Theory of Complex Work

Harry T. Larsen

Background

The manufacturing learning curve graph depicts the direct labor hours per unit of production plotted in its manufacturing sequence. When plotted with log/log scales, learning curves tend to follow a linear downward trend as shown in figure 1.

The power law, $\mathbf{y(x) = a\,x^b}$, has been found to describe the learning curve's trend, with $\mathbf{y(x)}$ = hours(unit), $\mathbf{x}$ = sequential unit number, $\mathbf{a}$ = the hours at unit one, and $\mathbf{b}$ the slope exponent. The power law is transformed to a linear form by taking its logarithm: $\ln(\mathbf{y}) = \ln(\mathbf{a}) + \mathbf{b}\ln(\mathbf{x})$. To summarize a learning curve's statistics, a least squares regression can be performed on the logarithms of learning curve data, $(\ln(y_1), \ln(1))$; $(\ln(y_2), \ln(2))$; $\cdots$, $(\ln(y_n), \ln(n))$. The regression estimate of $\mathbf{a}$ is termed the theoretical number one. By convention, the slope of a learning curve is $2^{\mathbf{b}}$. The learning curve is sometimes transformed to a cumulative average:

$$y_{cum}(x) = \sum_{i=1}^{x} y(i)/x$$

For this discussion, the hours per unit description will be used.

From an examination of historic learning curve's labor hours in the airframe industry, some attributes are apparent.

The slope of a long sequence of end item's labor hours is very seldom steeper than 70.7%, $\mathbf{b}$= -.5.

When new work, e.g., a new design, for part of an end item is introduced at a unit, the hours for the new work are approximated by $\mathbf{y(x) = a_{new}\,x^b}$, with $\mathbf{x}$ beginning at unit 1. The replaced work is removed via $\mathbf{y(x) = a_{replace}\,x^b}$, with $\mathbf{x}$ at its current value. With $\mathbf{a_{new}}$ = #1 hours for the new work and $\mathbf{a_{replace}}$ = the #1 hours for the replaced work. This produces a spike in the learning curve.

Processes earlier in the manufacturing process have flatter slopes than do the later activities.

In the aircraft industry major and final assembly curves often steepen as they progress, while fabrication curves typically are less steep.

The learning curves of very large quantities of end items flatten.

The learning curves of products with large aggregates of hours per unit tend to be smoother than learning curves of fewer hours per unit.

The probability distributions about regressions of power law to labor hour data are approximately log-normal. (This is true for both learning curve hours per unit as well as Cost Estimating Relationships (CERs) for labor hours.)

There is a large cost variation around estimates for complex projects, particularly those that involve new technology.

Although the power law describes broadly the learning curve's log-linear behavior, it does not offer insight into the causes or mechanisms of its realization. Nor does it suggest means by which its labor hours per unit may be affected .

Theory of Complex Work

This model of work is based on the concept of a task.

The construction of a product is accomplished by the successful completion of a set of tasks.

A task is an activity that has a criterion of success.

A task is accomplished through repetitive trials.

The trials are repeated until the criterion of success is met.

A trial is modeled as a binomial event, with a probability of success **p(t)**, with **t =** trial.

At the completion of each trial, **p(t),** is increased by an amount **(1 - p(t)) dl,** where **dl** is a constant and **dl** << **p(t)**.

p(t+1) = p(t) + (1 - p(t)) dl

The time duration of each trial is a constant.

Thus, a task's labor hours are proportional to the number of trials necessary to successfully complete the task. A task may be a part of a larger interrelated group of tasks. The outcome of a task's successful trial may cause another task's success criterion to change, requiring it to be redone. The successful completion of a product's tasks produces a unit of the learning curve.

To model this process, at each trial, **p(t)** is compared to **x ~** U(0,1), an event from a uniform distribution. If **x** < **p(t)** the trial is a success and the task for that unit is completed, otherwise the trial is repeated. Trials for the next unit's task begin with the **p(t)** from the preceding completed task. The sequential completion of tasks produces the learning curve.

This is a stochastic process, with inherent uncertainty. To produce an estimate, it is implemented as a Monte Carlo system, producing Probability Density Functions, PDFs. From those PDFs various statistics can be calculated, e.g., median, mean, and standard deviation. Or, if the estimate is to support a decision option, the PDF can be bifurcated.

Fig 1 depicts a learning curve created by such a sequential completion of tasks. Its initial probability, **p(0)**, is .03 while **dl** equals .001, with 400 tasks and a trial time of 1 hour.

Predictions

The expected value of the number of trials to a successful completion is approximately 1/**p(t)**. Thus, with **p(t)** small, the change in **p** between

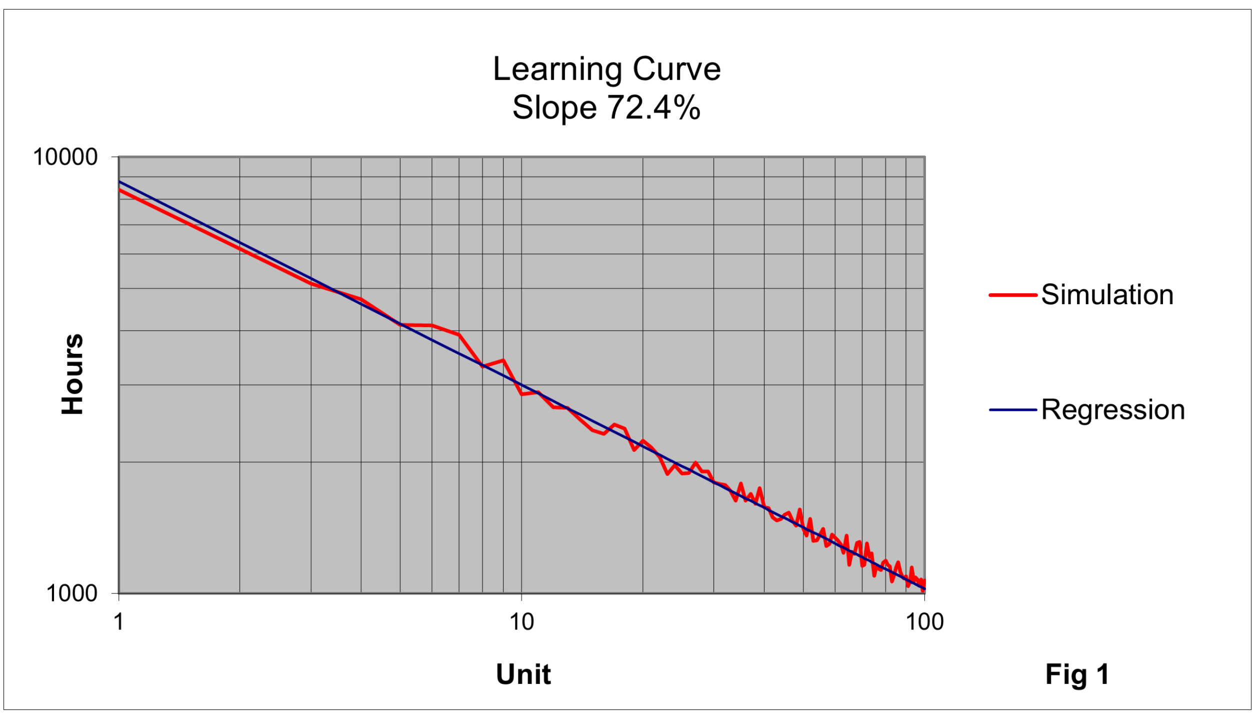


Fig 1

successful trials is about **dl/p(t)**. If this value is small the number of trials to successfully complete a set of tasks is a negative binomial distribution. A negative binomial distribution typically is defined as the discrete probability distribution of the count of failed trials up to a successful trial in a set of experiments. However, for this application, the count includes the successful trial.

The mean of a negative binomial distribution is:

(1) $\mathbf{u = T\ N / p}$, where **N** is the number of tasks, and **T** is the time per trial for a task.

Its standard deviation:

(2) $\boldsymbol{\sigma = T\ (N\ (1-p))^{.5} / p},$

The relative standard deviation is:

$$\boldsymbol{\sigma / u = (T\ (N\ (1-p))^{.5} / p) / (T\ N / p)}$$

or

(3) $\boldsymbol{\sigma / u = ((1-p) / N)^{.5}}$

The number of tasks is: From (3)

(4) $\boldsymbol{N = (1-p) / (\sigma/u)^{2}}$

From equation (1) it can be seen that a project's expected hours are proportional to the size of the project in terms of the number tasks, **N**, trial duration, **T**, and the difficulty of the project measured by the inverse of a trial's probability of success, **p.** Thus, a project may have high labor hours due to either its task content or its difficulty.

From equation (2), for **p** << 1, the standard deviation of labor hours is proportional to the square root of the number of tasks and the inverse of a trial's probability of success, 1/ **p.** Consequently, as illustrated by the relative standard deviation formula (3), a project with a higher task content, while holding **p** constant, will have a relatively lower standard deviation. Hence its depiction on log/log scales becomes smoother as the project's task content increases. Conversely, for a project that increases in size is due to increased difficulty (smaller **p**), its sequential standard deviation remains directly proportional to the increased labor hours and does not collapse with the larger project size. Thus, we should expect that large projects that advance the state of the art will have high relative standard deviations, while projects that do not will have low relative standard deviations.

For small **p** the relative standard deviation, $\boldsymbol{\sigma/u}$, is a function of **N.** Figure 2 depicts two learning curves, one with 200 tasks and one with 40 tasks. The relative standard deviation of the 40 task curve is $(200/40)^{.5}$ times the 200 task curve.

For very large quantities a learning curve may flatten. When **p** approaches 1 the hours per unit

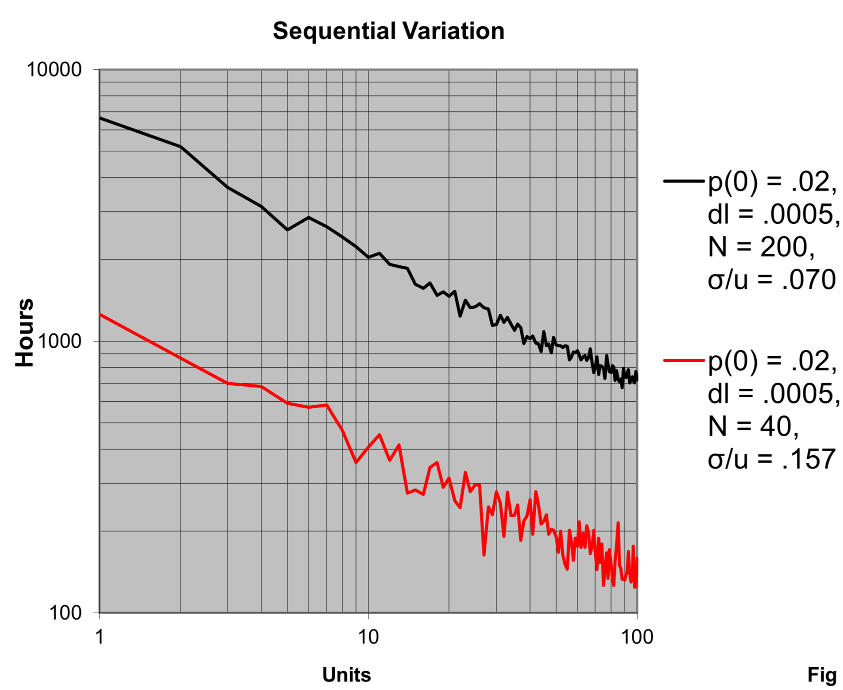


Fig 2

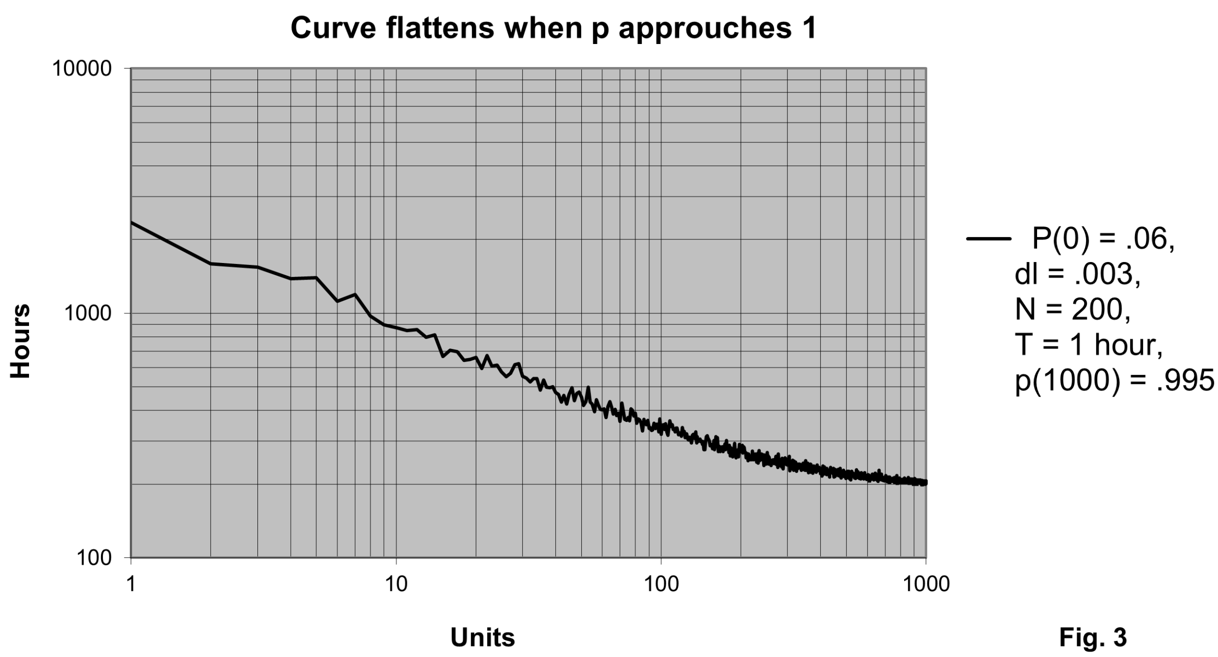


Fig. 3

approaches the product of the number of tasks and the hours per trial.

Figure 4 illustrates that **p(0)** and **dl** determine the hours for the first unit. As **p(0)** is increased the learning curve has a lower initial cost but remains asymptotic to a slope of 70.7% until **p(t)** approaches 1.

A combination of learning curves with differing **p (0)** values can produce a curve with a shallower slope.

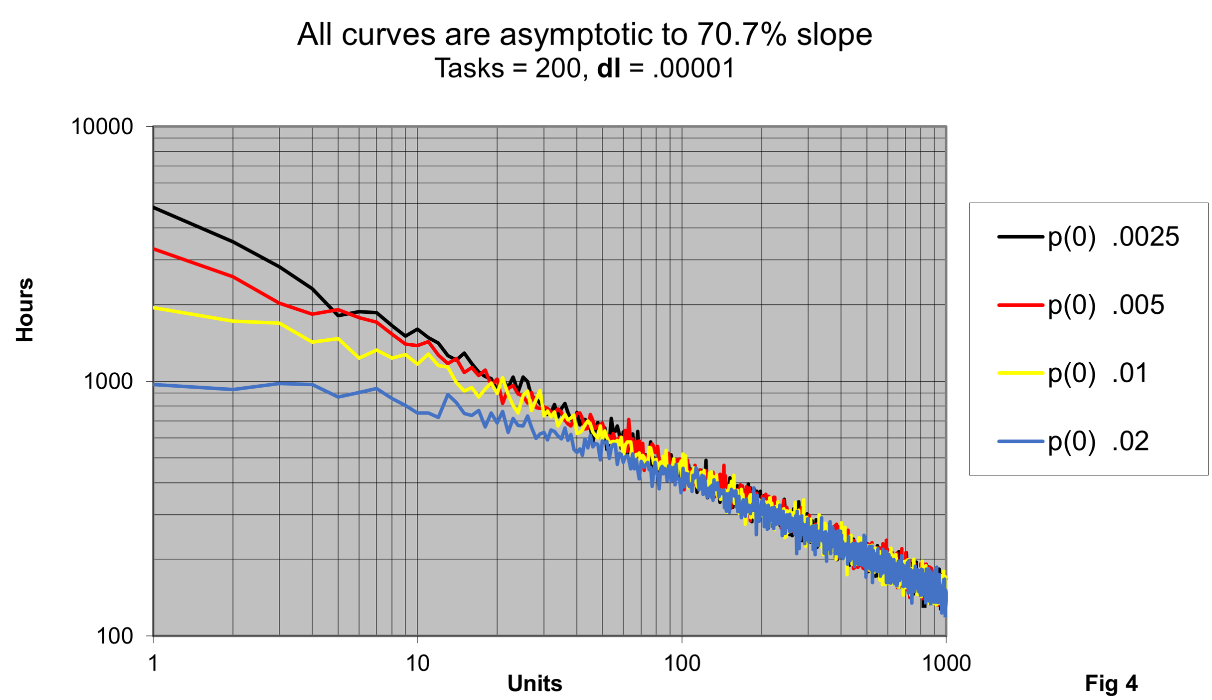


Fig 4

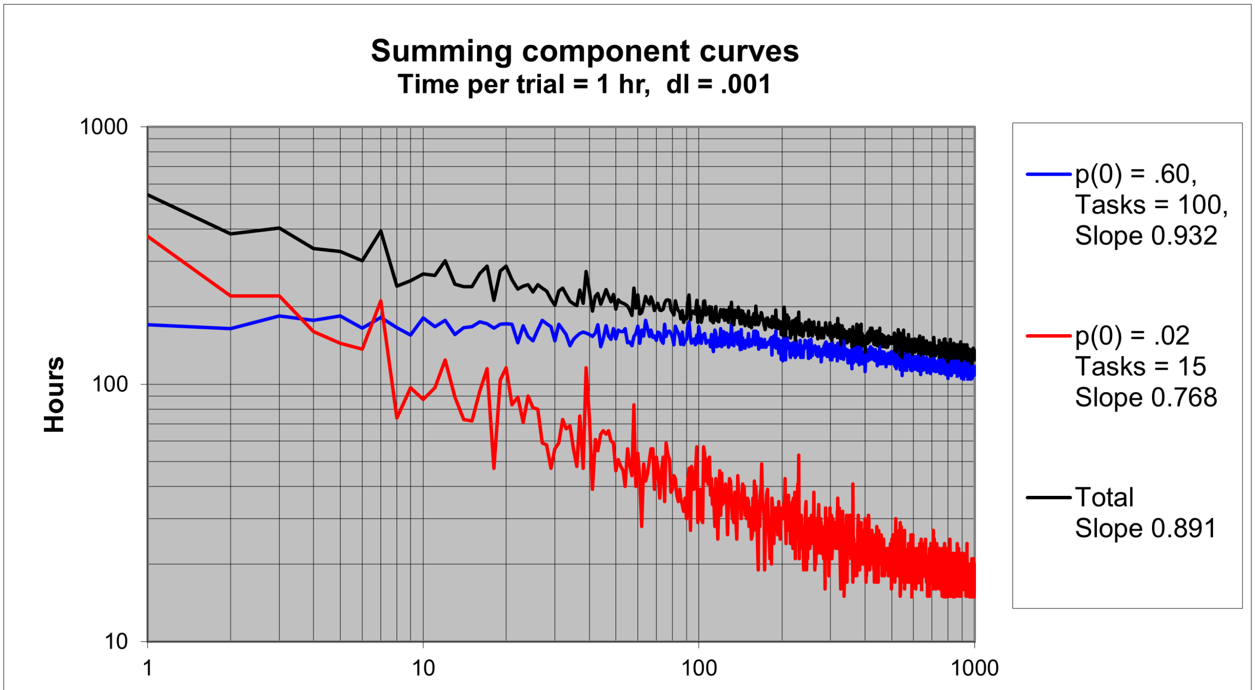


Fig 5

More realistically, a distribution of **p(0)** values can produce a learning curve with a small hump followed by a flattening. Figure 6 is a histogram of a lognormal distribution of 200 **p(0)** values, with a mean of .233 and a standard deviation of .133. Figure 7 is the resulting learning curve of 200 corresponding tasks beginning with those p(0) values. A log-log regression through unit 500 of the 1000 units shown has a slope of .822, a slope in the range commonly seen in aircraft learning curve history.

If a design change is introduced to an existing production process, it is added at the initial probability of the task, **p(0)**. The work replaced is removed at its current probability, **p(x)**.

In figure 8 the yellow line represents 70% of the tasks, which are unchanged. The red line scaled from one shows the new tasks. The orange, beginning at unit 200, shows the sum of new tasks and the continued unchanged tasks.

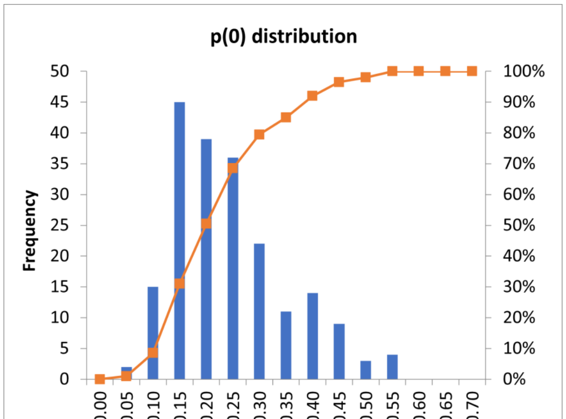


Fig 6

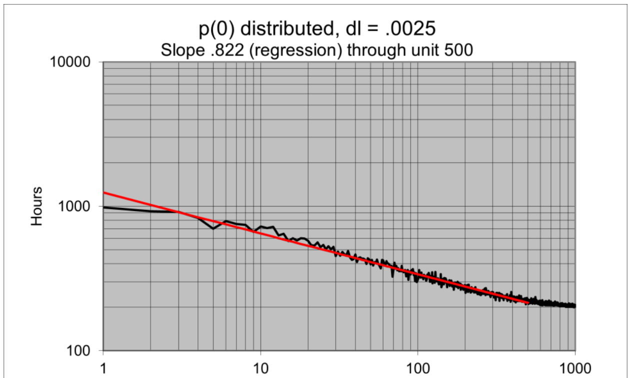


Fig 7

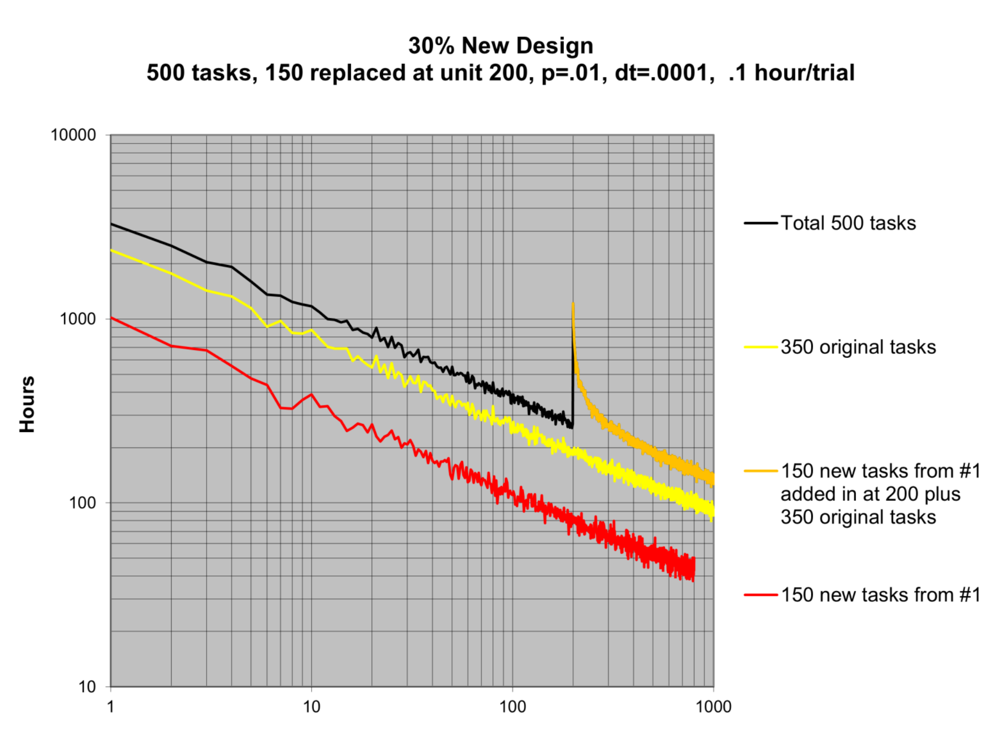


Fig 8

Feedback may occur between its tasks. When, for example, in an assembly task if a part does not fit due to a design or manufacturing error, the part may be reworked in the assembly activity to fit, but also, a design or specification change may be fed back to the fabrication area. The design change is treated as a new design, setting that task's current **p** to **p(0)** as shown in figure 6, but for a single task. Figure 9 shows a simulation of two sets of 150 tasks, fabrication and assembly. Design changes are fed back from assembly to fabrication. In this example, the likelihood of a fabrication task's **p** being reset to **p(0)** is proportional to the product of .0015 and the ratio of the number of trials for the preceding unit in assembly versus those in fabrication. Without

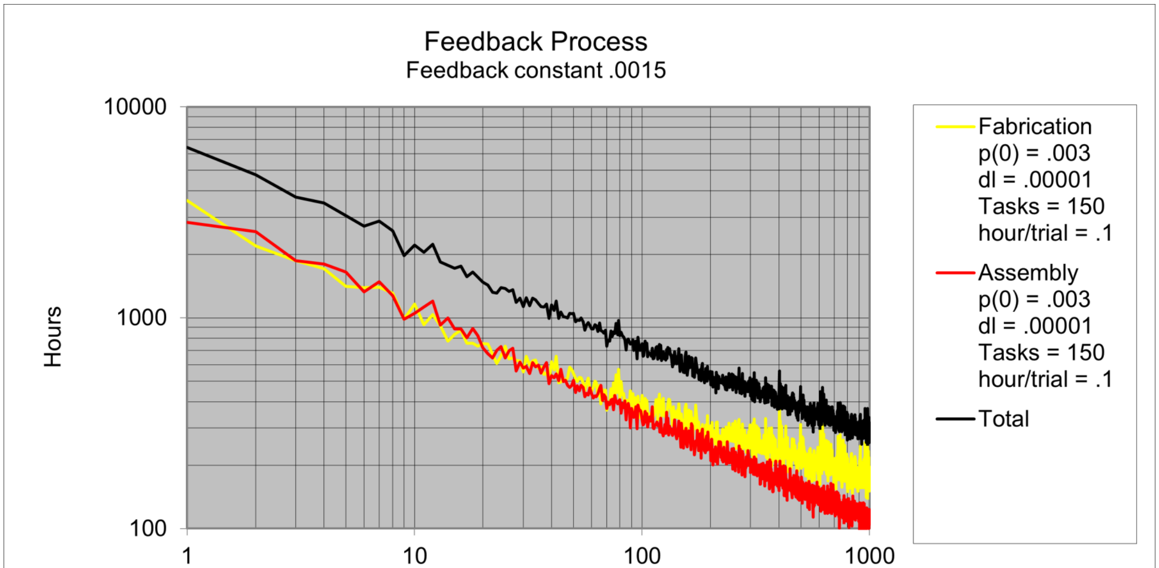


Fig 9

feedback, the fabrication curve would follow the trend of the assembly curve.

The Derivation of Model Parameters

For a system that cannot be transformed into a form suitable for the application of a regression, the author would normally use Excel Solver to find the least squares solution. Such a solution requires that gradients be calculated for each parameter at each step of the search. For a Monte Carlo system producing a partially random output a very large number of model executions would be required. Probably not computationally feasible, at least on a personal computer. Fortunately, a deterministic formulation of the model that very closely approximates the stochastic version is possible.

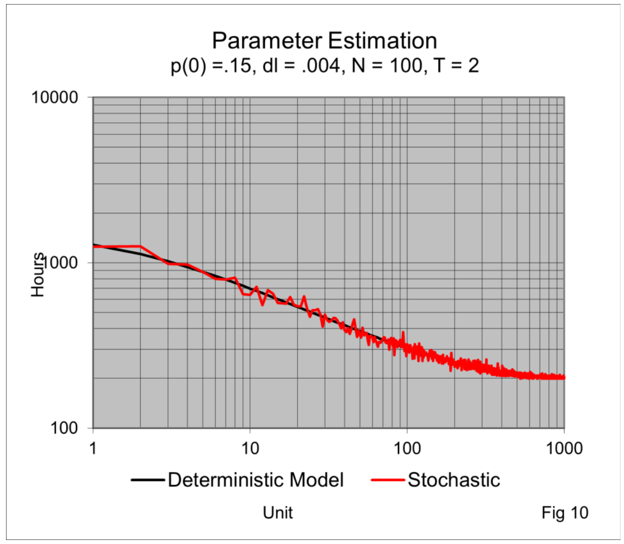
The model calculates labor hours to complete a task by iterating **$p(t+1) = p(t) + (1 - p(t))\, dl$** until $x \sim U(0,1)$ is less than **$p(t)$**. The average number of iterations for $x <$ **$p(t)$ is $1/p(t)$.** Thus deterministically,

$p(unit+1) \sim p(unit) + (1 / p(t))\,(1 - p(t))\, dl$, where **$p(unit)$** is the average probability during a unit's trials. The hours(**unit**) are equal to **$T\, N / p(unit)$.**

Figure 1 shows the calculation. For example for unit 2: **$p(unit\ 2) =$** $.178 = .156 + 6.4 * (1 - .156) * .004$.

Hours(**unit 2**) = 1126 = 2 * 100 / .178.

Table 1		**Deterministic fit**	**Stochastic model**
	p	0.156	0.150
	dl	0.004	0.004
	T	2.06	2.00
	N	97.2	100
Unit	**p(unit)**	**Deterministic Model**	**Iterations**
1	0.156	1284	6.4
2	0.178	1128	5.6
3	0.196	1021	5.1
4	0.213	942	4.7
5	0.227	881	4.4



The parameters of Table 1 generate the learning curve in figure 10. Both the deterministic and stochastic versions are shown.

The parameters of the model are **N**, **T**, **p,** and **dl**. **N** and **T** cannot be separately estimated from just the hours(**unit**) data. However formula 3, **$\sigma/u = ((1-p) / N)^{.5}$**, provides a means of calculating **N.** **$N = (1-p) / (\sigma/u)^2$** . It requires the calculation of **σ/u**, the relative standard deviation. One method is to calculate the sequential variation around the expected value of the hours per unit. Then correct the resulting standard deviation for the additional variation introduced by the sequential differencing. That correction factor is the square root of .5. An approximation of the relative standard deviation is thus: **$\sigma / u =$** .707 STDev(2 (hours**(nunit+1)** **-** hours**(unit)**) / (hours **(nunit+1)** **+**hours**(unit)**) taken over the range of units.

Once **N** is estimated, Excel's Solver can be used to find the **p(0)**, **dl**, and **T** that minimize the difference between the actual and deterministically modeled learning curve.

Of course, if the actual learning curve has dynamics beyond those modeled by the negative binomial distribution, those dynamics should be modeled. The modeling of design change feedback into manufacturing was introduced

Table 2						
Tasks						
WBS/CE	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Total**
Fabrication	125	23	15	8	10	181
Minor Assy	100	69	45	15	20	249
Major Assy	25	138	90	52	70	375
Total	250	230	150	75	100	805

earlier. It can be included in the Table 1 formulation by resetting the **p(unit)** values to **p (0),** in proportion to the ratio of design changes to the total designs at each unit.

Time Domain Formulation

During the design of a product, there may be information flows between engineering tasks. But unlike a production line, these new requirements are applied to the design tasks themselves rather than to a subsequent design. Modeling the equation, $\mathbf{p(t+1) = p(t) + (1 - p(t))\, dl}$, in the time domain for both engineering and manufacturing allows these information flows to interact within and between the design tasks and factory tasks.

To show these effects a small airplane program is modeled. It produces 300 aircraft over 7 years. Its design effort begins 3 years before the first factory complete aircraft. Table 2 shows 805 engineering tasks in this simulation each with an associated task set in manufacturing.

The engineering tasks are organized into a Work Breakdown Structure of 5 elements, and the corresponding manufacturing tasks into three cost elements. For example, there are 250 Structure design tasks. One hundred twenty-five of those designs are built in fabrication.

With modeling in time, design problems encounter in minor assembly and major assembly can be fed back to engineering where the design is changed and then sent on to the manufacturing elements. This creates a sustaining engineering effort as well as an increase in fabrication and minor assembly hours. In table 3 shows there is a .0006 probability that a trial in a major assembly or minor assembly task will cause a design/fabrication task to be redone, with an additional .0006 that a major assembly trial will cause a design/minor assembly task to be redone. This generates design changes proportional to the hours worked in manufacturing. After the design change is completed it is sent on to its corresponding manufacturing element where its **p** is set to **p(0)**.

Table 3									
Feedback Parameters									
WBS	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Cost**			
Structure	0.05	0.05	0.05	0.05	0.05	**Element**	**Fabrication**	**Minor Assy**	**Major Assy**
Subsystems	0.05	0.05	0.10	0.15	0.20	**Fabrication**	**0**	0.0006	0.0006
Avionics	0.05	0.05	0.10	0.15	0.20	**Minor Assy**		**0**	0.0006
Systems Eng	0.05	0.05	0.10	0.15	0.20	**Major Assy**			**0**
Test	0.05	0.05	0.10	0.15	0.20				

Table 4									
Program Parameters									
WBS	**Structure**	**Subsystems**	**Avionics**	**Systems Eng**	**Test**	**Cost Element**	**Fabrication**	**Minor Assy**	**Major Assy**
Start	10/01/23	10/01/23	10/01/23	06/01/23	12/01/24	#1 flow months	21	21	21
Planned complete	12/01/25	12/01/25	12/01/25	06/01/25	09/01/26	Flow red. slope	0.94	0.94	0.94
p(0)	0.003	0.003	0.003	0.003	0.003	p(0)	0.003	0.003	0.003
dl	0.00001	0.00001	0.00001	0.00001	0.00001	dl	0.00001	0.00001	0.00001
Hours/ task	3	3	3	3	3	Hours/task	0.50	0.75	1.25
Tasks	250	230	150	75	100	Tasks	250	272	283

Also, within engineering, .05 of Subsystems completed tasks produce a design change in Structure, while .15 of Systems Engineering produce design changes in Avionics, and .20 of Test produce design changes in Test. These changes are treated like repetitive units in manufacturing, that is, the probability, **p**, is iterated from its last successful completion.

To complete the description of the airplane program table 4 shows the engineering and manufacturing initiating parameters. Figure 11 has the aircraft production schedule. Its blue line shows the beginning of the fabrication effort for each unit. The time domain model is programmed to expend labor hours as needed to complete each unit on time. Thus, the planned complete and complete lines are nearly coincidental. This is of course unrealistic; but was done to keep this paper focused on the learning curve.

Figures 12, 13, and 14 show four simulations, black, red, yellow, and blue, each with an increased level of feedback.

The black lines show the engineering and manufacturing headcounts and learning curve when there is no feedback. The engineering effort is completed on 6/1/26 on the planned schedule. The learning curve slope is 71.6%.

The red lines have feedback initiated only in manufacturing. This feedback produces design changes in engineering, generating sustaining engineering. These design changes are fed back

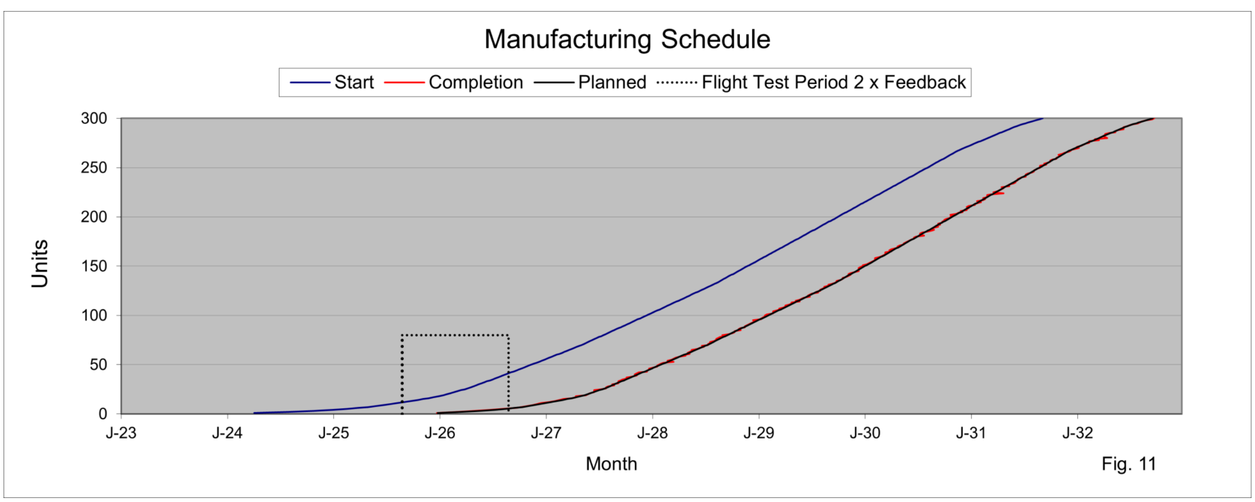


Fig. 11

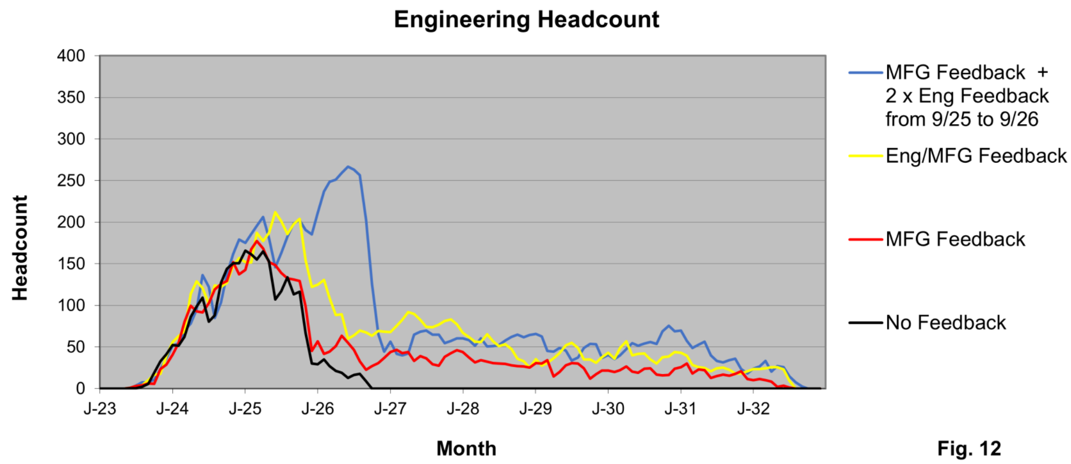


Fig. 12

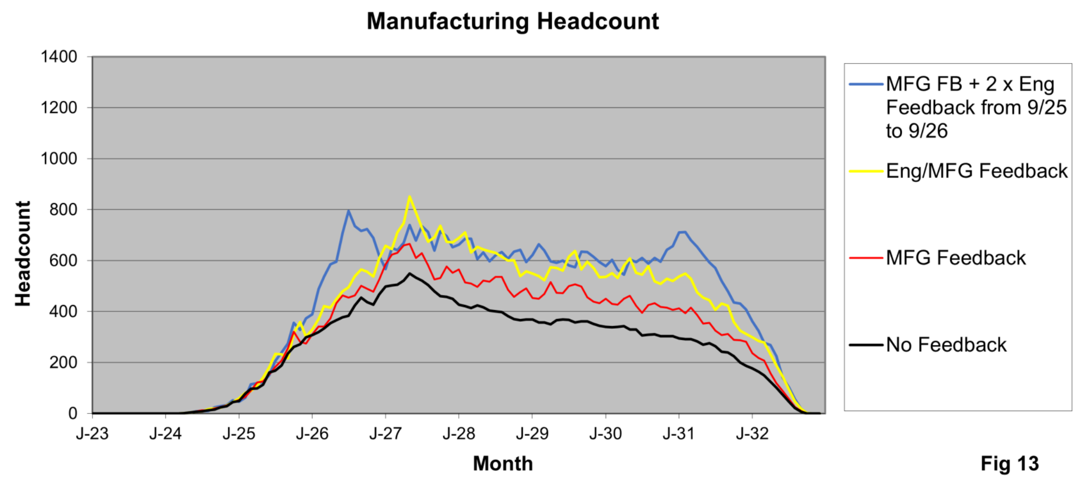


Fig 13

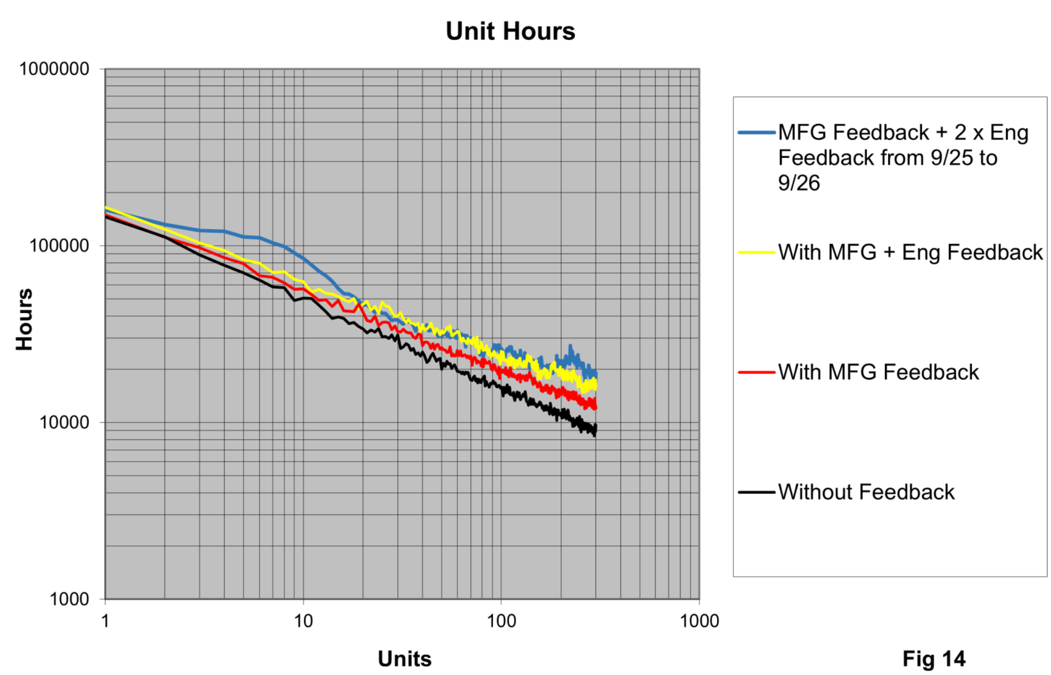


Fig 14

into manufacturing resetting its task's current **p(t)** values to their **p(0)**. The increased resources are seen in the red line of the manufacturing headcount chart and in the flattening of the learning curve.

The yellow lines show the effect of including the internal engineering feedback.

In a project that stretches the engineering capability, due to a new technology, inexperience, or other causes, the design process may have errors. These errors can be expected to show up in the later stages of the first aircraft's production and during the flight test period.

The blue lines illustrate the effect of those errors being doubled between 4 months prior to the first aircraft's completion and the completion of the engineering test effort, depicted as the box in figure 11. The simulated learning curve has the hump often seen in these circumstances. It is followed by a steep decline and then a transition to a more typical curve.

Uncertainty

When this stochastic process is implemented as a Monte Carlo system the predictions are in the form of Probability Density Functions, PDFs. In figures 12 - 14 there are two fundamental causes of variation, the uncertainty of the trial outcomes and the feedback processes.

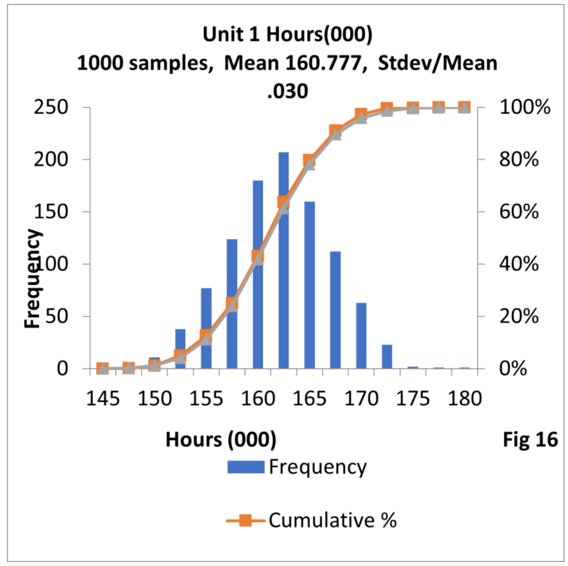


Fig 16

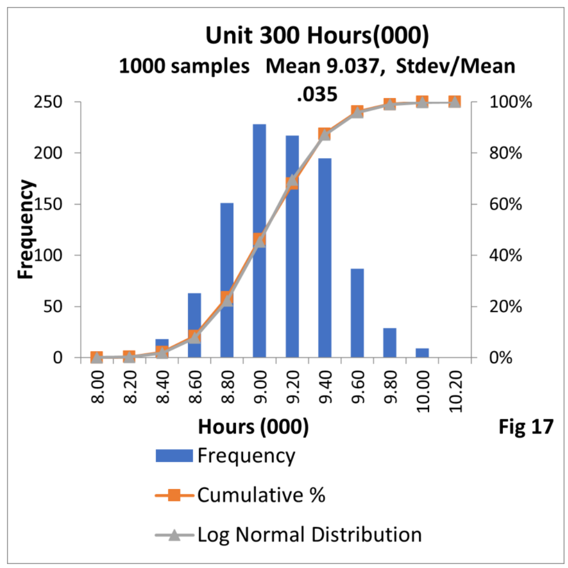


Fig 17

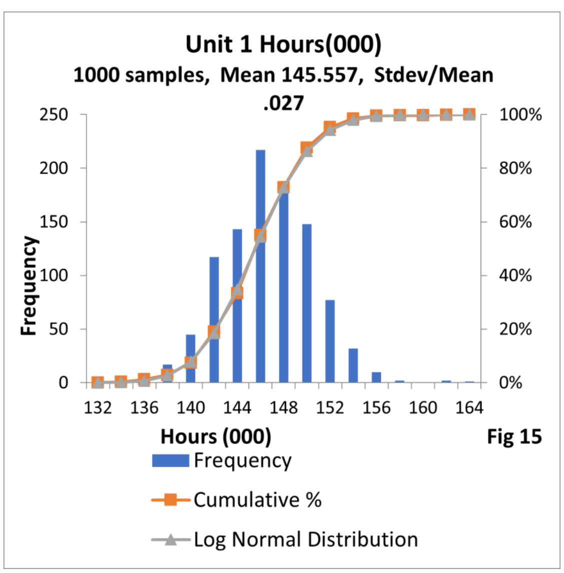


Fig 15

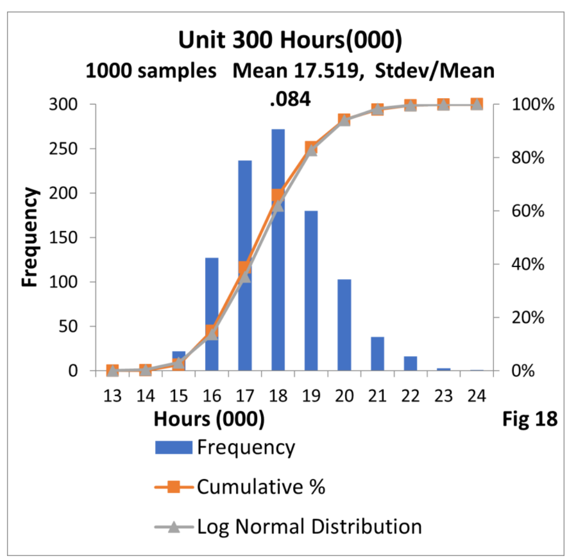


Fig 18

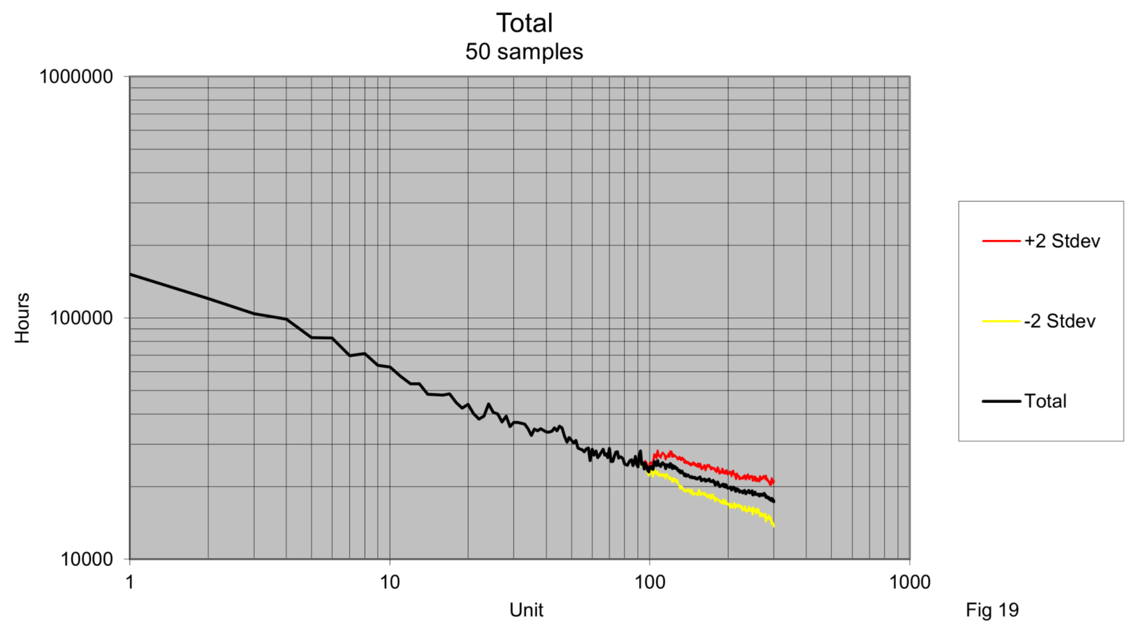


Fig 19

The histograms in figures 15 – 18 are from 1000 iterations of the time domain model. Figures 15 and 17 show the PDFs of unit 1 and 300 hours with only the trial outcome uncertainty, **p(t)** > **x** ~U(0,1). Figures 16 and 18 show the PDFs including the feedback processes; while holding all other input parameters constant. Feedback from design changes increases the variation as the project progresses. Figure 18 shows about twice the variation and hours as the corresponding simulation without design changes, figure 17.

All the distributions are close to lognormal.

Fixing the Monte Carlo model's random number generator to a single sequence for the first 6 years of the simulation produces learning curves with its uncertainty beginning at unit 96, shown in Fig 19, with expanded scales in figure 20.

It is notable that the uncertainty does not grow as one might expect from a typical random walk model. When the iterations of **p(t)**, and thus labor hours, to a successful completion are greater than 1/ **p(t)**, **p(t)** becomes larger than expected. For

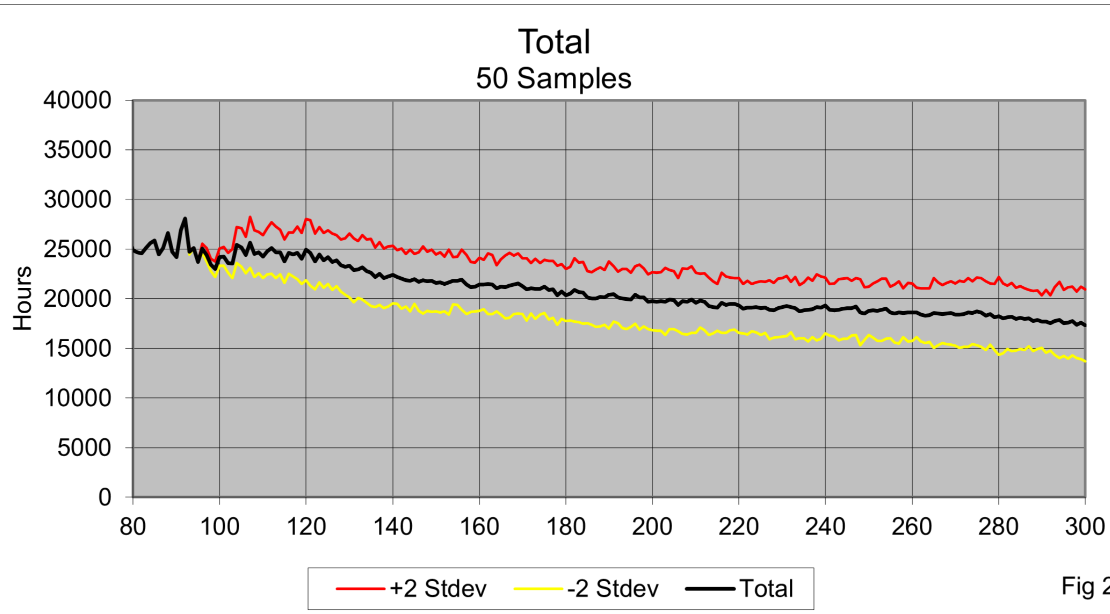


Fig 20

the next unit's iterations that larger probability reduces the expected number of iterations. Thus, a slight autoregressive dynamic is produced.

Information

The model links work to information. Figure 22 shows the bits per successfully completed unit for the two learning curves in figure 21. Information is calculated as:

$$I(\mathbf{unit}) = \left(-\sum_{i=t}^{t+n-2} \log_2(1-\mathbf{p}(i))\right) - \log_2(\mathbf{p}(t+n-1)),$$

where t is the first trial of **unit** and ***n*** is the number of trials to the successful completion of **unit.** Both curves have 1000 tasks and a trial time of 1 hour. The black curve, with p(0) = .01 and dl = .00017, has a slope of .717. The blue curve, with a **p(0)** = .15 and **dl** = .003 was chosen to illustrate a curve with a hump that flattens as **p** (large) approaches one.

From an information perspective, work can be thought of as the effort required to resolve an uncertainty. The relationship between a product's complexity, that is uncertainty to be resolved, and the work to resolve it is a subject for information theory. Connecting work with information allows the mathematics of information theory to be brought to bear on the nature of work.

Summary

The iteration of **p(t+1) = p(t) + (1 - p(t)) dl** describes much of the dynamics of the learning curve and work in general. With **N**, the number of tasks, and **p**, a measure of difficulty, both the size and complexity of a project can be modeled. While these parameters can be derived from actual learning curves. By embedding the iteration of **p(t)** into a feedback system the impact of the broader work system can also be evaluated.

The model **p(t+1) = p(t) + (1 - p(t)) dl** states that, given a measure of knowledge **p(t)**, with a trial that knowledge will likely be increased in proportion to the remaining unknown. While the constant of proportionality is on the order of **p (0)** squared. This process creates the learning curve.

It also explains:

Why initial development curves are flat. In some circumstances, the interaction of engineering and manufacturing can create a loss of control in the factory. Prediction is necessary for effective control.

Why learning curves eventually flatten. Knowledge of the process is fully gained.

How processes with a range of process knowledge can have a flatter slope. The sum of a

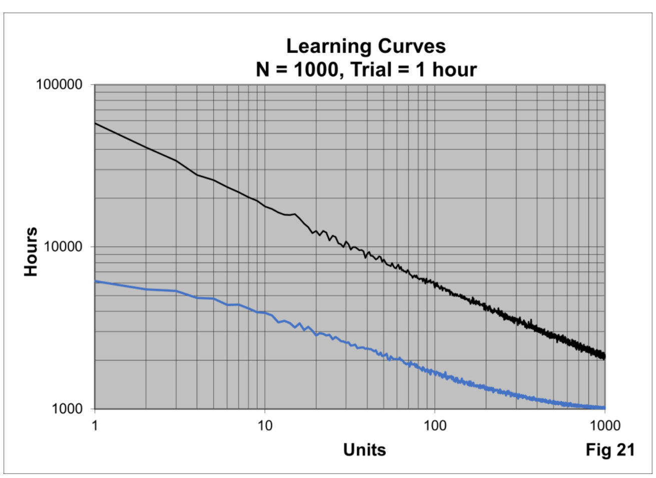


Fig 21

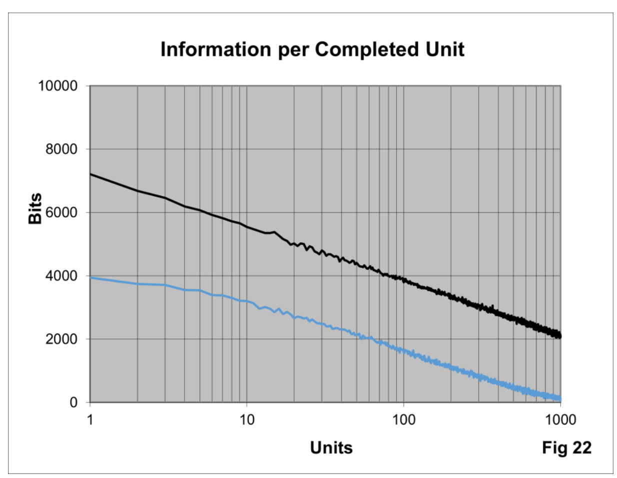


Fig 22

range of slopes creates a learning curve with a flatter slope.

The impact of design changes. The probability of success returns to **p(0**).

Why assembly slopes are generally steeper than fabrication. Feedback introduces a stream of design change into the earlier stages of manufacturing.

Estimates ultimately are intended to support a decision. Some decision criteria are a linear consequence of the cost estimate. For those, the expected value provides sufficient information. Others are options, a split of the estimates PDF, and the estimate's PDF is required. This model fulfills that requirement.

Harry Larsen *attended the University of Washington, graduating with a mathematics degree in 1966. After graduation, he was hired by the Boeing Company as a production estimator. He began building planning systems for Boeing operations on the company's new supercomputer. Many systems and twenty years later he became the Business Planning Manager of Boeing Marine Systems, a division of the Boeing Company, reporting to the division's general manager. He retired in 2002.*

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 JCAP.Editor@gmail.com

International Cost Estimating & Analysis Association

4115 Annandale Road, Suite 306 | Annandale, VA 22003

703-642-3090 | iceaa@iceaaonline.org