Upgrade for UH-1Y (4BN)
CH-53K Full Scale Development	H1 Upgrade Production Program - UH-1Y; China Lake
E-2D AHE Full Scale Development - Advanced Hawkeye Development	XF119-PW-100
EA-18G Full Scale Development	YF120-GE-100
F/A-18E/F Full Scale Development	

Table 1. Aircraft Programs Utilized in the Study

In Project Ginger II, reports from prime contractors usually include roll ups from all the subcontractor costs and EACs as well as their own. A roll up identifier is included in the Project Ginger II database to identify which contractor reports include roll ups of subcontractor costs called parent reports. If a program includes a parent report and a single WBS element with multiple reported EACs, the sub reports and EACs for that WBS element are ignored.

In this event, the prior Equation 1 is used to calculate the total program cost growth for each WBS element via that parent report.

If a prime contractor reports the same type of WBS element twice or more, the sum of each final EAC is divided by the sum of each initial EAC for each duplicate WBS element to calculate cost growth. Similarly, if no prime contractor is reported, the sum of each subcontractor's final EAC is divided by the sum of each initial EAC for each contractor and each contractor's duplicate WBS elements where applicable. For example, a prime submits a 1921 report and the Training WBS line is repeated two times because of two subs doing the training for separate EACs. Equation 2 captures the calculation of cost growth for these last two events where i represents each duplicate WBS element and k is the number of duplicates. As a reminder, all calculations for (1) and (2) only involve those 1921s whose WBS information corresponds to having the initial EAC being equal to or less than 20% complete, and a final EAC that is 92.5% or more complete.

$$\text{Cost Growth} = \frac{\sum_{i=1}^{k} EAC_{Final_i}}{\sum_{i=1}^{k} EAC_{Initial_i}} \quad (2)$$

This article seeks to further investigate Rosado's claim that the level 2 WBS element ST&E has a positive correlation with overall program EAC cost growth using ordinary least squares (OLS) but only with respect to aircraft programs. To perform OLS, we need to designate a dependent variable, the response, and a set of independent variables, the explanatory variables. The dependent variable in our analysis is the Cost WBS element, which represents the total EAC cost growth of a program. The EAC cost growth for every other WBS element is considered an independent variable. The value for each variable is calculated by finding the total program cost growth for that respective WBS element by either cost growth equations (1) or (2) as discussed earlier.

Since we are concerned with assessing relational correlations and not necessarily building a predictive model, we are not concerned if a Box-Cox (Box & Cox, 1964) transformation may be needed to account for non-normality and heterogeneity of OLS residuals. Additionally, we are not concerned with overly influential data points as assessed by Cook's D (Cook, 1977) or issues with multicollinearity as measured by variance inflation factors (VIFs). Lastly, to counteract a possible overweighting of a WBS element affecting the total Cost element due to its magnitude, we also incorporate a covariate where we account for element size. This is akin to standardizing the inputs into a regression model. In the next section, we present the findings of our single variable regressions and covariate OLS analyses.

Results

For the output presented in this section, we primarily used the Python libraries of Pandas and NumPy in addition to the JMP 15 Pro software. We first display the calculated EAC cost growths by WBS element. As evident in Table 2, there are a few very high values. Program 31 only has three values, and all are quite high. Additionally, Program 57 has a high Fee EAC cost growth, while Program 24 has a high EAC cost growth for the level 2 WBS element Training. Because these values are very likely to skew any correlation or OLS analyses, we excluded these values going forward. These exclusions remove Program 31, thereby reducing our sample size from 25 ACAT I aircraft programs to 24.

Figure 3 highlights the correlations between the total program EAC cost growth (Cost) and the other element EAC cost growths, while Table 3 presents the sample medians, means, standard deviations, and interquartile ranges for the EAC cost growths of our 24 aircraft programs. Due to missing data for some of the elements, namely Fee, some of the correlations needed to be estimated by pairwise method. Table 3 highlights that most of the elements have positive EAC cost growth, which is in keeping with Jones et al. (2023) findings of continuous DoD program cost growth over the years. Figure 3 reflects relatively high correlation between Cost and the level 2 WBS elements of ST&E, SEPM, and PME in descending significance respectively. Additionally, the correlation of 0.89 between SEPM and ST&E suggest a high degree of association between these elements' EAC cost growths.

1	1.45			1.34	4.2		1.65	0.68
2	2.81		2.21	2.34	7.76	3.90	5.10	0.94
4	1.54		1.63	1.92	1.26	1.44	1.82	1.61
10	1.46	0.43	0.47	2.06	0.77	1.76	0.69	0.98
11	3.01		0.8	1.95	0.68	1.20	1.05	0.85
13	1.32	0.9	1.66	1.48	0.81	1.06	1.64	0.29
16	1.39	1.57	1.68	2.52	0.83	1.01	1.22	1.20
17	1.99		2.65	5.74		4.26	2.86	2.53
18	1.44		2.15	1.39	1.07	1.10	1.17	
20	1.62	1.50	0.81	1.83	0.86	2.89	1.81	0.90
23	1.3	1.13	1.46	1.33	0.72	1.38	1.24	0.75
24	1.13	1.18	3.6	1.18	0.85	0.94	1.23	15.57
27	1.08			1.28	1.94	0.96	1.02	
29	1.77	1.08	0.61	1.84	1.21	1.61	1.71	0.82
31	26.93			12.62		22.54		
35	1.38		0.84		2.52		1.46	
42	1.25	0.10	1.91	1.80		1.04	1.03	0.81
43	2.05		0.74	1.66	1.0	2.15	1.81	4.05
48	1.49	2.44	0.04	1.52	0.71	2.17	1.75	0.43
53	4.84		2.50	3.39		4.48		
55	2.55		0.14	2.59	14.44	2.07	1.87	0.66
56	2.01		0.14	1.90		1.40	0.89	0.23
57	3.04	10.84	1.42	7.42		0.81		
58	5.68			5.56		5.54	6.24	
59	2.82		1.41	3.13		2.69	1.90	

Table 2. Calculated EAC cost growths by WBS. Very high values shown in red will be excluded in further analysis.

Metric	Cost	Fee	Data	PME	SE	SEPM	ST&E	Training
Sample Size	24	9	21	23	17	22	22	16
Mean	2.10	1.15	1.37	2.49	2.45	2.08	1.87	1.11
Median	1.58	1.13	1.42	1.9	1	1.53	1.65	0.84
Standard Deviation	1.15	0.68	0.93	1.62	3.58	1.34	1.33	0.95
Inter-quartile Range	1.36	0.87	1.36	1.11	1.44	1.69	0.69	0.48

Table 3. Sample EAC cost growth metrics for the 24 ACAT I aircraft programs in the study's analyses.

Correlations

	Cost	Fee	Data	PME	SE	SEPM	ST&E	Training
Cost	1.0000	0.2753	0.0717	0.6178	0.5237	0.7397	0.7927	0.1290
Fee	0.2753	1.0000	-0.2899	-0.0767	-0.2085	0.4328	0.6475	-0.1956
Data	0.0717	-0.2899	1.0000	0.1669	-0.1801	0.1834	0.2670	0.2497
PME	0.6178	-0.0767	0.1669	1.0000	0.5378	0.4594	0.6685	0.3710
SE	0.5237	-0.2085	-0.1801	0.5378	1.0000	0.4345	0.4726	-0.1610
SEPM	0.7397	0.4328	0.1834	0.4594	0.4345	1.0000	0.8883	0.3469
ST&E	0.7927	0.6475	0.2670	0.6685	0.4726	0.8883	1.0000	0.1847
Training	0.1290	-0.1956	0.2497	0.3710	-0.1610	0.3469	0.1847	1.0000

There are 17 missing values. The correlations are estimated by Pairwise method.

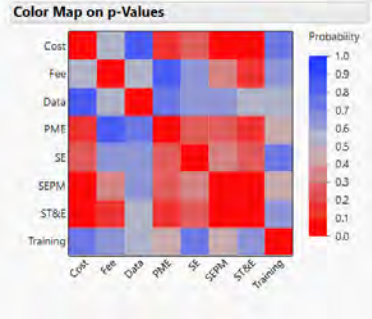


Figure 3. Correlation table and associated p-value heat map between Cost EAC cost growth and the EAC cost growths of the other WBS elements. Significant correlations shown in red.

Table 4 highlights the single variable regression analysis of each WBS element and Cost with respect to total EAC cost growth. We only include the coefficient of determination (McClave et al., 2017) for significant variables. As evident by the very low p-values, both ST&E and SEPM EAC cost growth are highly significant of total EAC cost growth of a program's Cost. PME is also significant with borderline significance of SE.

Element	P-value	R^2
Fee	0.4735	N/A
Data	0.7576	N/A
PME	0.0017	0.38
SE	0.0310	0.27
SEPM	< 0.0001	0.55
ST&E	< 0.0001	0.63
Training	0.6339	N/A

Table 4. P-values and associated R2 values for those variables significant at the 0.05 level of significance.

Considering there may be a natural cause-and-effect between SEPM and overall Cost (which we discuss in the Conclusion section), we redid the analysis of Table 4 while introducing a covariate for all three variables. The covariate in the context of this analysis is defined as the final EAC of each WBS element divided by the final Cost EAC for that program. This is not to be confused with the cost growth for each WBS element.

The intent of this covariate analysis is to identify if any of the WBS elements are predictive of program cost growth based on their percent contribution to the program cost. This is achieved by creating a multiple OLS model for each WBS with the EAC cost growth of the WBS element and its covariate value as the explanatory variables. Because only PME, SEPM, and ST&E were significantly correlated with EAC cost growth of Cost, we limit this follow-up analysis to just these three level 2 WBS elements. Table 5 shows those results. We use adjusted R^2 in lieu of R^2 for R^2 can be inflated just by adding more explanatory variables in the regression model (Kutner et al., 2004).

OLS Model	Variable	P-value	Adjusted R^2	VIF
1	PME	0.0017	0.33	1.03
	PME Covariate	0.5188		1.03
2	SEPM	< 0.0001	0.61	1.00
	SEPM Covariate	0.0333		1.00
3	ST&E	< 0.0001	0.60	1.01
	ST&E Covariate	0.6023		1.01

Table 5. Multiple OLS results for PME, SEPM, and ST&E while accounting for WBS element size. The adjusted R2 is for each model.

The results from Table 5 confirm what we discovered from Table 4. Both SEPM and ST&E EAC cost growths are highly predictive of total program EAC cost growth with PME being also significant. However, Table 5 now reveals that neither the ratio of final ST&E and PME EAC to total program final EAC is predictive of the actual EAC cost growth of the program; but this is not true

for SEPM. It appears the ratio of final SEPM EAC to final program EAC is significant at the 0.05 level of significance. Lastly, all the VIF scores are very low, which implies no multicollinearity issues and *p*-value stability in our OLS models. We next discuss our findings and make a recommendation going forward.

Conclusion

As indicated at the beginning, PMs require relatively accurate forecasts of cost, schedule, and performance to manage and to administer their projects effectively. With respect to cost, PMs generally rely on the EAC to forecast each component of their program. The initial EAC becomes the baseline for which the rest of the program's development cycle is compared. Rosado's (2011) analysis suggested a positive correlation between EAC cost growth of the ST&E WBS element and a program's overall EAC cost growth. Although statistically significant, his preliminary findings were caveated by a limited sample size.

Due to the difficulties of EVM-CR at that time, Rosado was required to amalgamate a collection of DoD programs and platforms to obtain a statistically robust database for analysis. Using CADE data via Project Ginger II, we were able to generate a focused dataset to perform confirmatory analyses with respect to ACAT I RDT&E aircraft programs. Additionally, as shown in Figure 2 in comparison to Figure 1 in Rosado (2011), we were able to obtain enough aircraft programs whose duration spanned most of the program to obtain realistic initial EAC and final EACs.

Overall, our analysis does confirm that there appears to be a very strong statistical association between EAC cost growth for the level 2 WBS element ST&E and a total program's EAC cost growth. We also detected a very strong statistical association for the level 2 WBS element SEPM; and a strong association for the level 2 WBS element PME with respect to total program EAC cost growth. Since SEPM consists of both systems engineering and program management, this WBS element is often referred to as overhead because these disciplines are mainly manpower costs and are not part of any actual work being completed.

When we look at the mean percentage of SEPM cost to the final program cost, we observe a value of approximately 11% for our 24 programs. In a macro perspective, one can consider this percentage as the typical overhead cost of an aircraft program. The covariate analysis indicates that as a program increases in cost so does the associated overhead cost. This observation makes logical sense. Putting everything together, we conclude that even though the EAC cost growth of SEPM is highly correlated to EAC cost growth of a program, we must recognize there is overarching overhead cost, which by default will increase as the program costs increase. From a PM perspective, there is not much to really control or recommend here. The statistics appear to support the reality of managing a program.

However, when we look at the results of PME and ST&E, we get a different perspective. The covariate analysis did not show any statistical association of the cost of the level 2 WBS element to the final cost of program, but the EAC cost growth factors were associated. That is, as the EAC cost growth factor of either PME or ST&E increases, so does the EAC cost growth factor of the total program. When we investigate the mean percentage PME costs to the total program cost, our 24 programs report a value of 59%. Since the PME WBS element encompasses technology development, engineering efforts, integration efforts, final design, and production of the entire primary system, this relatively high percentage makes logical sense. Furthermore, one would expect the EAC cost growth of PME would be *a priori* highly correlated to EAC cost growth of the total program, since PME is typically a huge majority of a program. Again, from a PM perspective, there is not much to really control or recommend here. As before, the statistics appear to support the reality of managing a program.

When turning our attention to ST&E, we find an interesting detail. When we investigate the mean percentage ST&E costs to the total program cost, our 24 programs report a value of 9%. This value is a much smaller percentage than that for PME and suggests that ST&E is not a majority of a program's final total cost. Therefore, one would not necessarily expect the EAC cost growth change of ST&E would strongly affect the EAC cost growth of the total program; but that is not what we observed. In fact, we saw a magnitude of statistical significance of at

least 10 times that demonstrated by PME. Thus the implication is though overall ST&E is a small piece of the final program cost, the impact of EAC cost growth of the ST&E WBS element is very strong on the EAC cost growth for the entire program.

Overall, ST&E appears to be the most impactful WBS element on the total cost of an aircraft program; that is, both are perhaps unappreciated and can be greatly influenced by additional efforts of the PM and the PM's Integrated Product Team. This revelation signals a need for greater emphasis on planning and controlling costs for the ST&E WBS element, which has historically been among the first of the second level WBS elements to experience budget cuts and a decrease in scope (Phillips, 2016). We recognize a limitation of our study is the results may only be applicable to DoD ACAT I aircraft programs. However, coupled with Rosado's preliminary findings (2011), we suggest this ST&E relationship to overall program cost growth might be applicable to other platforms. We encourage other researchers to perform these additional analyses.

References

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological) 26*(2), 211-252. https://www.jstor.org/stable/2984418

CADE. (2023). *Cost assessment data enterprise home page*. OSD CAPE. Retrieved from https://cade.osd.mil/about (https://cade.osd.mil/about)

Christensen, D., & Heise, S. R. (1993). Cost performance index stability. *National Contract Management Journal 25*(1), 7-15.

Cook, D. (1977). Detection of influential observation in linear regression. *Technometrics 19*(1), 15-18. https://doi.org/10.2307/1268249

Defense Acquisition University. (2023). *Estimate at completion*. Defense acquisition university. Retrieved from https://www.dau.edu/acquipedia-article/estimate-completion-eac

Defense Acquisition Research Journal. (2019). New Research in Defense Acquisition. *Defense Acquisition Research Journal 26*(1), 84-88.

Department of Defense. (2020). *Major capability acquisition (DoDI 5000.85)*. Office of the Under Secretary of Defense for Acquisition and Sustainment. https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500085p.pdf?ver=2020-08-06-151441-153

Department of Defense. (2022). *MIL-STD-881F: work breakdown structures for defense materiel items*. Department of Defense.

Government Accountability Office. (2020). *Cost estimating and assessment guide: Best practices for developing and managing program costs* (GAO-20-195G). Government Accountability Office.

Jones, S. J., White, E. D., Ritschel J. D., & Valentine, S. M. (2023). Cost estimation trends for major defense acquisition programs. *Defense Acquisition Research Journal 30*(2), 96-123. https://doi.org/10.22594/dau.22-894.30.0

Kim, D. B., White, E. D., Ritschel, J. D., & Millette, C. A. (2019). Revisiting reliability of estimates at completion for Department of Defense contracts. *Journal of Public Procurement 19*(3), 186-200. https://www.emerald.com/insight/content/doi/10.1108/JOPP-02-2018-0006/full/html

Kutner, M., Nachtsheim, C., & Neter, J. (2004). *Applied linear regression models* (4th ed.). McGraw Hill/Irwin Series.

Markman, M. R., Ritschel, J. D., & White, E. D. (2021). Use of factors in development estimates: Improving the cost analyst toolkit. *Defense Acquisition Research Journal*, *28*(1), 40-70. https://doi.org/10.22594/10.22594/dau.19-848.28.01

McClave, J. T., Benson, P. G., & Sincich, T. (2017). *Statistics for business and economics* (13th ed.). Pearson.

Phillips, B. (2016). *Best practices for the business of test and evaluation*. Defense Business Board. Retrieved at https://dbb.defense.gov/Portals/35/Documents/Reports/2017/DBB%2017-01%20Test%20%20&%20Evaluation%20Study%20(Complete)%20Final%20v2.pdf

Rosado, W. (2011). *Comparison of development test and evaluation and overall program estimate at completion* [Unpublished master's thesis]. Air Force Institute of Technology.

Smith, A., McDowell, J., Fitcher, L., Blevins, B., & DeTore, N. (2012). *Cost risk and uncertainty analysis metrics manual* (CRUAMM). Tecolote.

Tracy, S. P., & White, E. D. (2011). Estimating the final cost of a DoD acquisition contract, *Journal of Public Procurement*, *11*(2), 190-205. https://doi.org/10.1108/JOPP-11-02-2011-B002

USAF. (n.d.). Inflation indices. U.S. Air Force Economics and Business Management Directorate. https://usaf.dps.mil/sites/11534/FMCE/SitePages/Inflation%20Indices.aspx

---

*Capt **Kyle Marquis** is currently assigned as an acquisition officer in the Aerospace Systems Directorate at Wright-Patterson Air Force Base (AFB), Ohio. Capt Marquis is a recent graduate of the Air Force Institute of Technology (AFIT) where he earned an MS in Systems Management.*

*Dr. **Edward D. White** is a Professor of Statistics in the Department of Mathematics and Statistics at AFIT. He received his BS in Mathematics from the University of Tampa, MAS from The Ohio State University, and PhD in Statistics from Texas A&M University. His primary research interests include statistical modeling, simulation, and data analytics.*

*Dr. **Brandon M. Lucas** is an Assistant Professor of Systems Integration and Cost Analysis in the Department of Systems Engineering and Management at AFIT. He holds a BA in History from the University of Texas at Austin, an MS in Cost Analysis from AFIT, and a PhD in Economics from George Mason University. Lucas's research interests include profit analysis, cost and economic analyses, and incentive structures.*

*Dr. **Robert D. Fass** is an Assistant Professor in the Department of Systems Engineering and Management at AFIT. He holds a BA in Economics, an MBA from the University of New Mexico, and a PhD in Business Administration and Management from New Mexico State University. Dr. Fass's research interests include cost analysis, decision analysis, risk analysis, behavioral economics, organizational behavior, organizational change, and government acquisition policy.*

*Dr. **Jonathan D. Ritschel** is a Professor of Cost Analysis in the Department of Systems Engineering and Management at AFIT. He received his BBA in Accountancy from the University of Notre Dame, MS in Cost Analysis from AFIT, and PhD in Economics from George Mason University. His research interests include public choice, defense cost economics, cost analysis and economic institutional analysis.*

*Mr. **Shawn M. Valentine** is an Operations Research Analyst at AFLCMC/FZCE at Wright Patterson AFB, Ohio. Mr. Valentine graduated from Ohio University with a Bachelor of Science (BS) in Actuarial Science and a Master of Science (MS) in Financial Economics.*

Market Dimensional Expansion, Collapse, Costs, and Viability

Douglas K. Howarth

Most government programs have cost caps and minimum force requirements. Commercial projects usually begin with a budget, sales targets, and specifications. All too often, in both cases, producers and customers give little thought to the changing market structures they face. When it comes to Demand, markets self-organize to form up to four boundaries each, including 1) Upper (price-limited), 2) Outer (saturation-limited), 3) Inner (efficiency-limited), and 4) Lower (margin-limited) Demand Frontiers. When new market segments appear as different product forms with enhanced functionality over existing options, as the new markets grow, the product groupings they replace may contract across one or more Demand Frontiers. This paper examines preparing for these inevitable eventualities in an N-dimensional framework.

Introduction

Standard economic theory works for simple commodities with one primary feature, such as those for iron ore, platinum, silver, or gold. In those cases, simple supply and demand models dating back to the 1800s work well.

We can better describe all other markets with four-dimensional systems, in which a three-dimensional Value Space shares a common price axis with a two-dimensional Demand Plane, forming a 4D system. Since all of them share the price axis, with appropriate dimensional expansion and collapse, it is possible to portray each simultaneously and over time. Expansions and collapses of markets have happened since their dawn. This paper examines their nature in both cases, offering insights into how to deal with market expansion and collapse.

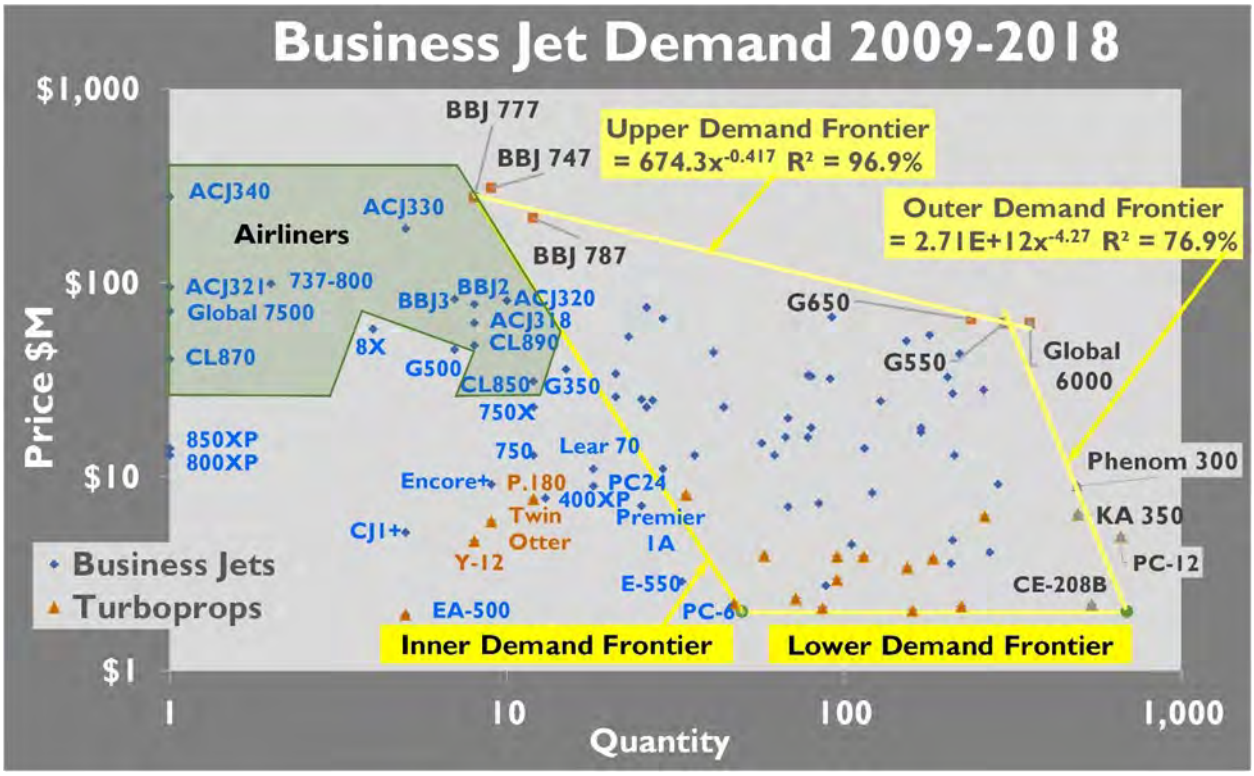


Figure 1 – This Market has Minimum, Upper, Outer, and Lower Demand Frontiers [1]

What is a 4D System?

We can describe the location of any vessel using latitude and longitude. Current measurements using these metrics often use as many as six places past the decimal point for the degrees. With one degree of latitude taking about 69 miles of arc, we can plot and pinpoint a ship's position within a few inches. That amounts to a 2D coordinate system.

If we wanted to plot a plane's location in flight, we could add elevation to our plotting scheme, revealing a 3D coordinate system.

As we'll discover presently, a 4D market plot takes one 2D system and appends it to an appropriate one in 3D. The trick, such as it is, is that 4D systems exist in market or mathematical realms and do not have direct connections to those we find in physical systems.

The simplest way to start building a 4D market system is to plot its Demand over a fixed period. If, for example, we were to plot the Quantity of business aircraft sold from the beginning of 2009 to the end of 2018 on the horizontal axis and their average prices on the vertical axis, we would have the view we get in Figure 1

Figure 1 shows that Business Aircraft Demand is not the simplistic single line drawn in virtually every introductory textbook on economics. Instead, it consists of a series of points, with the quantities of a given model sold shown on the horizontal axis and their average Price on the vertical. It has boundaries. Specifically, it may have up to four distinct limits, known as Demand Frontiers.

The *Upper Demand Frontier* represents the price-limited constraints the market imposes upon itself. It is impractical to try to target sales far beyond the statistical limits of this line, as Aerion found out with its AS2 Supersonic Business Jet. Aiming for 300 units sold in a decade at an average price of $120, they only received 20 firm orders and went bankrupt. [2]

An *Outer Demand Frontier* exposes the threshold at which the market saturates itself with products and represents the quantity limit markets set for themselves. A Price reduction in this market may result in a few more sales, but inevitably, the market cannot absorb more products.

Often, *Lower Demand Frontiers* reveal themselves, as shown in Figure 1, where a lower boundary of $2 million a plane has formed. At such low prices, there

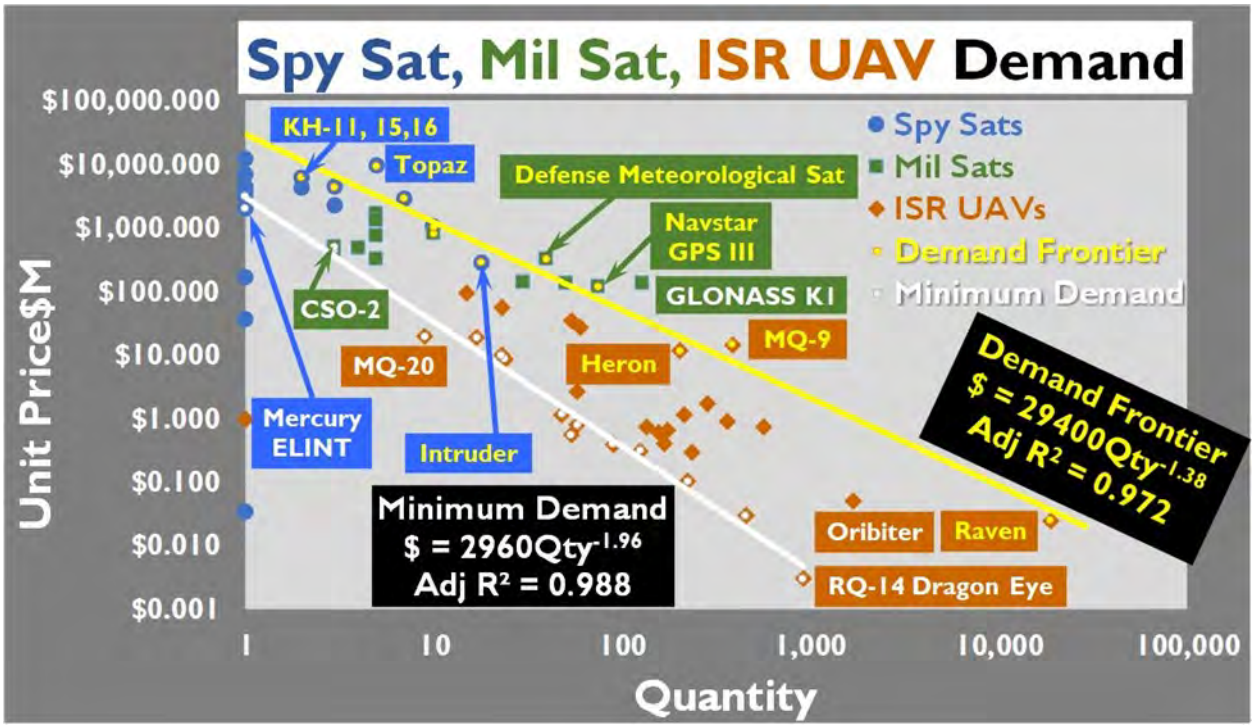


Figure 2 – This Market has highly correlated Minimum and Outer Demand Frontiers [3]

is not enough margin between the Price of the plane and its recurring cost. Someone can build an aircraft for less than $2 million in a category with lower standards, as all or most General Aviation planes sell for less than that amount. Still, such vehicles do not meet the stricter FAA requirements for turboprops and business jets.

Inner Demand Frontiers represent the minimum Quantity needed to keep a production line going, which varies according to Price. Sometimes, this line is highly correlated, as we find studying unmanned reconnaissance platforms in Figure 2. If we collect the quantities sold and prices for Unmanned Air Vehicles (UAVs) and add their military and spy satellite counterparts, we get the view we see in Figure 2.

We observe in Figure 2, examining a market created by and solely used for governments, that a sharp Minimum Demand line has formed for these devices. It essentially relates to us that if we are good enough to get past the first unit (and note that a few models were not, and made just one version of the model, and were stopped), we can be assured of making a certain baseline amount we can call *Minimum Demand.* This threshold is a common feature of government markets, as the same limits form in the markets for bombs, missiles, fighters, and bombers.

But, in the commercial world, things are different. Back in Figure 1, that dividing line is fuzzy. In this market, in the shaded area, companies such as Airbus, Boeing, and Bombardier, all of whom produce airliners, can stand relatively low rates for their business jets, as those planes are modifications to units coming off their production lines. Their production lines can continue to make planes as long as Demand exists for the models in question. Models that command fewer sales than this line are more likely to stop production than those past it, as shown in Figure 3.

Some companies, such as Textron, failed to make enough planes to keep their production lines running efficiently and went out of business, taking their four models at the bottom of Figure 3 with them. One of the early market entrants, Learjet, which had been in business for nearly 60 years, fell victim to the same fate. Bombardier, Gulfstream, and Pilatus ceased building the CL-850, G350, and PC-6, respectively,

Model	Status Jan 2024
Gulfstream G500	In production
Falcon 8X	In production
Harbin Y-12	In production
Viking Twin Otter	In production
Avanti P.180	In production
Pilatus PC-24	In production
Cessna CJ1+	Stopped
Pilatus PC-6	Stopped
Eclipse E-550	Stopped
Learjet 70	Stopped
Bombardier CL-850	Stopped
Gulfstream G350	Stopped
Extra EA-500	Stopped
Textron Premier 1A	Stopped
Textron 400XP	Stopped
Textron 750	Stopped
Textron 750X	Stopped
Cessna Encore +	Stopped

Figure 3 – Aircraft models below the Inner Demand Frontier *are more likely to cease production than those past that line*

but all had other models that succeeded them. Cessna ceased the Encore+ and CJ1+ and also used replacement models. The Eclipse 550 was an attempt to revive the poorly conceived Eclipse 500, and that program went under with just 33 sales. [3] The Extra EA-500 was an attempt to compete with the thousands of Pilatus PC-12 and the Socata TBM series that the market loved, and it and the company ceased operations after 14 years of production.[4]

The models in production in Figure 3 that were in production were either gearing up to full rate production (as in the case of Viking with its Twin Otter, Pilatus with its PC-24, Gulfstream with the G500, and Harbin with its latest version, the "F," of its Y-12 model) or when faced with thin sales, relied on heavy subsidies from their respective governments (this includes Dassault, with their Falcon 8X, receiving subsidies from the French government, and Avanti, getting money from the Italian authorities). [5], [6]

So, while we've seen how buyers react to the prices offered with quantities sold in a couple of markets, we still have not addressed what holds those prices up. These reactions vary from market to market, but generally, we notice that markets like features that generalize to broad categories. While we could choose from features such as endurance, safety, and flexibility for aircraft, primary attributes such as capacity and speed are often dominant. We might guess that we could use some measures of each to predict the sustainable Price of business jets, and we do that in Figure 4, where we characterize capacity as Cabin Volume in cubic feet (shortened to Cab Vol) and portray our speed component as Maximum Cruise Speed in Miles Per Hour (or Max MPH). Both features combine to produce points in 3D space that reflect speed and capacity on the horizontal axes and Price on the vertical axis.

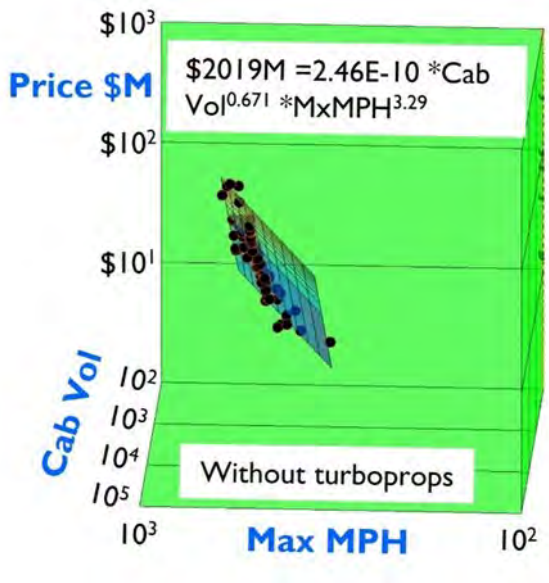


Figure 4 – Business Aircraft Value comes from Speed and Cabin Volume

While the data in Figure 4 is very well correlated, that fact is not as important to us now as seeing what Figure 4 has in common with Figure 1. Sure, both figures address the business aircraft market over the same time frame, but something else should catch our eye. That is, both illustrations share the Price axis. This means that they abut one another in a 4D arrangement consisting of ordered quads, which here form as (Max MPH, Cabin Vol, Price, Quantity) and which generalizes to (Valued Feature 1, Valued Feature 2, Price, Quantity). We see how this complete arrangement looks in Figure 5

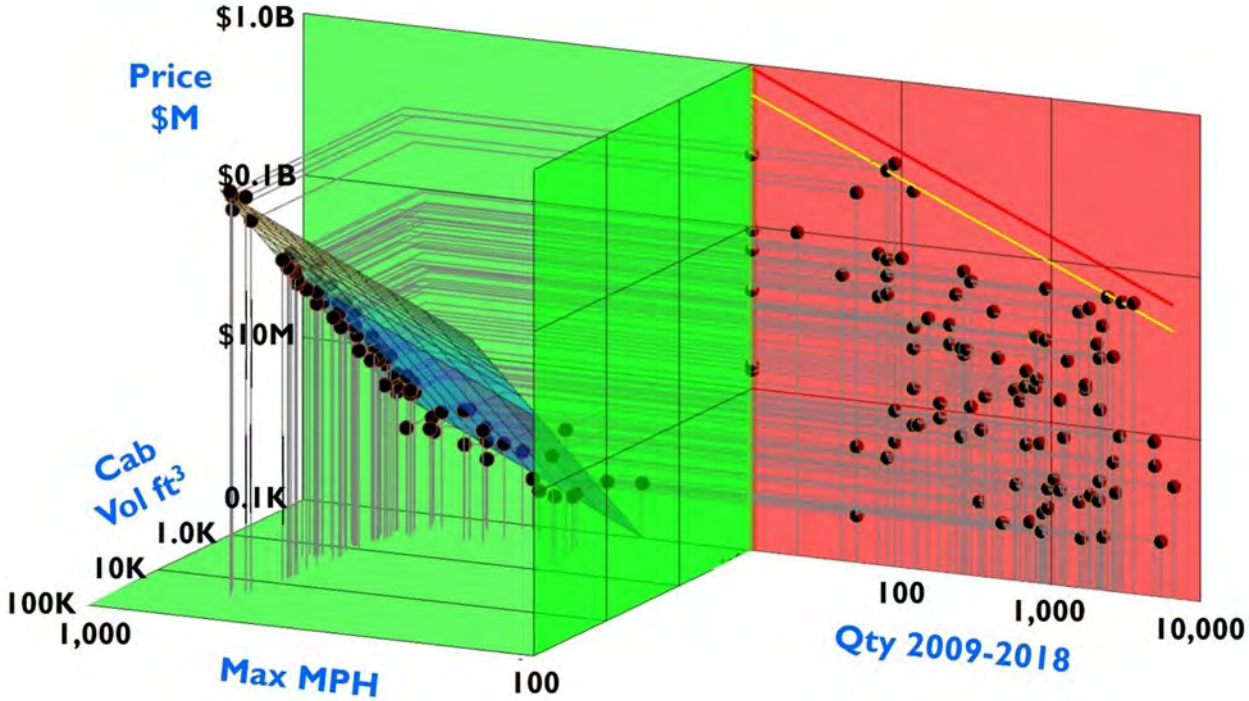


Figure 5 – A 4D view of the business aircraft market, where all points in the left-hand Value Space connect to their right-hand counterparts on the Demand Plane as point-lines

Given that we have found one market we can describe in four dimensions, we might reasonably wonder if we could find more. With just a bit of research, we discover we can.

What is a 7D System?

We can create a 7D system if we start with one 4D system and add another. We can do this because all 4D market systems share the price axis. So, understanding that lets us draw the 4D market for the turbofan engines that go into the business jets we just studied, as shown in Figure 6.

There, we find that the Value of turbofan engines goes up with added thrust but down with increased Specific Fuel Consumption (Specific Fuel Consumption, or SFC, is akin to the miles per gallon of range we measure in cars with internal combustion engines). Demand in Figure 6 has a flatter slope for its Upper Demand Frontier, and no discernible Outer Demand Frontier appears. Observe that we have also changed the scaling from the log form we used for business jets to linear versions for all axes.

Figure 6 shows us that 4D coordinate systems are not limited to business aircraft, as we can apply the same techniques we used for aircraft to their engines. Now, if we were to change the scaling in Figure 6 and match it to what we used for Figure 5 and then recognize that both planes and engines share the price axis, we could get the 7D view we see in Figure 7

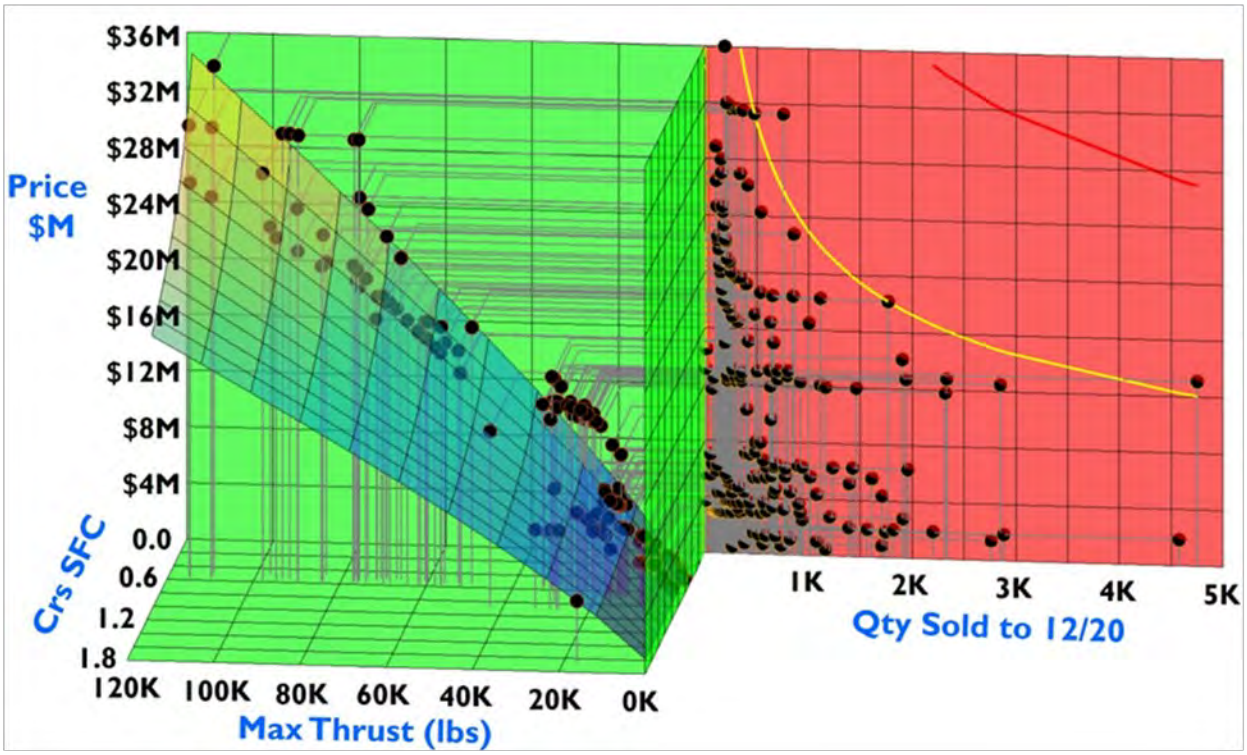


Figure 6 – Here's a 4D view of the turbofan engine market. Value in engines goes up for added thrust and down for increased SFC. As with Business Jets, all points in the left-hand Value Space connect to their counterparts on the Demand Plane as point lines.

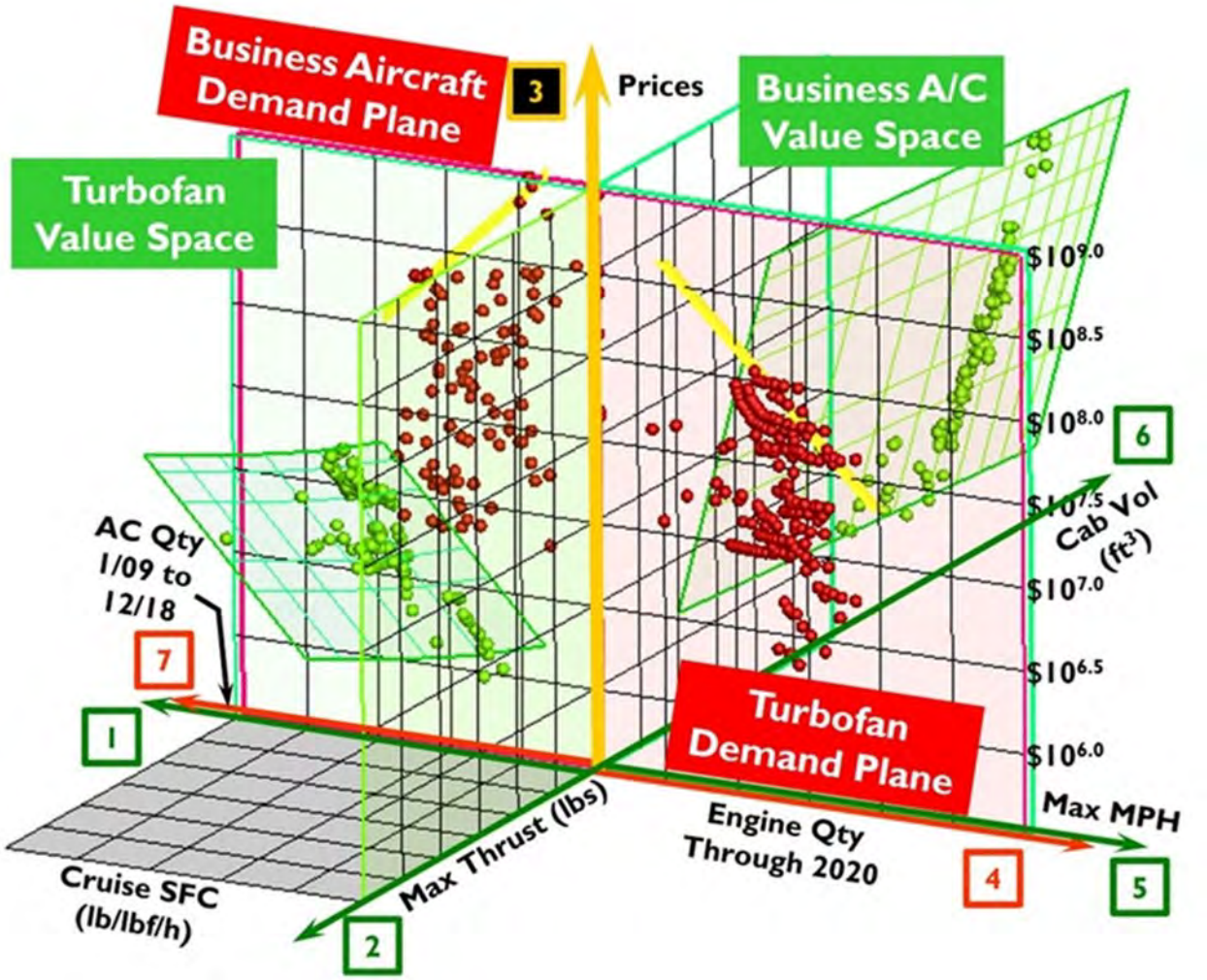


Figure 7 – Since the 4D business jet market and the 4D turbofan engine market share the price axis, when it comes to dimensions, 4+4=7. In this view, we have the turbofans in front, with Cruise SFC (Dimension 1) and Max Thrust (Dimension 2) for Valued Attributes driving Price (Dimension 3), while engine Prices limit their Quantity Sold (Dimension 4). At the same time, in the back portion of the diagram, Max Miles Per Hour (Dimension 5) and Cabin Volume (Dimension 6) push business Aircraft Prices (Dimension 3, as with the engines), with their Quantity Sold (Dimension 7) limited by prices and market saturation

How can we get to 13D?

Since we've managed to wrangle a 7D system that we could never see before, we might wonder what other techniques we might employ to expand our view further. Some seemingly simple observations can help lend insight into this problem we've set before ourselves. Let us entertain the swinging panel display we see in Figure 8. Such devices have been around for decades at least, and while they may appear simple, they have some unique ways of storing information.

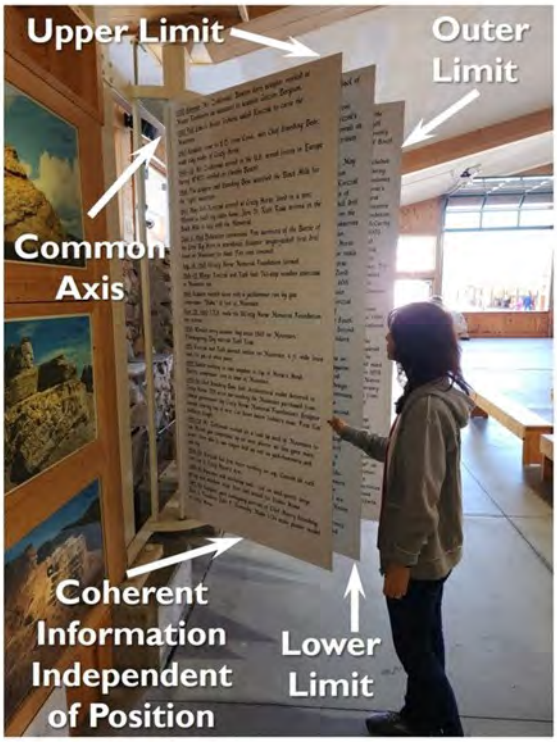


Figure 8 – Posters on a poster rack not only have upper, lower, and outer limits, but they all have a common axis, and the information on each card remains coherent regardless of position.

As Figure 8 displays, each card limits itself to its upper, lower, and outer boundaries. Crucially, in addition to those properties, each card works off the same vertical axis, and all of them retain all their data no matter what angle they take concerning the other cards. Given these observations, let's examine Figure 9, a simple 4D table of three general aviation aircraft models.

Once we have the information in Figure 9, we have enough data to craft a standard 4D model, as shown in Figure 10. While Figure 10 looks familiar and useful enough, we recall what we discovered in Figure 8. Specifically, we found that the data on a poster held its coherence regardless of its position on the central vertical axis. Knowing that, we can rotate the Demand Plane 15˚ clockwise in Figure 11 without losing information. Armed with that observation, we continue to rotate the Demand Plane to 30˚, 60˚, and 75˚, as depicted by Figures 12, 13, and 14, respectively. Finally, in Figure 15, we rotated the Demand 90˚, which lies flat against its respective Value Space. In the process, we've compressed the space needed to display a 4D system from 180˚ of arc to 90˚.

Model	Seats	Max MPH	$M (2005)	Quan-tity
SR22	4	212	$0.29	1705
Baron 58	6	232	$1.00	330
Caravan I	8	213	$1.50	504

Figure 9 – Seats and Max MPH contribute to the Price of these General Aviation aircraft models and restrict their quantities sold

Since we successfully swung the Demand Plane onto Value Space, reducing the degrees of arc needed to portray a market,we might wonder if we can compress our data storage further.

With Figure 16, we begin to study what we can do based on the six figures previous to it. In this Figure, we plot some whole number ordered pairs, which we label as indicated. Every pairing matches our long-standing sense of how we account for position in this realm.

But, in Figure 17, we adjust the angle between the traditional X and Y axes from 90˚ to 75˚. Importantly, we can still make sense of the location of all the ordered pairs with which we began as the angles between them form parallelograms, with their "vertical" distance to the origin remaining constant and their "horizontal" component relative to their neighbors unchanged as well. These are Polar Parallel Coordinates, using pantographs like Thomas Jefferson did with his Portable Polygraph, built by John Hawkins.[7] Figures 18 and 19 confirm we continue to track those numbers as the subtended axes collapse.

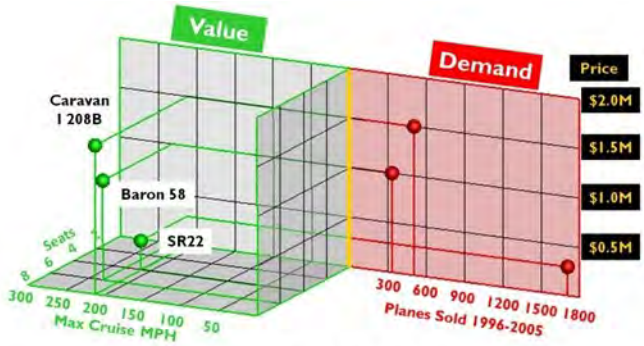


Figure 10 – A standard 4D model uses planes which are orthogonal to one another

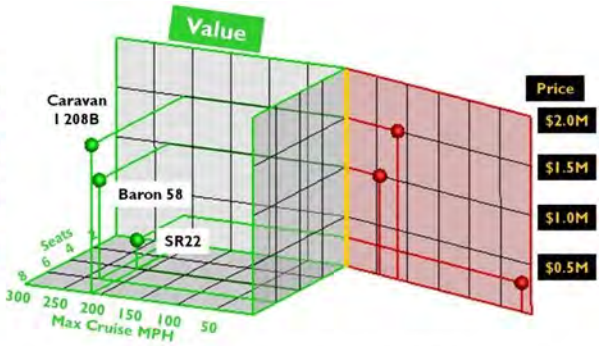


Figure 11 – The data in that model is not lost as we rotate the Demand Plane 15°

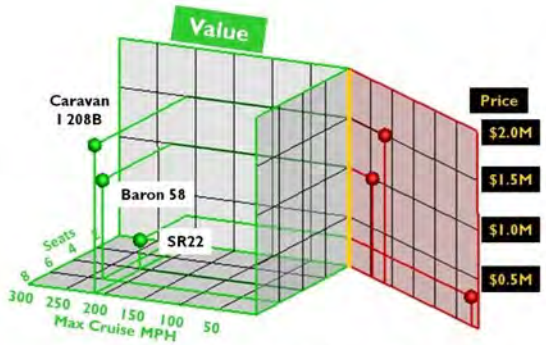


Figure 12 – The data stays coherent as we rotate the Demand Plane 30°

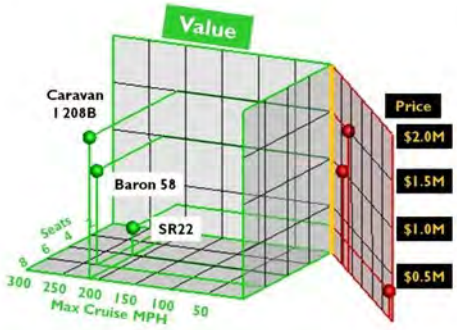


Figure 13 – No information loss occurs as we rotate the Demand Plane 60°

Figure 14 – We keep our information with the Demand Plane rotated 75°

Figure 15 –The Demand Plane lies flat against Value Space when rotated 90°

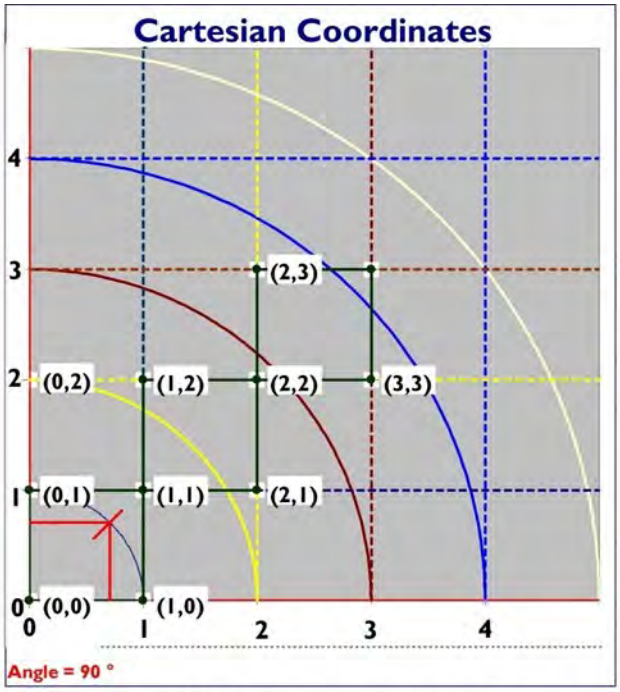


Figure 16 – Standard ordered pairs in standard orthogonal coordinates

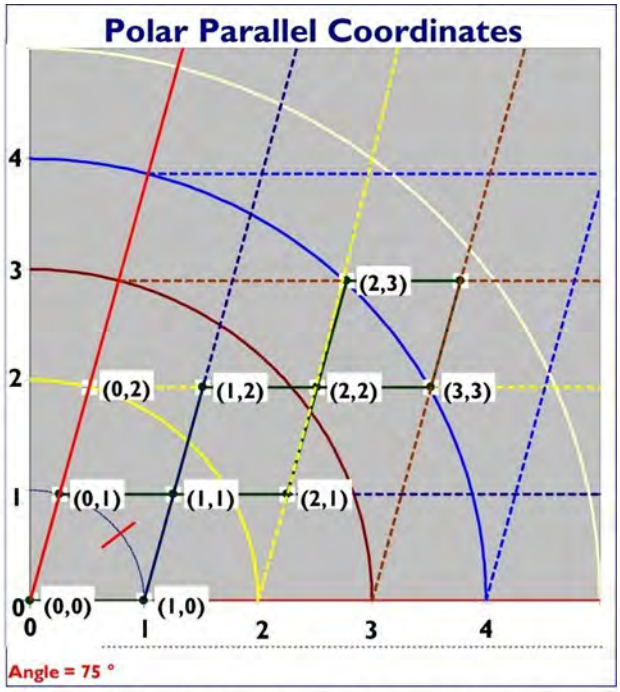


Figure 17 – Ordered pair data is not lost as we use parallelograms, condensing the view

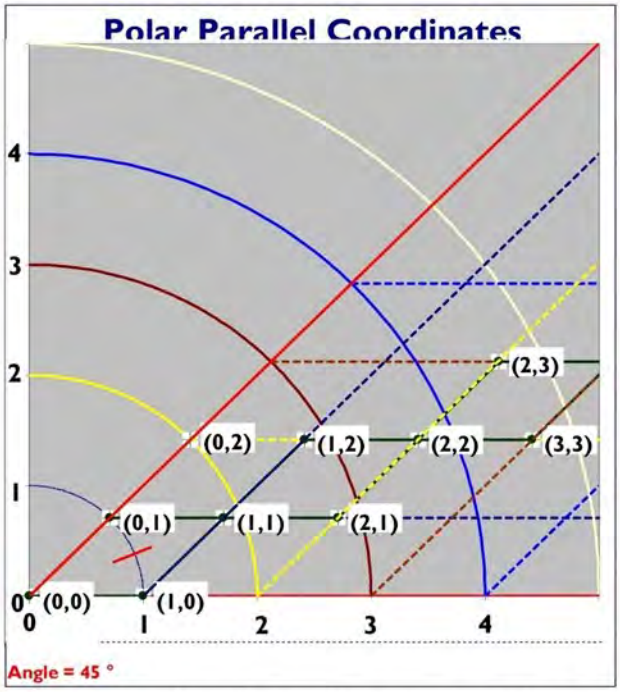


Figure 18 – The information remains intact with further data compression

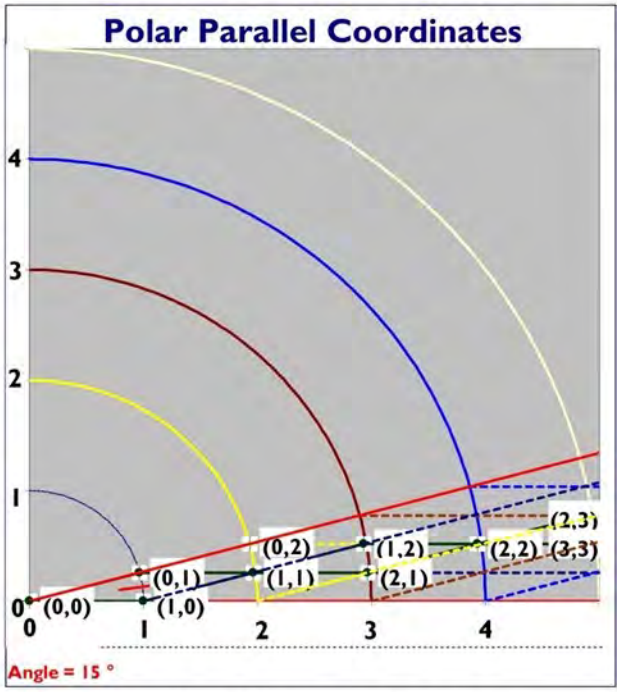


Figure 19 – The information is clearly infinitely compressible

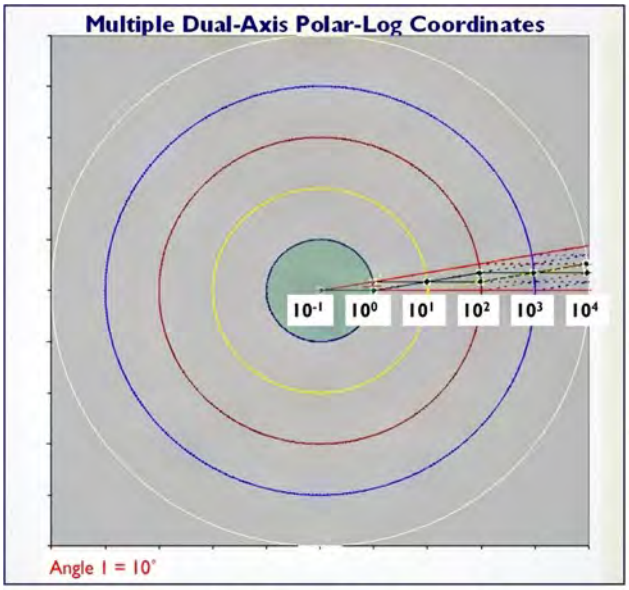


Figure 20 – The data we put into a 90° environment can be placed into a 360° view

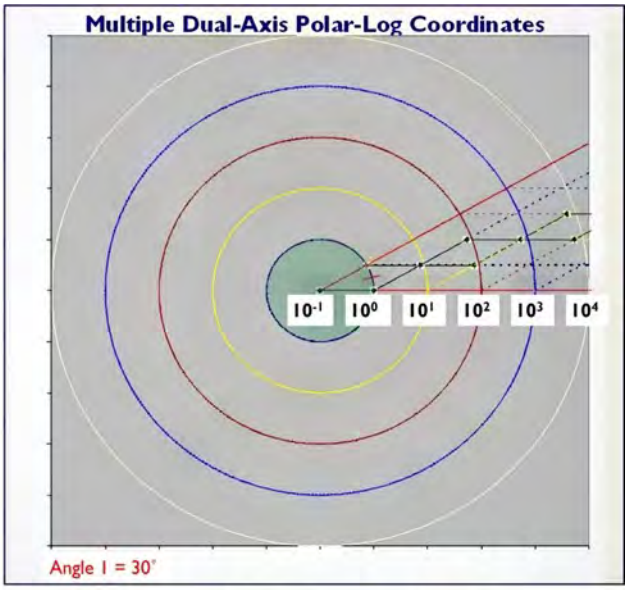


Figure 21 – As before, we can show any market with the appropriate angle

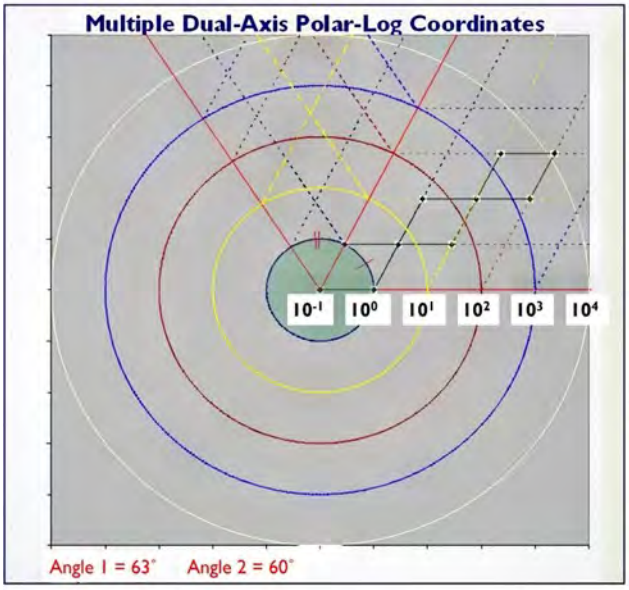


Figure 22 – We can add another market if we need to expand our analysis

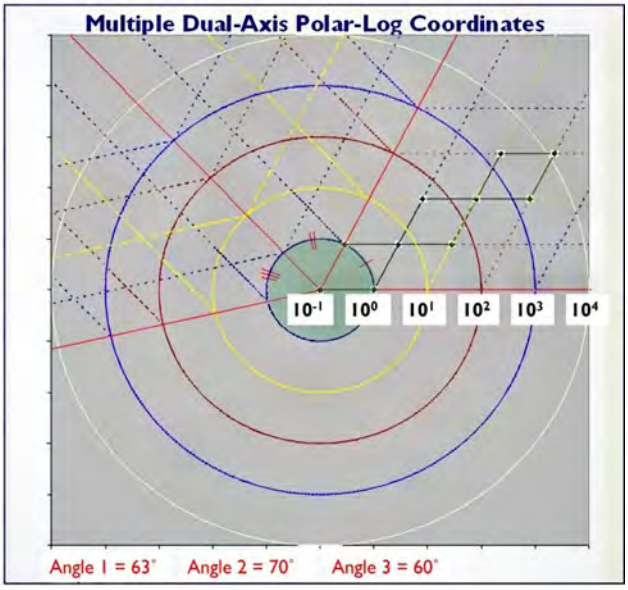


Figure 23 – We can keep adding as many markets as we would like to see

While Figures 16-19 examine the dimensional collapse across what amounts to one quadrant, or 90˚, of arc, Figures 20-23 study the effects of expanding the analysis to 360˚ and adding other markets. Figures 20 and 21 remind us of what we saw in the four figures previous to them, excepting only that they now appear across an entire 360˚ circle. Crucially, in Figure 22, we find we can easily add another market of a certain angle, while in Figure 23, yet one more market appears, and we have a full accounting of it. So, if we wanted to portray more markets, how would we go about it? Clearly, we just demonstrated that we could add as many markets as we care to. How might we apply these concepts to World GDP?

First, let's entertain how World GDP is broken down into three primary categories, according to the United States Central Intelligence Agency (CIA), as they envisioned it in 2014 in Figure 24.

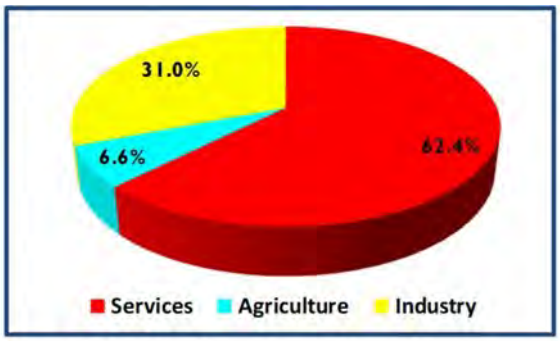


Figure 24 – The United States Central Intelligence Agency estimated 2014 World GDP to be $78.28 trillion, broken into three categories [8]

While Figure 24 is highly informative, given our recent work, we wonder if we could add some additional insight into what the CIA has offered us. We might start by depicting GDP as a cylinder, as in Figure 25.

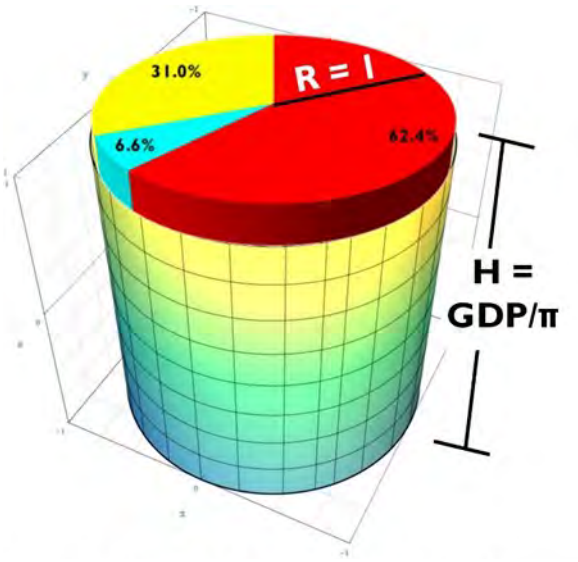


Figure 25 – We can envision GDP forming a tall cylinder, of height GDP/π

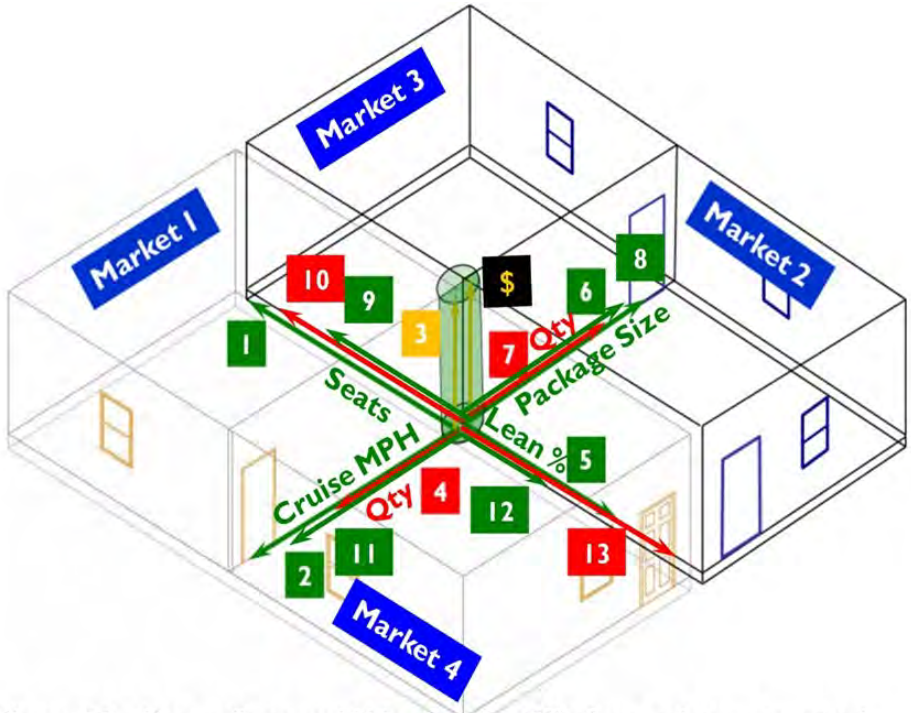


Figure 26 – If we collapse each Value Space to 90° of arc, and swing its paired Demand Plane fat against it, we can produce this system in which we portray four markets simultaneously.

Markets	Value Dimensions	Quantity Dimensions	Currency Dimension	Total Dimensions
1	2	1	1	4
2	4	2	1	7
3	6	3	1	10
4	8	4	1	13
n	2n	n	1	3n + 1

Figure 27 – Every market has two primary Value Dimensions and one Quantity Dimension. All markets share the Currency (or Price) Dimension. Thus to plot n markets, we need 3n+1 dimensions

Now, Figure 10-15 taught us that we could reduce four markets simultaneously, each requiring 90˚ of arc. We also learned in Figure 25 that we could portray GDP as a tall cylinder, which, if it had a radius of 1, would have a height of GDP/π. Now, if we combine both of these insights and take the log of the GDP cylinder, we get the view we see in Figure 26. Each of the four markets has its Value axes at 90˚ to each other and has its Demand Plane lying flat against the Value Space. Here, Market 1 is an extension of the data used in Figures 10-15, where we studied General Aviation aircraft. Market 2 considers ground beef, where package size and leanness are a pair of primary Value considerations and where it has its Demand Plane lying flat against the Value Space. Markets 3 and 4 (which could be anything) behave the same way. When we study these markets collectively, we notice a pattern forming, which we display in Figure 27.

It recognizes that the first market has a pair of Value dimensions, a single Quantity dimension, and one price dimension for four dimensions. The second market also has an identical number of Value and Demand dimensions but shares the Price axis with the first market. That pattern repeats for the third and fourth markets, as every nth market has two Value dimensions and one Quantity dimension but shares the price axis with all other markets. Thus, Figure 27 observes that to plot any number of n markets, we need 3n+1 dimensions.

Figure 28 shows how to plot a multidimensional picture, showing the single market for commercial aircraft in 2014. There, the green central cylinder represents the world GDP, using the logs of the conventions described in Figures 24 and 25, with a radius of 1 and a height of $10^{13.4}$. Note that the circular rings running parallel to the base go outward in multiples of ten to a maximum of one trillion (10^{12}). Splitting the data up according to the GDP taken by each market begins with the "Zero Angle" plane, shown with a red outline, which separates manufacturing (the pie section from red to orange) from agriculture (that slice going from red to blue). The balance of the GDP goes to services, marked by the largest section that extends from the orange divider to the blue one.

By convention, each market has three planes that describe it. All of them share a common upper border, which is the most expensive item in that market in that year, while the lower border is the least costly good or service offered for sale in that market. In the case of Figure 28, the upper limit was $414M, while the lower limit was $129K. The lateral extent of each plane depicts the maximum extent of the sales (as quantities, here, 5063 for one year) or features offered for sale (with the maximum number of seats at 525 and Miles Per Hour at 617). With the Demand Plane hard against one of the Value Space Planes, the angle between the Value Planes offers that market's portion of the World GDP. In the condition at hand, this market accounts for a little over 1˚ of arc or roughly 0.31% of the world's output.

What's a 5-Market 16D System?

What we can imagine now, given the techniques we've adapted to our purposes, is that we can plot any number of markets at the same time. In Figure 29, employing the methodologies we just discovered, we rotate the Demand Planes onto their appropriately condensed Value Spaces for five markets and their associated 16 mathematical dimensions. This Figure consists of two Valued Feature axes for each market, one Quantity axis for each market, and a single currency axis common to all markets. In addition to the Currency Dimension (Dimension 1) used for all markets, we include those from Figure 28 that we used for Aircraft (Dimensions 2, 3, and 4). Rapid shipping services employ Dimensions 5, 6, and 7, while we describe the market for ground beef in the United States using Dimensions 8, 9, and 10. For personal transportation, we entertain Dimensions 11, 12, and 13 for cars with Internal Combustion Engines (ICE) and Dimensions 14, 15, and 16 for electric cars.

Observe that the Internal Combustion Engines (ICE) car market forms the largest market in this study, with 2.35% of world GDP, over 347 times that for electric cars.

As we can see, there is no limit to the number of markets we can characterize in this fashion.

Importantly, suppose we reflect on markets dating back to the beginnings of humanity and compare them to what we see in Figure 29. In that case, we can imagine some dramatic changes in what people buy and sell now versus what they did long ago.

To make our reflection more complete, we'll focus on a market that has been around for as long as we have, and take it from a long time ago to the present, stopping at various points along the way. In so doing, we'll see how that market evolved as we did.

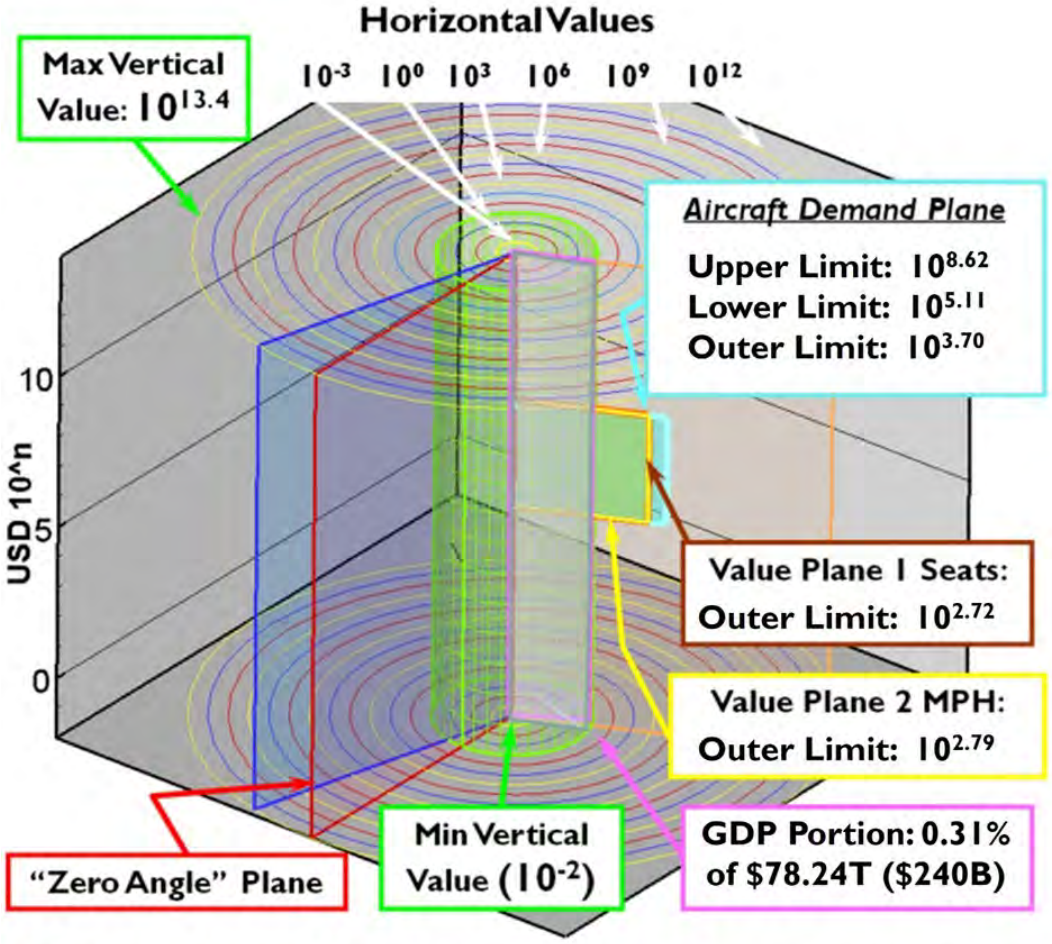


Figure 28 With the "Zero Angle" red plane as a starting point, we go counterclockwise to the orange, blue, and back to the red planes, thus marking industry, services, and agriculture in that order. Note that the concentric horizontal values begin at 10^{-4} and go in powers of 10 to 10^{12}, which is our arbitrary outer limit. Note the inner green circle with a radius of 1 (10^0), which is our GDP pie column. The pink plane that reaches from the center horizontally out to 1, and up from 10^{-2} (our bottom) to $10^{13.4}$ forms a wedge with the orange plane. This wedge is the portion of GDP (0.31%) dedicated to commercial aircraft sales. We mark this market's demand in the light blue plane, with the bottom of it representing the lowest priced vehicle in the market ($129,000 or $10^{5.11}$), while the upper limit is for the most expensive vehicle ($414 million, or $10^{8.62}$), while the lateral extent of this plane describes yearly sales (5063, or $10^{3.7}$). We keep Value Planes with the same upper and lower limits, while the lateral extents of Value Planes 1 and 2 indicate the maximum number of seats (525 or $10^{2.72}$) and miles per hour (617 or $10^{2.79}$), respectively. The angle between the aircraft value planes is 1.11° or 0.31% of the global GDP. This market is a little more than 1/300th of GDP.

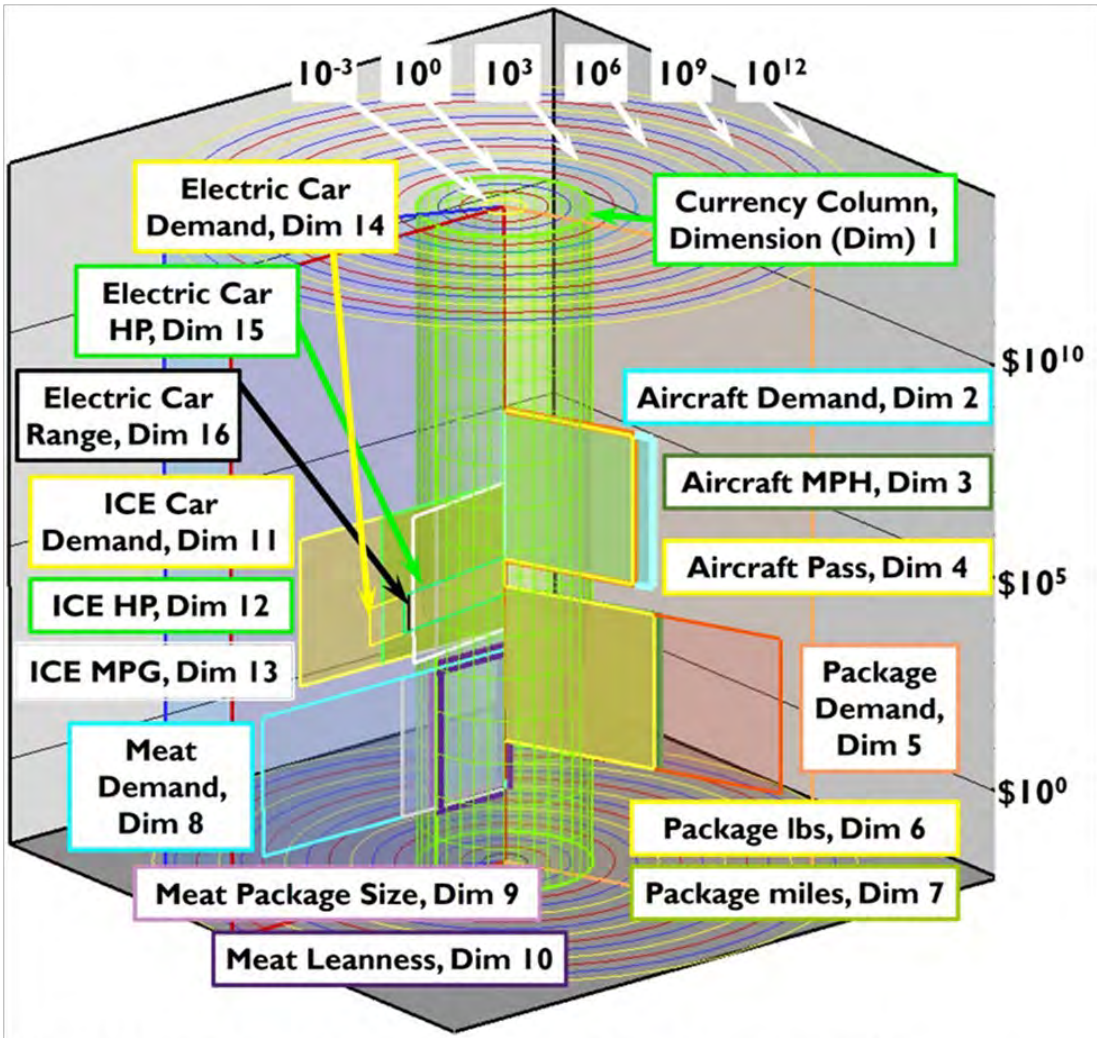


Figure 29 - Using some of the recent techniques that we just learned, we swing the Demand Planes onto compressed Value Spaces for five markets and employ 16 mathematical dimensions. This consists of two Valued Feature axes for each market, one Quantity axis for each market, along with a single currency axis common to all markets. In theory, there is no limit to the number of markets that we characterize in this fashion. Note that the market for cars with Internal Combustion Engines (ICE) forms the largest market in this study, with 2.35% of world GDP, over 347 times that for electric cars.

A Partial History of Surveillance

Surveillance predates mankind. How can I make such a statement? It's not as if we have records of what happened hundreds of thousands or millions of years ago. What do we have to make that assertion?

A few months ago, I was at the end of a trailhead after a long run. Standing in the parking lot, bent over and struggling to catch my breath, an ant caught my eye.

Ground Level Surveillance

Now ants had famously been interesting to Richard Feynman, the Nobel Prize-winning physicist.[10] He used to see how long it would take to find some sugar he put out for them. So, I began to study this ant. I thought, "If it is worth Dr. Feynman's time, why can't I spend a minute doing this?"

By jagged fits and starts, this tiny ant, something less than a quarter inch in length, began turning in a

counterclockwise motion, slowly bending to its left in what eventually appeared to be a circle, except that after turning a complete 360˚, the endpoint was outside where it started. With each pause, it appeared on a slightly higher rock than before. It made another counterclockwise ring and then one more, forming a broadening spiral. Then it hit me: That ant was doing reconnaissance.

I rushed home and put "ant reconnaissance" into a Google search bar. A short search revealed that ants who nest in flat rock crevices perform surveillance while looking for new nests, seeking to avoid competing colonies and weighing out features such as floor space, headroom, and cleanliness of their sites.[11] More research revealed that this genus, Temnothorax, has been on Earth for 140-168 million years, far predating humanity.[12]

Thus, with humans being smarter than ants, we can conclude that people have been doing surveillance since their emergence. While we don't have any records of these activities before the historical record began, we do have examples of how we performed surveillance in the past.[12]

Fixed Elevated Surveillance

The ant upon which I had fixated used at least two surveillance methods I could observe. The widening circles gave it a broader area of coverage. And, by pausing on elevated rocks millimeters higher, the ant could increase its distance to its horizon. Both techniques offered increased information about the surrounding area.

Humans evolved more ways to gain data about their adversaries.

The watch tower Vartovka (see Figure 30), erected to warn the citizens of nearby Krupina, Slovakia, of invading armies, was built in the late 1500s. Its elevation offered a longer distance to its field of view and, when joined by several other like structures, gave the town advanced early warning of armies invading from multiple directions.

While it and others like it constructed atop the surrounding hills served the town well, it placed a heavy financial burden on the local townspeople. Such structures are still being built for prisons and spotting fires, but armies needed more flexibility.

What if the enemy is on the move or far away from the city, needing protection? Heads of state required other methods to gain insight into their battlefields.

Figure 30 – The Vartovka watchtower, (with a modern staircase and reconstructed porch added to it), was paired with similar sentry posts to offer advance warning of approaching armies

Mobile Aerial Surveillance

Brothers Joseph and Etienne Montgolfier in Annonay, France, created the first lighter-than-air balloon in 1783. After carrying animals on their first flight, they quickly transitioned to carrying people in the same year, first using tethers and then graduating to free flights.[13] The French military was quick to appreciate its surveillance potential.

The French were the first to use balloons by military forces in 1794, and they were used irregularly for reconnaissance in the French Revolutionary Wars.[14] They appreciated the portability. Now, anywhere a battle arose, they could enjoy views that previously would have taken an intensive tower construction for a fraction of the cost.

Nearly 70 years later, the United States was fighting the Civil War. As in all conflicts, each side wanted

Figure 31 – The Union balloon Intrepid *had a capacity of 32,000 cubic feet of lifting gas (Hydrogen). Hydrogen-generating wagons supplied it. Thaddeus S. C. Lowe, one of the most famous balloonists at the start of the Civil War, built these wagons, which reacted sulfuric acid with iron filings to produce hydrogen gas to fill the* Intrepid..

to know what the other was doing, and both decided that using balloons could aid in doing reconnaissance. We can see how the Union went about the business of filling one of these balloons in Figure 31.[15] [16]

Perhaps their most famous use in that war was when Thaddeus S.C. Lowe went on a tethered flight in one of his balloons on July 24, 1861, three days after the Battle of Bull Run, and determined the Confederate troops were not on their way to Washington DC. This observation was helpful information, and tethered balloons became a part of modern reconnaissance during the Civil War. Their large size and relatively fixed positions became problematic, though, as they afforded large, slow-moving targets.

As time passed, motorized balloons became important in World War I, either non-rigid (blimps) or rigid (Zeppelins, for example). However, given their large size and relative lack of mobility, they were often easy targets for enemy fire and were retired mainly in favor of smaller, more agile platforms.

Those less bulky and faster machines were, of course, airplanes.

Faster, Higher Aerial Surveillance

In World War I, slightly more than a decade removed from the Wright Brothers' first flight, not entirely satisfied with the information they could get from the blimps, whose low ceilings, slow speeds, and large sizes made them excellent targets, the Allied Powers worked hard to get planes that flew higher to retrieve the intelligence they each needed.

One workhorse of the conflict for the Allies was the Caudron G.3. Over 2800 copies were built in World War 1, the vast majority coming from France. While it went only slightly faster than the best lighter-than-air aircraft of the day (the G. 3's top speed was 66 miles per hour, and some German Zeppelins could hit 54 to 60 miles per hour), its ceiling was quite a bit higher (the G.3 could reach 14,100 feet altitude, most German Zeppelins but early in the war had ceilings less than 10,000 feet).[17] [18]

While the Caudron G.3 technological improvement offered the Allies an edge early in the conflict, the plane's limitations became evident as the war dragged on, as shown in Figure 32. At right, getting too close to the action nearly cost the pilot his life. Going higher and faster seemed like a better option

Figure 32 – On the left is a pilot of the Allied Powers in a Caudron G.3, used for training and reconnaissance in World War I. Note the metallic aft fuselage to the left of its serial number 6366. At right, the same model after a mission piloted by Everett Howarth (my grandfather). Note the torn-off aft fuselage, hit by friendly fire. Reconnaissance missions are uniquely dangerous, and there has been an ongoing movement to perform surveillance using better and safer methods and platforms since surveillance began.

Ever Faster & Higher Surveillance

The hard-won lessons about reconnaissance plane losses from World Wars I and II were not lost in the intelligence community. With anti-aircraft batteries improving for the Soviet Bloc, as Cold War tensions brewed, it became imperative to be able to outfly ground-launched attacks on spy planes.

Lockheed (before it was Lockheed Martin) came up with a pair of solutions, the U-2 and SR-71 (Figure 33).[19][20]

First came the U-2, a single-engine spy plane that could fly higher at altitudes exceeding 80,000 feet, as it cruised at a rather pedestrian 470 miles per hour. As it entered operational service, that service ceiling exceeded the ability of any opponent's missiles to reach it. But then, after years of enjoying no opposition to its flights, a U-2 was shot down over the Soviet Union.

That incident called for another innovation from Lockheed, the SR-71. Flying only slightly higher than the U-2 at 85,000, its primary advantage was its top speed, which, at 2200 miles per hour, was nearly five times faster than the U-2, exceeding the ability of any surface-to-air missile to reach it. It also introduced the world to the first revealed attempts at reducing the radar cross sections of the plane as one of the first implementations of stealth. No SR-71 was ever lost to hostile actions. However, it was not immune to budget issues, and in 1989, it was retired due to the high cost of its maintenance.

Figure 33 – The U-2 spy plane on the left flew higher than anything else in the sky when it was first conceived and flown. It could fly with impunity over Eastern Bloc territory for years. Then, as often occurs in this arena, opponents to US interests caught up, as the Russians designed and built a missile that took down a U-2 over Soviet Territory. That incident force the development of the SR-71 (right), which flies at 85K+ feet and travels over 2200 MPH. While it was never shot down, its high price forced the USAF to discontinue its use.

Unmanned Aerial Surveillance

Our Figure 2 revealed that a large portion of modern aerial surveillance has migrated away from planes with pilots in the cockpit to Unmanned Air Vehicles (UAVs) and Satellites. As we found out, these disparate product forms perform the same missions and end up having the same Minimum Demand and Outer Demand Frontiers. We might guess that customers for both types of platforms respond to the same types of features offered by both groups, and we would be right, as we see in Figure 34.

The beneficial and perhaps unexpected outcome of combining UAVs with satellites is that the features that predict their Values and the markets' response to them boil down to one single equation.

That equation is

$$\$M = 0.0290 * PL\ lbs^{0.766} * Max\ MPH^{0.301} * Alt\ Miles^{0.169} * Qty^{-0.283} \quad (1)$$

Where:

$M = Estimated Price per unit in 2021$M
PL lbs = Payload pounds
Max MPH = Maximum Speed in Miles Per Hour
Alt Miles = Maximum Altitude in Miles
Qty = Total Quantity Sold through 2021

Equation 1 has an Adjusted R^2 of 96.1%, a Mean Absolute Percentage Error (MAPE) of 71.3%, an overall p-value of 8.83E-39, and p-values of 3.16E-15, 1.53%, 0.42% and 0.13% for PL lbs, Max MPH, Alt Miles, and Quantity sold, respectively.

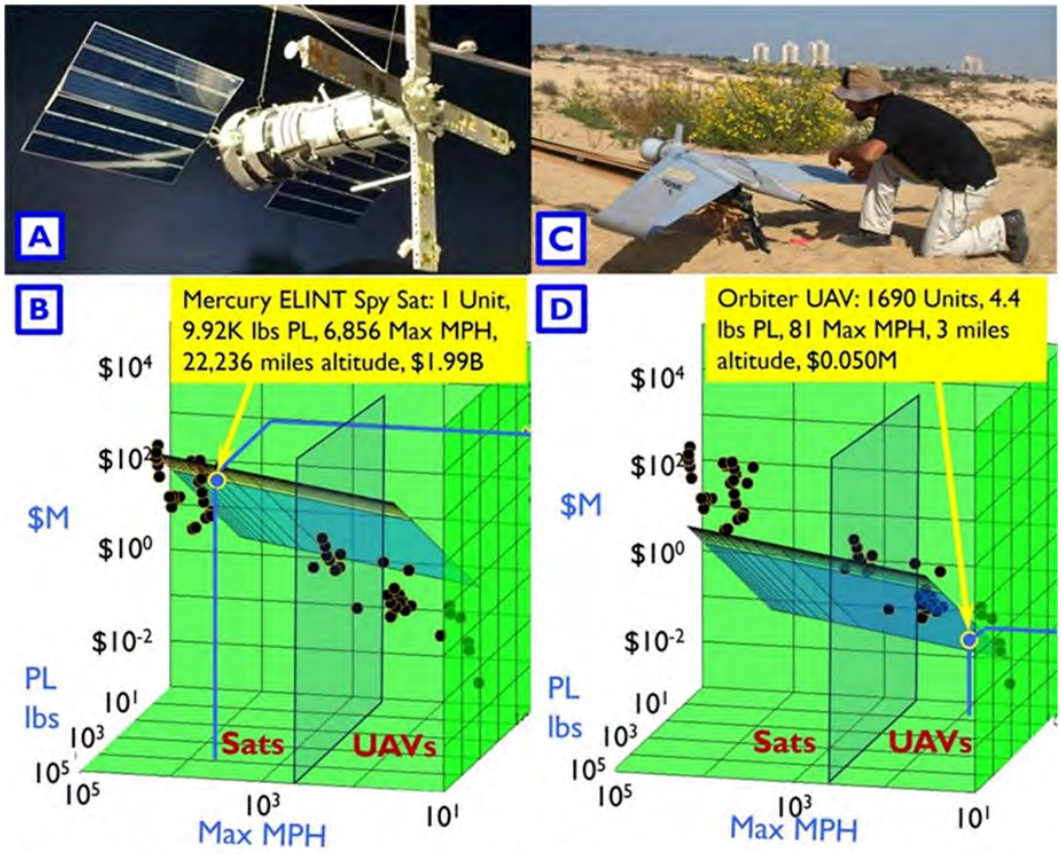


Figure 34 – The Russians run ELectronic INTelligence (ELINT) satellites over the US (A). Western forces must get the same insight to be as well-informed. Using the same equation, we can predict the Value (sustainable price) of Western Bloc satellites and UAVs. In (B), with 30 Sats on the left and 30 UAVs on the right, we predict the Value of the US Mercury ELINT satellite using an equation considering 1) Payload, 2) Max MPH, 3) Altitude, and 4) Quantity sold. We highlight the Israeli Orbiter UAV (C) estimate using the same equation in (D). With an adjusted R2 of 96.1%, its p-value is 8.83E-39.

When we combine the analysis of the Value of these platforms with their Demand and recurring costs, we can obtain the view we see in Figure 35. Here, we have posited making a new Low Earth Orbiting (LEO) satellite, with an orbit 500 miles above the Earth. We're proposing a 4000-pound payload, and, given the orbit we've chosen, we'll be going about 17,000 miles per hour. If we set our targeted sales to 20, our ultimate average Price will be $290M per unit. Given our assumed Empty Weight, we'll find that the cost of our first unit, or T-1, will be relatively high. To gain all the sales we want for this new platform, we must seek to keep our learning curve for it at 91% or lower.

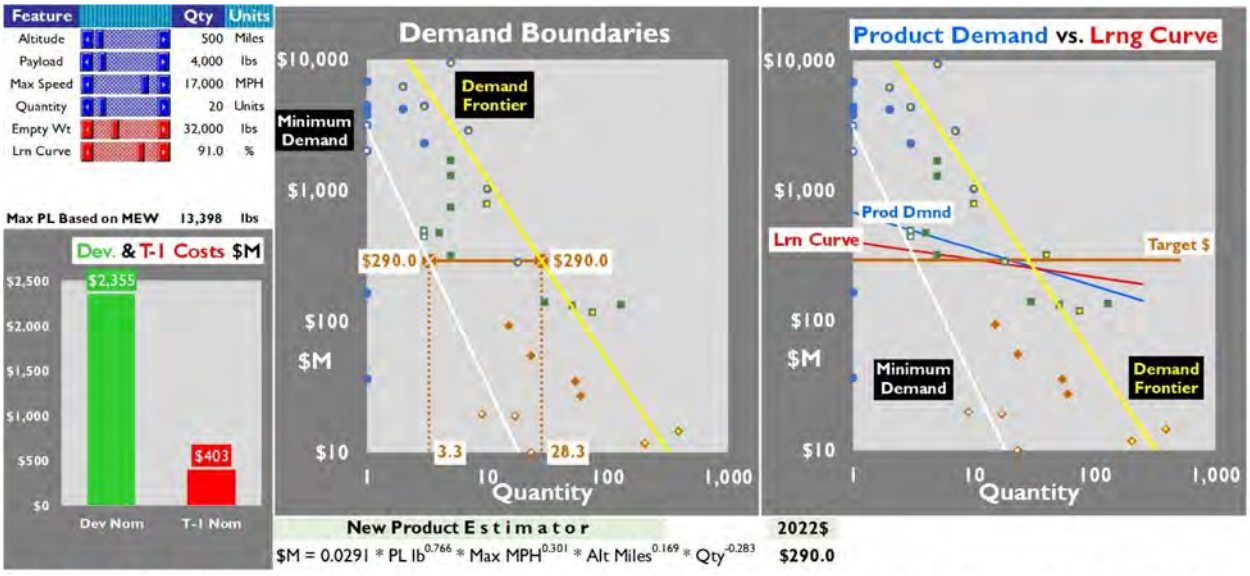


Figure 35 – You want to build an unmanned surveillance platform—what features will you offer? That combination of specifications will determine the product Value (as a sustainable price). Assuming that your new product is viable, that Price sets Minimum and Maximum Demand Limits. Once you know your price and quantity target, you'll need to calculate the T-1 cost which, in conjunction with your Learning Curve, either lets you hit your ultimate sales goal or fall short.

Summary and Conclusions

Some markets come and go. Others change over time. Modern plotting techniques let us portray any market in four dimensions, two in seven, and so on. Understanding how related markets work together is critical to ensuring they will survive. Plotting all markets worldwide simultaneously using a system that amounts to a vertical rolodex with unique properties laid out in advance is possible.

Humans have spent time doing surveillance since they first walked on Earth. While the methods change, the objective has always been to provide more information about our surroundings and potential enemies than we had before. Even ants search for higher ground, and we can quantify what humans will pay for those vantage points. Speed is crucial in many realms, but never more so than in surveillance, where delays in getting information can kill you. Newer platforms may require us to take along specialized gear as payloads, and the more Payload we can carry, the more it is worth to us. On the other hand, as prices rise, we buy less, and when it comes to aerial surveillance, as we buy more, each succeeding unit is worth less to us. Knowing how these forces work with and against each other is crucial if we are to make new products successful.

References

1. Howarth, Douglas K., "8D Cost Trades with Entanglement," *Journal of Cost Analysis and Parametrics*, Volume 11, Issue 1, April 2023, pages 70-88
2. Neale, Rick, "Aerion Supersonic assets may be liquidated by year's end after company shut down," *Florida Today,* September 1, 2021, Retrieved January 8, 2024, from https://www.floridatoday.com/story/news/2021/09/01/aerion-supersonics-assets-sold-off-auction-after-company-shut-down-may/5680350001/
3. https://en.wikipedia.org/wiki/Eclipse_550, Retrieved January 9, 2024
4. https://en.wikipedia.org/wiki/Extra_EA-500, Retrieved January 9, 2024
5. Alderman, Liz, "France's Aerospace Industry to Get $17 Billion in Government Support," *New York Times*, June 9, 2020, Retrieved January 9, 2024, from https://www.nytimes.com/2020/06/09/business/France-aerospace-industry.html
6. Sarsfield, Kate, "Italian government throws Piaggio €700 million lifeline," *Flight Global,* June 24, 2019, Retrieved January 9, 2024, from https://www.flightglobal.com/systems-and-interiors/italian-government-throws-piaggio-700-million-lifeline/133323.article
7. The American Philosophical Society, Portable Polygraph, owned by Thomas Jefferson and made in London circa 1806 by John Isaac Hawkins, from https://www.amphilsoc.org/item-detail/portable-polygraph-owned-thomas-jefferson, Retrieved January 18, 2024
8. Data from CIA World Fact Book, Retrieved January 17, 2016, from https://www.cia.gov/library/publications/the-world-factbook/geos/xx.html.
9. Drawing from Sam Derbyshire, February 21, 2012, Retrieved January 18, 2024, from https://commons.wikimedia.org/wiki/Category:Cylinders#/media/File:Circular_Cylinder_Quadric.png
10. Garnier, Simon, SwarmLab, "Richard Feynman Had a Thing For Ants," August 8, 2015, Retrieved February 8, 2024, from https://www.theswarmlab.com/blog/2015/08/08/Richard-Feynman-had-a-thing-for-ants/
11. Franks, Nigel R., et al., "Reconnaissance and Latent Learning in Ants," Proceedings of the Royal Society of the Biological Sciences," April 10, 2007, https://doi.org/10.1098/rspb.2007.0138, Retrieved February 8, 2024, from https://royalsocietypublishing.org/doi/10.1098/rspb.2007.0138
12. Krupina – Vartovka Watchtower History, Retrieved February 8, 2024, from https://medievalheritage.eu/en/main-page/heritage/slovakia/krupina-vartovka-watchtower/
13. Balloon (aeronautics) article on Wikipedia, Retrieved February 10, 2024, from https://en.wikipedia.org/wiki/Balloon_(aeronautics)
14. National Park Service, Air Balloons in the Civil War, Retrieved February 10, 2024, from https://www.nps.gov/articles/000/air-balloons-in-the-civil-war.htm
15. Brady, Matthew or Brady-Handy, *View of balloon ascension*, circa 1862. Retrieved February 10, 2024, from https://commons.wikimedia.org/wiki/File:Brady_-_Balloon_ascension_HD-SN-99-01887.JPEG
16. Pohl, Robert, History: Lost Capitol Hill: Lowe's Balloon Gas Generators, May 3, 2021, Retrieved February 10, 2024, from https://thehillishome.com/2021/05/lost-capitol-hill-lowes-balloon-gas-generators/
17. Caudron G.3, Wikipedia article, Retrieved February 10, 2024, from https://en.wikipedia.org/wiki/Caudron_G.3

18. List of Zeppelins, Wikipedia article, Retrieved February 10, 2024, from https://en.wikipedia.org/wiki/List_of_Zeppelins
19. Lockheed U-2, Wikipedia article, Retrieved February 10, 2024, from https://en.wikipedia.org/wiki/Lockheed_U-2
20. Lockheed SR-71, Wikipedia article, Retrieved February 10, 2024, from https://en.wikipedia.org/wiki/Lockheed_SR-71_Blackbird\

Doug Howarth*, a Keynote for Chartwell Speakers, discovered Hypernomics, which studies markets in 4+ dimensions. He's published 15 papers and won 5 ICEAA awards, including Best Paper Overall. His company, Hypernomics Inc., worked for NASA and Lockheed Martin. He co-holds US Patent 10,402,838 for the world's first 4D analytic software. Wiley published his book* Hypernomics: Using Hidden Dimensions to Solve Unseen Problems *in 2024, which reached No. 1 in Macroeconomics on Amazon.*

A Markov Model of the Learning Curve

Harry T. Larsen

Background

Historically, the learning curve has been modeled as a power function, hours(unit) = **a** unit$^{\mathbf{b}}$, with **a,** the hours for the first unit**,** and $2^{\mathbf{b}}$ its slope. A least squares regression of its logarithmic transformation, ln(hours) = ln(**a**) + **b** ln(**unit**), can be applied to calculate its parameters, **a** and **b**. A database of $2^{\mathbf{b}}$ parameters summarizing learning curve slopes is often used as a basis to predict the learning curve of a new project. With a choice of slope and an independently derived number one estimate, hours per unit can then be predicted.

Other, more statistically robust approaches are possible. For example, a weight-based Cost Estimating Relationship (CER) can be combined with the learning curve as in hours(unit) = **a** weight$^{\mathbf{c}}$ unit$^{\mathbf{b}}$. The logarithm of this model, when fitted to a set of historical learning curves using ordinary least squares, yields both the parameter estimates and the parameter covariance matrix. Statistically, the error distribution of this transformation is close to Gaussian. The modeling space can be further widened by employing a nonlinear function. Its parameters may be estimated using, for example, Excel's Solver.

Once a project is underway, the labor hours of the first and subsequent units become known sequentially. With this knowledge, the CER's number one uncertainty is replaced by the first unit's actual value. At this point, the estimation problem is to project from the expected value of past units, using an estimated learning curve slope, to determine subsequent unit cost.

To do so, the power function can be transformed into a sequential linear system, and a Kalman Filter applied to it. To illustrate its application, the filter is used to forecast ten 200-unit forecasts over a twenty 1000-unit learning curves. Like a regression application, the filter's parameters are chosen to minimize the sum of squared errors between the filter's predictions and actual learning curve values.

However, unlike a regression, a Kalman Filter produces both an expected value and a variance prediction. Thus, the optimization of the filter's parameters minimizes both the sum of squared differences between the expected and actual values and the predicted and the actual variances.

A Markov Model

The logarithm of the power law, ln(hours) = ln(**a**) + **b** ln(**unit**), can be transformed into a sequential form by taking its derivative, **d(ln(hours)) / d(unit) = b / unit.** From this, we can form the sequential equation: ln(**hours** $_{\mathbf{unit+1}}$) = ln(**hours** $_{\mathbf{unit}}$) + **b(unit) / unit.**

The second derivative of a power function with a negative exponent is positive, **-b /units$^{\mathbf{2}}$**. That is, the slope at unit **t** is steeper than the slope between units **t** and **t**+1. Thus, projecting from unit **t** by the slope **b / t** will understate the next unit's value. Adding a constant to **t** corrects this bias. For a learning curve of 80%, the slope between units **t** and **t**+1 is approximately **b**/(**t** +.46). Thus, the power function reformulated as a sequence can be written as ln (**hours** $_{\mathbf{t+1}}$) = ln(**hours** $_{\mathbf{t}}$) + **b / (t**+.46**)**. This equation, given an initial value of 13 and a slope exponent of -.322, (80%), when iterated over units, produces the straight line in Figure. 1. To illustrate that this formulation retains the flexibility of the power law, a constant added to **t** +.46, creates a Stanford B curve, a formulation that model's a worker's prior experience.

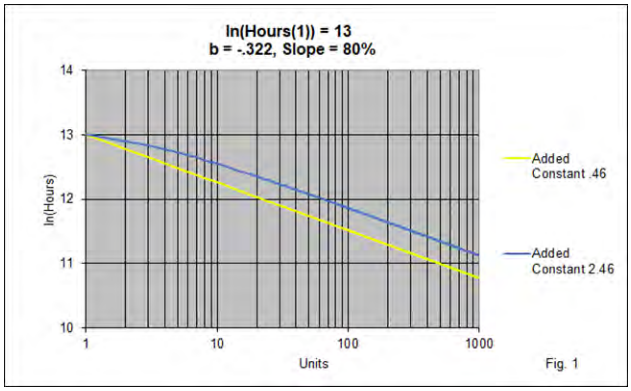


Fig. 1

In practice, learning curves are not straight lines; the slope **b** in **b/t** varies from unit to unit. To account for this, **b** is indexed to unit **t.** With this addition, the sequential equation of the log of the power law becomes ln(**hours** $_{t+1}$) = ln(**hours** $_t$) + **b(t) / (t**+.46**)**, with **b(t+1) = b(t).** This is a linear equation in 1/ **(t**+.46**)** and thus can be written in matrix form as:

$$\begin{pmatrix} \ln(\mathbf{hours}(t+1)) \\ \mathbf{b}(t+1) \end{pmatrix} = \begin{pmatrix} 1, & 1/(t+.46) \\ 0, & 1 \end{pmatrix} \begin{pmatrix} \ln(\mathbf{hours}(t)) \\ \mathbf{b}(t) \end{pmatrix}$$

The Kalman Filter

The Kalman Filter is the optimal estimator in the least-squares sense for a sequential linear system with Gaussian noise. It thus can be applied to the Markov formulation of the learning curve estimation problem. The filter models a two-step process: the transition of the system's states, in this case from t to t+1, followed by the update of those states with an observation.

The Kalman Filter's state transition equation is
X(t+1|t) = A(t) X(t|t). eq (1)

With **X(t+1|t)** read as X at **t+1** conditioned upon data through **t.**

This state transition equation is linear at each **t** to **t**+1 step, with **A(t)** as a function of **t,** changing at each step. Thus, the state transition equation can model the non-linearities of the learning curve's slope dynamics.

For this application, **X(t|t)** is the vector, $\begin{pmatrix} \mathbf{x_1}(t|t) \\ \mathbf{x_2}(t|t) \end{pmatrix}$, where $\mathbf{x_1}(t|t)$ is the expected value of ln(**hours**$_t$) and $\mathbf{x_2}(t|t)$, the expected value of **b(t)**, while **X(t+1|t)** is the corresponding vector at **t+1** conditioned upon observations through **t**.

A(t), the state transition matrix, is $\begin{pmatrix} \mathbf{1} & \frac{\mathbf{1}}{t+.46} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$.
Written by element in state notation, the state transition equation is:

$$\begin{pmatrix} \mathbf{x_1}(t+1|t) \\ \mathbf{x_2}(t+1|t) \end{pmatrix} = \begin{pmatrix} 1, & 1/(t+.46) \\ 0, & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x_1}(t|t) \\ \mathbf{x_2}(t|t) \end{pmatrix} \text{ eq (2)}$$

Once the states have been projected forward one unit, the filter models a delay **t** + 1 → **t**, and **t** → **t – 1.** With that delay comes the observation of the projected unit's measured value, its actual value, **Y(t)**.

The update equation is:
X(t|t) = X(t|t-1) + K(t) (Y(t) - C X(t|t-1)), eq (3)

where **K(t)** is the Kalman Gain and **C** is a matrix that extracts from the state vector those elements that correspond to the observation**.** For this model, **C** is (1, 0), and **C X(t|t-1)** is thus the projected **ln(hours)**. To complete its update, the difference term **Y(t) - C X(t|t-1)** is multiplied by the Kalman Gain, with the product added to the projected state. The reader may notice that this is an exponential moving average process, analogous to a Box and Jenkins's MA(1) process.

The Kalman filter models not only the state's expected values but also its covariance. Corresponding to the state transition equation is the covariance transition equation:

Sn(t+1|t) = **A(t) Sn(t|t) A(t)**^{T}**+ Sa(t+1)**. eq (4)

where **Sn(t|t)** is the state's covariance at **t** and **Sn(t+1|t)** is the covariance projected to **t+1**. The state transition is assumed to be imperfectly modeled by the **A(t)** matrix. This error is modeled by introducing additive noise, **Sa(t+1),** as a covariance matrix. The covariance matrix, **Sn**, is projected forward by the pre- and post-multiplication by the state transition matrix.

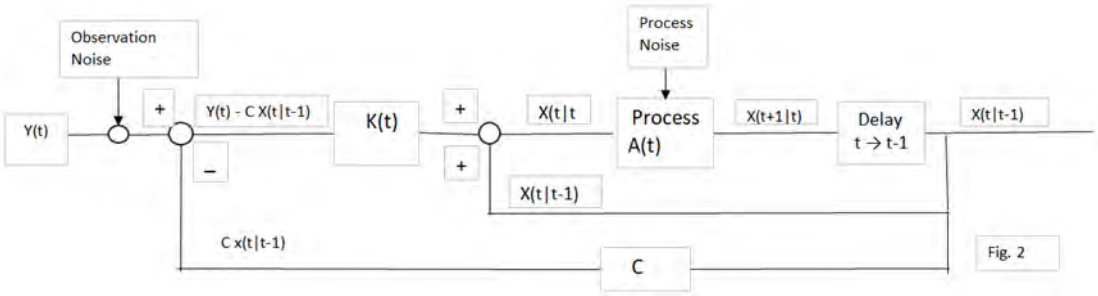
Corresponding to the update of the state vector, the covariance matrix is updated by:

Sn(t+1|t+1) = Sn(t+1|t) - (Sn(t+1|t) C ^{T} **(C Sn(t+1|t) C** ^{T} **+ Mn(t+1)) -1 (Sn(t+1|t) C** ^{T}**)** ^{T} eq (5)

Where **Mn(t+1)** is the observation covariance matrix, in this application, at each step forward, there is one measurement, ln(**hours** $_{unit}$), and thus **Mn(t+1)** is a unitary value. In the term **(C Sn(t|t-1) C** ^{T} **+ Mn(t))** $^{-1}$**,** the pre- and post-multiplication by **C** extracts from **Sn(t|t-1)**, **Sn**$_{(1,1)}$**(t|t-1),** a unitary value. Thus, their sum is invertible via division rather than by a matrix inversion, greatly simplifying the calculation.

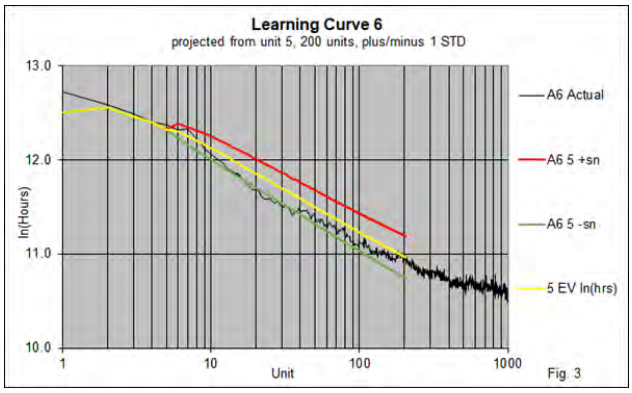
The Kalman gain is: **K(t+1) =**
Sn(t+1|t) C ^{T} **(C Sn(t+1|t) C** ^{T} **+ Mn(t+1)) -1** eq (6)

Figure 2 illustrates the filter steps in a block diagram.



Forecast Example

Figure 3 shows a simulated actual learning curve in black. The Kalman Filter's expected one-unit-ahead estimate, beginning at the #1 CER value, through unit 5, is shown in yellow. The reader will notice that in this case, the expected value, $\mathbf{x_1(1|0)}$ is lower than the actual #1 value, but the filter updates it to a value close to its actual value by unit 2. From the first five actual values, the filter has updated its slope parameter estimate, $\mathbf{x_2(5|5)}$. The yellow line shows its projection 200 units forward. The red and green lines show the one-standard-deviation limits around the projection. As will be shown, they model the actual standard deviations of the projection.



Application Overview

The expected value of $\ln(\text{hours}_1)$ and slope, the vector **X(1|0),** and its covariance matrix, **Sn(1|0),** begin the Kalman Filter sequence. They can be derived from a CER fitted to the first unit and slope exponent of a set of learning curves. The initial covariance matrix is calculated from the variation of actuals around the #1 estimates and the variation in slope around its mean. From the initial values, the state and covariance transition equations (2) and (4) project the values to the next unit. As units are actualized, the update equations (3) and (5) and the Kalman Gain, Eq. (6), are calculated. This sequence is then repeated with the realization of each additional unit's actual value.

For predictions beyond the last actual, there are no observations, or equivalently, the observation variance is infinite. In equation (5), the last term has the expression $\mathbf{(C\ Sn(t|t\text{-}1)\ C^T + Mn(t))^{-1}}$. With a very large observation variance, **Mn(t),** this term approaches zero, and thus equation (5) becomes

Sn(t|t) = Sn(t|t-1). The Kalman gain, Eq. (6), has the same term, and so K(t) also becomes zero for the projected units, and Eq. (6) becomes **X(t|t) = X(t|t-1)**. The transition equations, however, remain intact, and the process covariance continues to be incremented by the additive noise, **Sa**, expanding the process uncertainty, **Sn**, into the future. Figure 3 illustrates that the standard deviation limits expand proportionally to the square root of the number of units projected.

The Kalman Filter differs from ordinary least squares regression (OLS) in that it comprises two types of models: the expected-value model **A(t)**, which corresponds to an OLS model, and the covariance models **Sn**, **Sa**, and **Mn**. In a Kalman Filter application, the **A(t)** matrix is typically derived from the problem's physics; in this case, the derivative of the power function. Additionally, some free parameters are derived from a fit that models the non-linearities of the actual learning curve data. The errors around this model provide the data used to model the covariance matrix. **Sn** is calculated sequentially using Eq. (4), the process covariance transition equation. It, in turn, is driven by the additive-noise model **Sa(t+1)**. Thus, the criterion for

the additive noise model is that **Sn** approximates, in the least-squares sense, the errors about the predictions. Like OLS, the Kalman Gain is optimal; that is, it minimizes the model's variance. With **Sn** derived to fit the actual projected variance, this attribute can be used to derive the observation variance that yields the optimal Kalman Gain.

Learning Curve Data

The model ln(hours) = ln(a) + b ln(unit) has parameters that must be estimated from a set of historical learning curves. However, such data is typically treated as proprietary and not publishable. There is one source the author is aware of: the "Source Book of World War II Basic Data Airframe Industry Progress Curves"**. It has the unit identified direct labor hours for all U.S. WWII aircraft programs. However, the data is reported monthly, with only a single unit reported per month, not sequentially unit by unit. As noted earlier, the filter can be projected past the last actual data point by setting the measurement noise to a large value for the future unit or units. Using this approach, the filter could be applied to the WWII data. This study is, however, intended to demonstrate the Kalman Filter's application to an ongoing production program with sequential unit-by-unit direct labor hours, not a sparsely populated learning curve history. What is needed is a set of sequential data with the statistical properties of actual learning curves. The Theory of Complex Work* implemented as a simulation can produce learning curves, in this case, twenty 1000-unit curves.

The theory sizes a project by the number of tasks it involves. To produce the range of learning curves in this study, the number of tasks was varied over a uniform distribution from 2000 to 4000. To increase the slope variation to mimic the author's experience with learning curve history, the feedback of design changes was varied over a uniform distribution by a factor of 1.8. In addition, design changes were increased by 60% during the early portion of the curves in four simulations, creating a small hump. Figures 4 and 5 show the 20 curves plotted on a logarithmic scale. Some have a small hump, some flatten at larger unit numbers, and some steepen.

Four statistics were checked to verify that these simulated curves are representative of actual learning curves: the slope, the stationarity of the sequential variation, its magnitude, and its autocorrelation. The mean of the regressed slopes of the function, ln(hours) = ln(**a**) + **b** ln(**unit**+.4614), applied to the 20 curves is 80%, and its standard deviation of 2.1%. Sequential variation is calculated by removing the trend from the curves. This can be done by projecting the state transition one unit ahead, $\mathbf{X(t+1|t) = A(t)\ X(t|t)}$, and calculating its error, $\mathbf{y(t+1) - x_1(t+1|\ t)}$. Figure 6 shows the error sequence for a representative curve. The autocorrelation of the sequential variation of the twenty curves is shown in Figure 7. Its one-unit lag is -.4; beyond that, it is zero. This suggests a one-period lag moving-average model, as in the Kalman

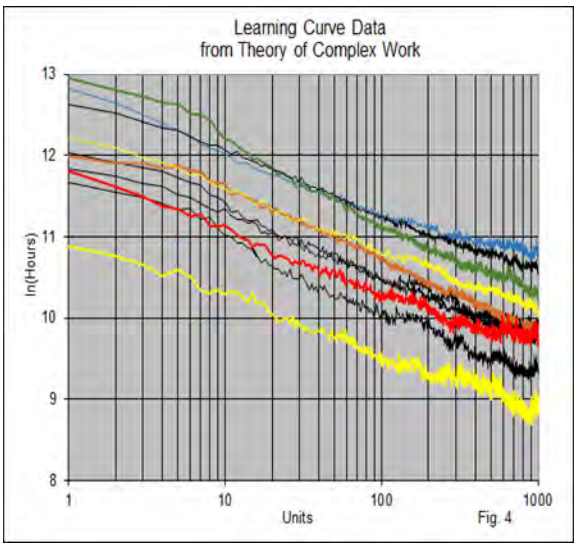


Fig. 4

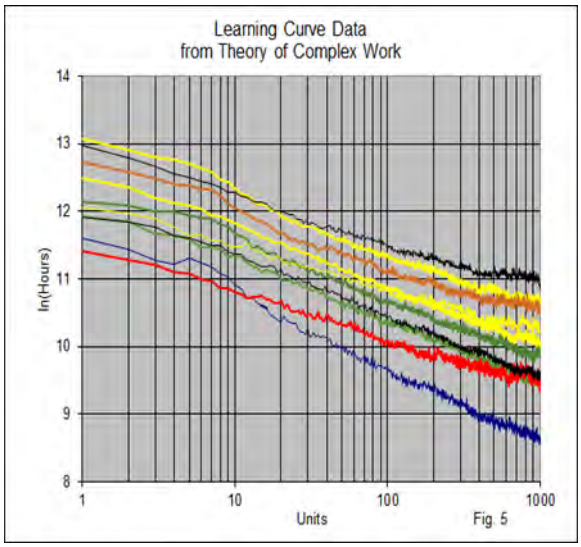
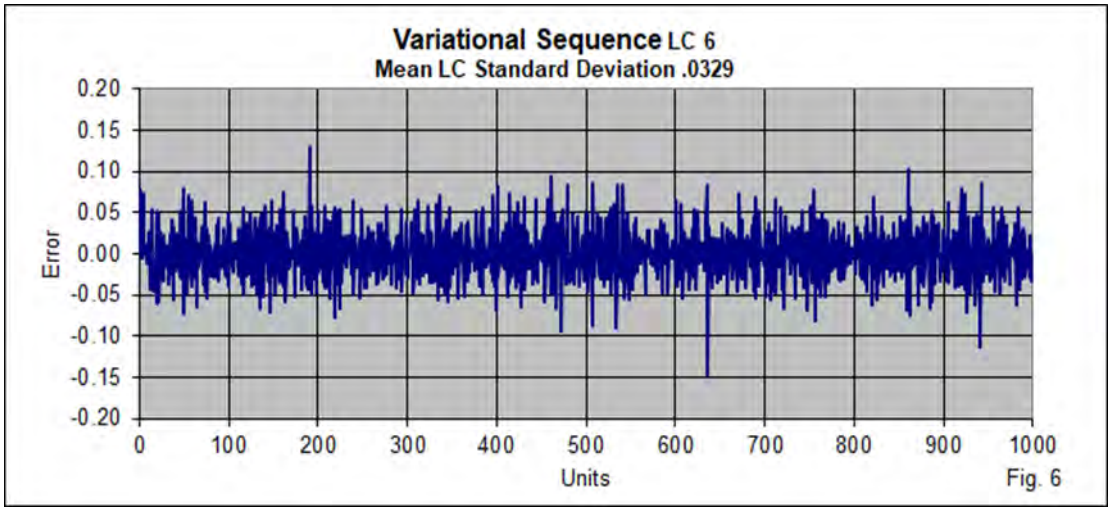
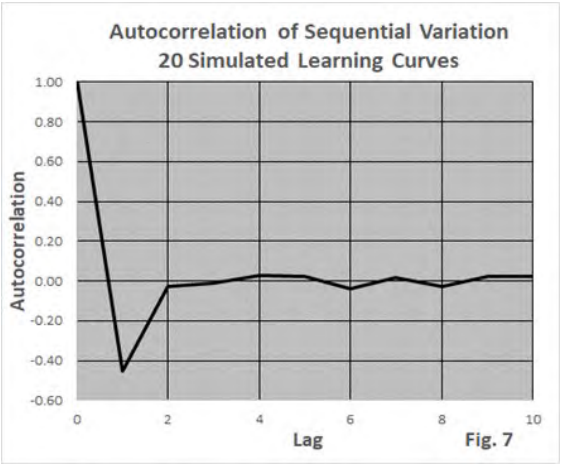
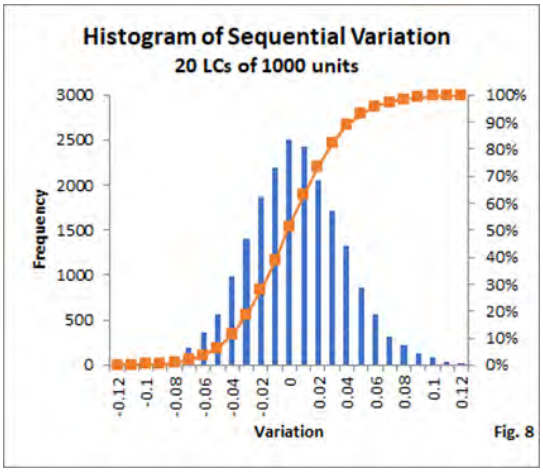


Fig. 5



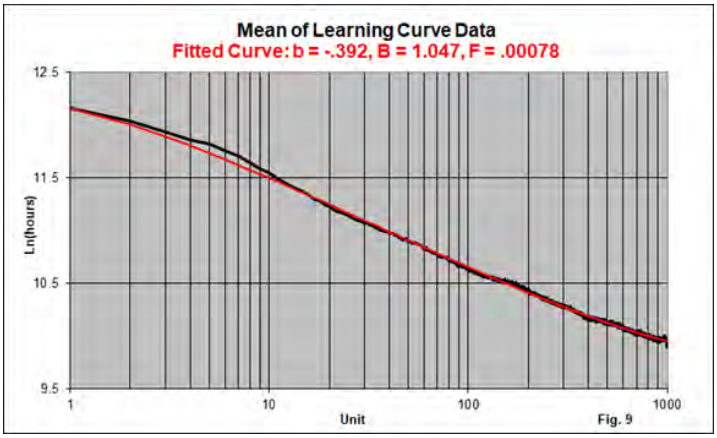


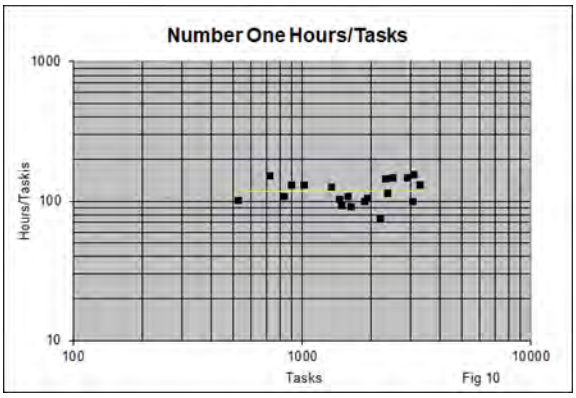


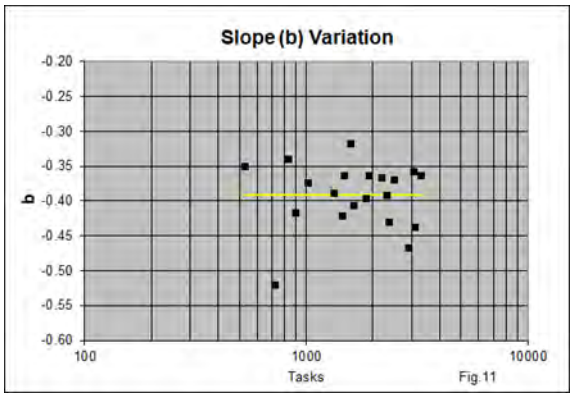
Filter's error update, Eq. (3). Figure 8 shows the histogram of the sequence. The variational sequences meet the noise requirements of the Kalman Filter; that is, they are Gaussian. And, with the application of the error update, random.

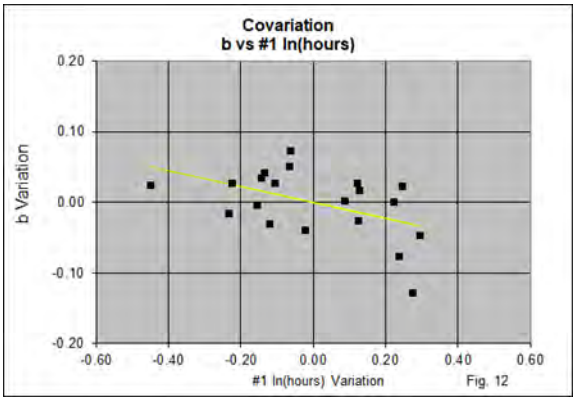
The A(t) Model

The **A(t)** matrix of the Kalman filter models the learning curves' expected dynamics, while the Kalman Gain, **K(t)**, incorporates the projection error information into the state vector, **X(t+1|t)**. The expected dynamics can be shown by taking the mean of the learning curve data unit by unit. In Figure 9, the black line represents the mean of the twenty learning curves. It has a small Stanford **B** hump and flattens slightly as it extends toward unit 1000. To model the Stanford **B** hump, a constant can be added to the denominator of $\mathbf{a}_{1,2}$, 1/(**t** +.4614 + **B**), while adding the term **F unit**^{2} flattens the slope for large unit numbers. The red line shows the model **X(t+1|t) = A(t) X(t|t)** with $\mathbf{a}_{1,2}$ = 1 /(**t** +.4614 + **B** + **F** $\mathbf{t}^2$), $\mathbf{X}_1(1|1) = \mathbf{Y}(1)$, and $\mathbf{X}_2(\mathbf{t}|\mathbf{t}) = \mathbf{b}_m$. The model parameters $\mathbf{b}_m$, **B**, and **F** are calculated to minimize the summed squared error between the mean and model lines using Excel's Solver.









Sn(1|0)	Covariance		Standard Deviation	
	#1	b	#1	b
#1	0.0415	-0.0047	0.204	
b	-0.0047	0.0022		0.047
Correlation		-0.486		Fig. 13

The Initial Values

The filter begins with the estimate **X(1|0)**, and its covariance. $\mathbf{X_1(1|0)}$ is derived from a CER estimating the learning curves' #1 values. Their source, the Theory of Complex Work, does not model a #1 weight-based CER but does size a project based on its tasks. Figure 10 shows the #1 hours-per-task for the 20 learning curves. Corresponding to the #1 hours, the values of **b**, best fitting each curve, are calculated. The variation around the model's slope, **b**, is shown in Figure 11. The covariation of the initial states is shown in Figures 12 and 13. The correlation between the #1 and **b** is negative; that is, learning curves with high #1 relative to their task-based CER have a steeper slope. Or, otherwise stated, these learning curves tend to converge.

The Covariance Models

The Kalman Filter has three covariance matrices: observation, **Mn**, process, **Sn**, and additive process, **Sa**. The process covariance has two roles. In conjunction with the observation covariance, it computes the Kalman gain (equation 6), which updates the state vector to reflect the latest actual values. Secondly, its square root is the standard deviation of the state estimates, both as they are updated and as they are projected into the future. In Figure 3, we see the tracking of the observed values through unit five. From unit six forward, the expansion of the process covariance is shown as the estimate moves beyond the last observed unit. The additive process covariance drives its expansion. $\mathbf{Sn_{1,1}}$ is modeled based on the data such that the projected process uncertainty equals the actual variation around $\mathbf{X_1(t+projection|t)}$. It was found that additive variance, driving **Sn**, was well modeled by the functions $\mathbf{Sa_{1,1}(t)} = \boldsymbol{at^b} + \boldsymbol{c}$, $\mathbf{Sa_{1,2}(t)} = 0$, and $\mathbf{Sa_{2,2}(t)}=0$.

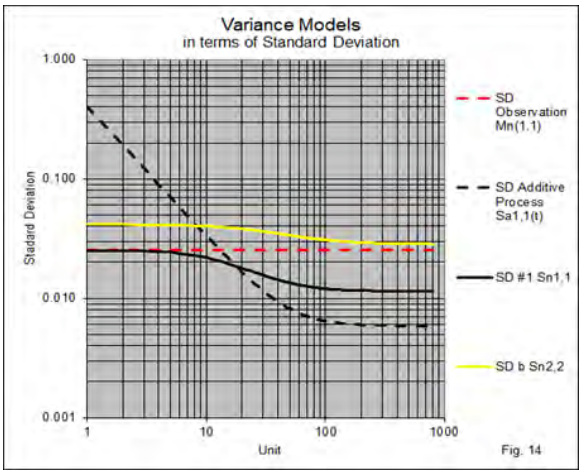
In Figure 6, we see that the learning curve's sequential variation is a constant. This suggests that the observation variance can be modeled as a constant, $\mathbf{Mn(t)} = \boldsymbol{m}$.

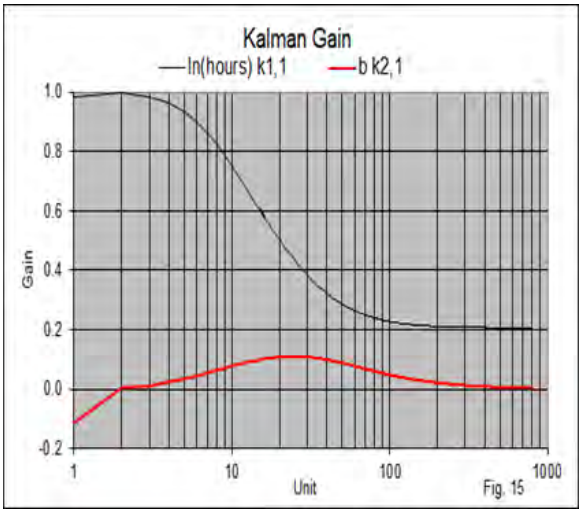
With these models, the Kalman Filter's variance parameters were estimated by Excel's Solver to find the ***a***, ***b***, ***c***, and ***m*** that minimize the sum of squared errors between the learning curve data and $\mathbf{X_1}$

(t+projection|**t)** plus the difference squared between the actual variation around $\mathbf{X_1(t+}$projection|**t)** and the modeled variance, **Sn(t+**projection|**t)**.

The covariance models are illustrated in Figure 14, along with the resulting Kalman gains in Figure 15. The standard deviation of the observation model is .025. For the additive process variance, the model $y = ax^b$ was initially fitted. It was found that the modeled variation beyond unit 100 was significantly lower than the actual variation around the expected value projections. Adding a constant to the additive process variance model increased $\mathbf{Sn_{1,1}}$, aligning the process standard deviations with the actual variation.

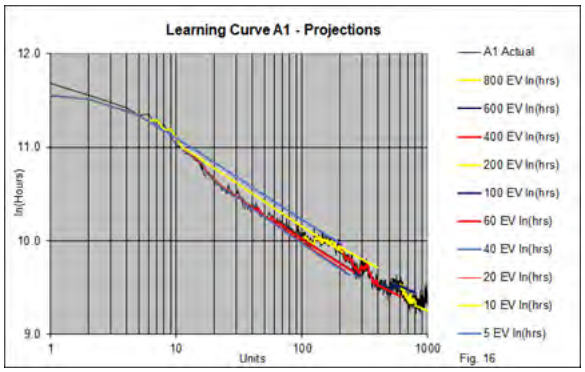
During the optimization, it was found that the best-fit additive process variance for **b**, $\mathbf{Sa_{2,2}(t)}$, is zero. In some respects, this is not too surprising. The slope of a learning curve in the Theory of Complex Work is not a fundamental property but rather an emergent property. That is, there is no actual process transition for slope, so there is no additive noise.

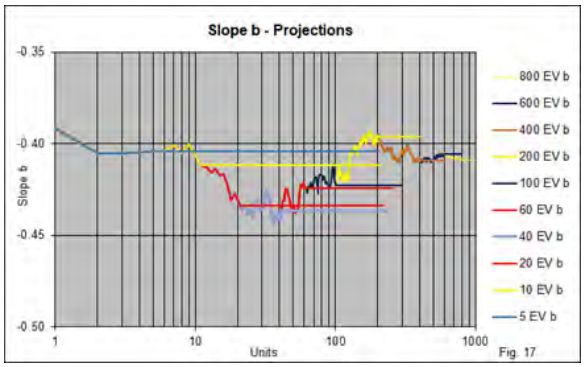


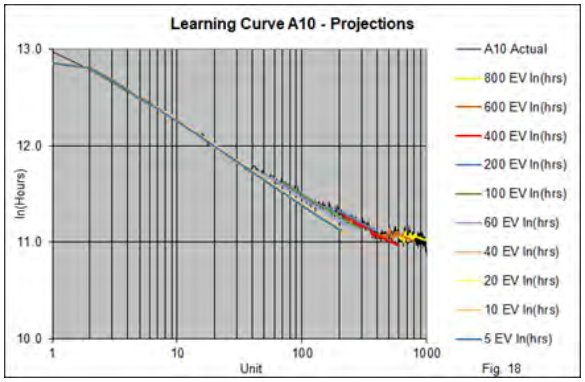


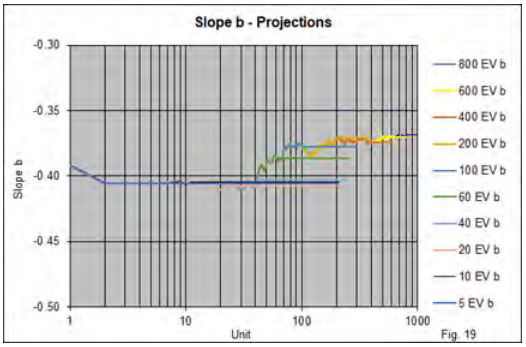
The Kalman Gain's $\mathbf{k_{1,1}(1)}$, the ln(hours) coefficient, is near 1, incorporating nearly all the CER's estimating error into $\mathbf{x_1(1|1)}$. By unit 100 and beyond, it is near .2. $\mathbf{k_{2,1}(1)}$, the **b** coefficient, is a negative -.1. This is due to the negative initial covariance $\mathbf{Sn_{1,2}(1,0)}$. By unit 800, it has dropped from a peak of .11 at unit 28 to .004.

These gains, applied to learning curves A1 and A10, yield the ten 200-unit projections of ln(Hours) in each of Figures 16 and 18. The blue line through unit 5 is $\mathbf{x_1(t|t-1)}$, the expected value of unit t conditioned on the data from its preceding unit. Their first value is the CER #1 estimate for the learning curve. The straight lines in Figures 17 and 19, beginning at units 5, 10, 20, etc., are the **b(t+**projection**)** values for each of the corresponding ln(Hour) projections.

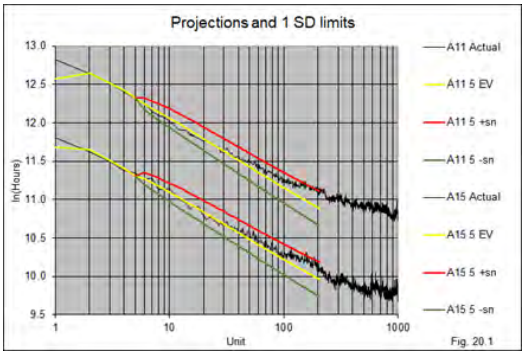


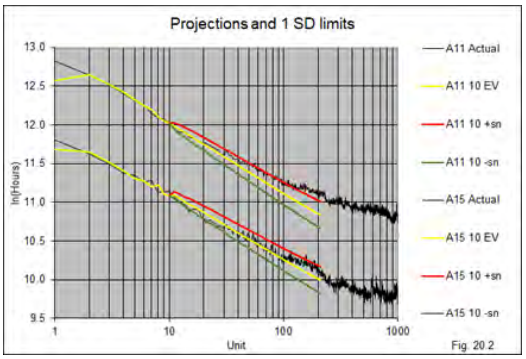


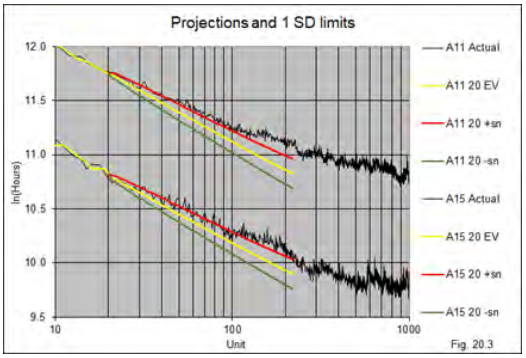


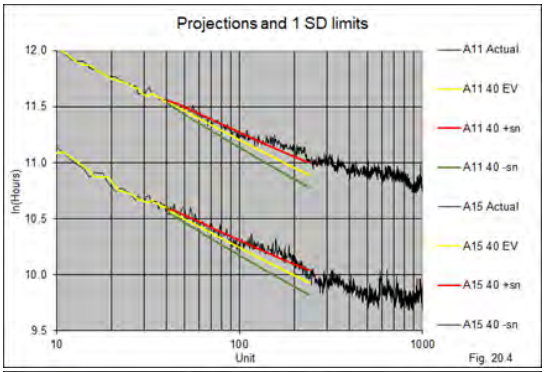


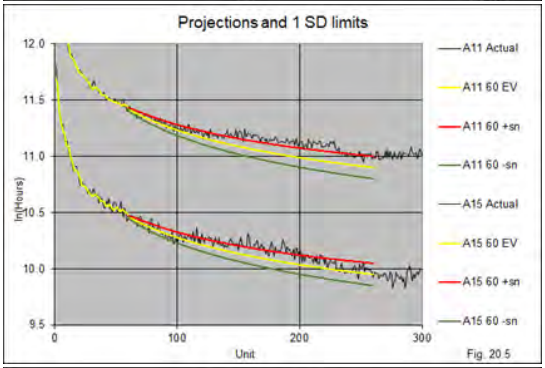
In Figures 20.1 - 20.8, the projections for two representative learning curves, A11 and A15, of the twenty curves. They are shown with their standard deviation limits. For the 20 learning curves and 10 projections, the actual values fall within these limits 68% of the time.

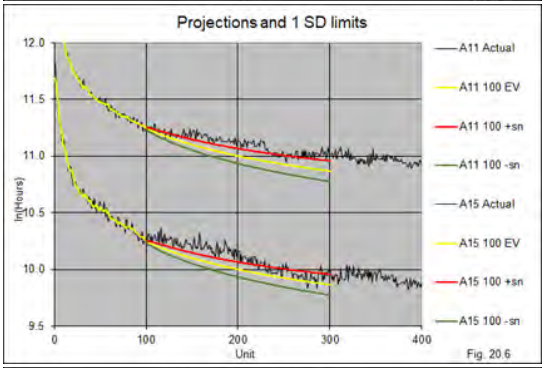


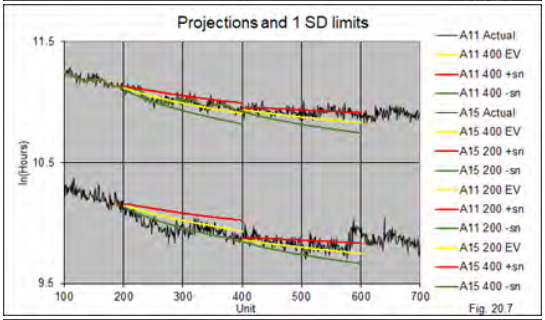


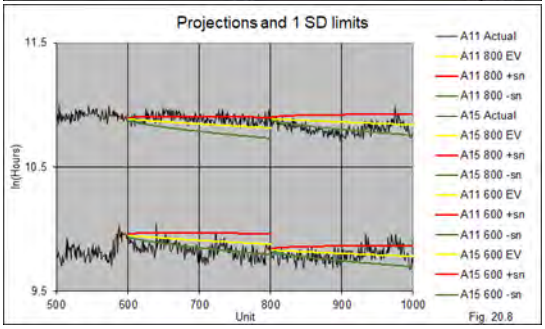












In Figure 21, five representative learning curves are shown, along with their corresponding slope predictions in Figure 22. The slope predictions from units 5, 10, 20, 60, 100, …, 800 are the values projected for the following 200 units at each of these units, for each learning curve. The slopes begin at the model's **b(1)** value, $\mathbf{x_2(1|0)}$ = -.391.

In Figure 21, the CER #1 estimate is plotted as a square with the same color as each corresponding learning curve. The abrupt change in slope from units 1 to 2 for each curve is due to the Kalman Gain for **b(1)**, $\mathbf{k_2(1)}$ = -.11, multiplication by the actual-estimate difference being added to $\mathbf{x_2(1|0)}$. For example, for the A11 learning curve, in black, the #1 actual is greater than its CER estimate. This difference multiplied by -.11 is a negative added to **b (1),** lowering its value at unit 2 to -.419.

The Kalman Gain, Fig. 15, for **b** is near zero at unit two, increases to near .1 in the twenties, and falls back to near zero beyond unit 200. The slope estimates correspondingly diverge over this range, stabilizing beyond unit 200.

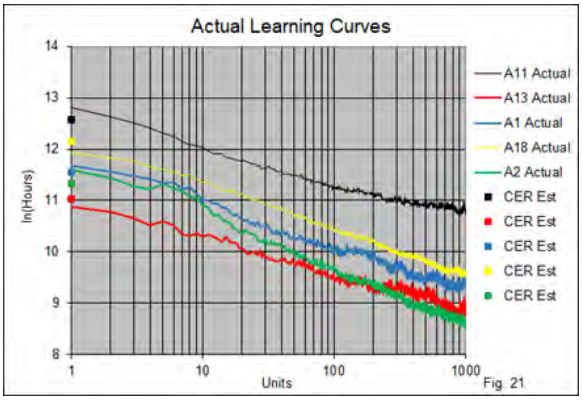


Fig. 21

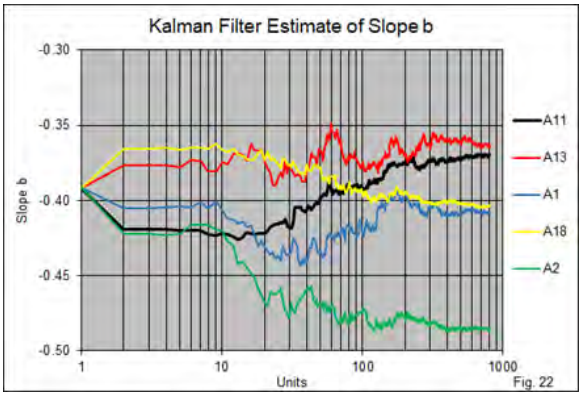


Fig. 22

In Figures 20.1 through 20.8, the yellow lines represent the expected values, while the black lines represent the actual values. From this data, the standard deviation per unit of the variation in actuals around expected values can be calculated across the 20 learning curves. In Figures 23 and 24, these standard deviations, rescaled around zero, are shown as solid jagged lines. The modeled standard deviations, the square root of **Sn(t+projection|t),** are shown as the dashed lines. These are the same lines as the red lines in Figures 20.1 through 20.8, but here they are rescaled around zero rather than about their expected values. At the bottom of each chart is a dashed line beginning at unit one. This is the standard deviation of $\mathbf{Sn_{1,1}(t|t)}$, after the observation **y(t),** that is, the uncertainty of the expected value of ln(hours) after being observed.

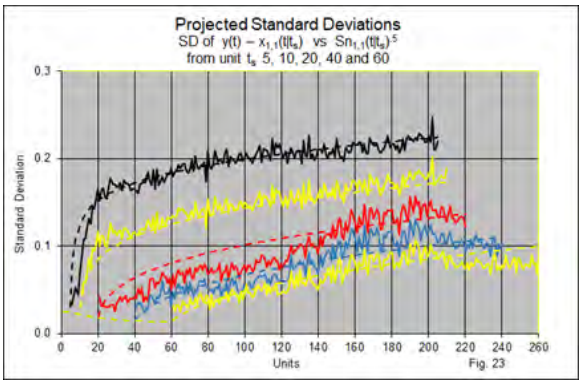


Fig. 23

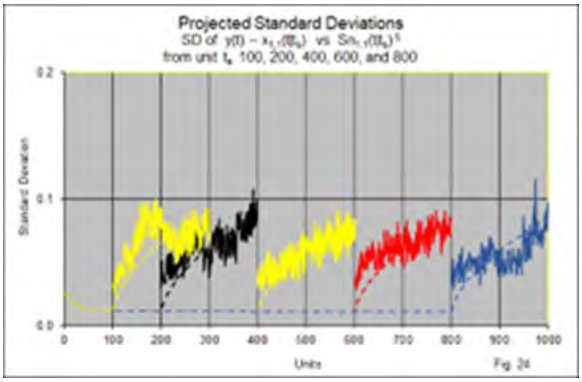


Fig. 24

Summary

The power function is transformed into a Markov form by taking its derivative. With two parameters added to fit the non-linearities in the simulated data, the state transition model was fitted to the mean of a set of simulated learning curves. With the state transition established, an error sequence was calculated dependent on the Kalman Filter's

covariance models. The free variables of the covariance models are estimated to satisfy the filter's optimality property: minimizing the sum of squared errors around the expected value estimates and the difference between the variance model and the actual variance.

In this application, the filter produces, by unit, the conditional PDF of a learning curve projection. The PDF enables the calculation of a wide range of risk-related applications, for example, when a conservative estimate is required for funding or financial reporting purposes, or when the PDF is split when the program faces an option choice.

Computation

The Kalman Filter algorithm can be coded on an Excel sheet. This two-state model has four elements in its covariance arrays. The filter's computations involve matrix additions, subtractions, and several multiplications. Each forecast requires the projection and update matrix operations by element, as well as the calculation of the plus and minus one standard deviation lines. For this two-state model with 10 200-unit projections across four learning curves, 964 columns were required in the Excel sheet. Roughly five times that number would be needed to code twenty learning curves. Although this is well within the upper bound of an Excel sheet, it would be a large task and a maintenance nightmare. Coding the mathematics in Visual Basic for Applications (VBA) while using the Excel sheet for input/output results in a single, verifiable structure that is far easier to develop and maintain.

The Excel add-in Solver uses a gradient-directed search to estimate the filter's free parameters. It calls a VBA function and minimizes its value, in this case, the sum of squared errors. A Solver-executed function has four properties relevant to this calculation: it can be coded to read a single array from a sheet, it returns its parameter estimates and function value to a sheet, it may call subroutines, and it cannot output formatted data to the sheet. The limitation of sheet output by a function is, of course, a problem. However, since a subroutine may call a function, a solution is to place the model and output coding in a subroutine that can be called by either a function or another subroutine. When called by Solver via the function, the output within the model is disabled; when called by the subroutine, it is enabled. Thus, Solver can execute the function with the output switched off, or the subroutine can call the model with the output switched on when the output is needed.

The simulated learning curve data, the Excel program, and the VBA code are available on the author's site:
https://harrylarsen.yolasite.com/

References

*Larsen, Harry, 2022. "Theory of Complex Work," MPRA Paper 113369, University Library of Munich, Germany. https://ideas.repec.org/p/pra/mprapa/113369.html

Larsen, Harry, "The Theory of Complex Work" Journal of Cost Analysis and Parametrics, Volume 10, Issue 2, November 2022, 90-103

https://www.iceaaonline.com/wp-content/uploads/2022/11/JCAPv10i3.pdf

** The "Source Book of World War II Basic Data Airframe Industry Progress Curves" unit direct labor hours are available in Excel format at: harrylarsen.yolasite.com

The International Cost Estimating and Analysis Association is a 501(c)(6) international non-profit organization dedicated to advancing, encouraging, promoting and enhancing the profession of cost estimating and analysis, through the use of parametrics and other data-driven techniques.

www.iceaaonline.com

Submissions:

Prior to writing or sending your manuscripts to us, please reference the JCAP submission guidelines found at

www.iceaaonline.com/publications/jcap-submission

Kindly send your submissions and/or any correspondence to:
jcapeditor@iceaaonline.org

International Cost Estimating & Analysis Association

8280 Willow Oaks Corporate Drive, Suite 600
Fairfax, VA 22031

703-642-3090 | iceaa@iceaaonline.org