

Received 10 October 2023, accepted 4 November 2023, date of publication 8 November 2023,  
date of current version 13 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331112

 THEORY

# Predicting the Dynamics of Earned Value Creation in the Presence of Technical Debt

HOWARD KLEINWAKS<sup>1,2</sup>, ANN BATCHELOR<sup>1</sup>, AND THOMAS H. BRADLEY<sup>1</sup>, (Member, IEEE)

<sup>1</sup>Department of Systems Engineering, Colorado State University, Fort Collins, CO 80523, USA

<sup>2</sup>Modern Technology Solutions Inc., Alexandria, VA 22312, USA

Corresponding author: Howard Kleinwaks (howard.kleinwaks@colostate.edu)

**ABSTRACT** Technical debt, the long-term impact of decisions made to achieve a short-term benefit, has a unique impact on a project schedule. Technical debt does not impact the ability to complete the task on which it is incurred but rather impacts successor tasks causing unplanned schedule delays or budget increases. The impact of technical debt is uncertain and therefore must be modeled probabilistically. When unaccounted for and unmanaged, technical debt can build up in the project with increasing impact, eventually forcing forward progress to stop while the technical debt is remedied. Traditional project scheduling methods allow for uncertain task durations but do not provide explicit means of modeling the impacts of technical debt. Instead, they assume that each task is unaffected by the completion status of its predecessors and its duration is only dependent upon the initial estimates. This research addresses this gap by providing a novel model of the impact of technical debt on the project schedule through estimating the dynamics of value creation in the presence of technical debt. Equations are developed for estimating the probabilistic impacts of technical debt on the generation of earned value. These equations are then inverted and used to calculate task duration in the presence of technical debt and included in a Monte Carlo analysis. Comparisons are made to an existing Monte Carlo schedule analysis and technical debt impacts are explored.

**INDEX TERMS** Earned value, Monte Carlo, schedule analysis, technical debt, leading indicators of schedule performance, schedule risk assessment.

## I. INTRODUCTION

Project managers traditionally handle uncertainty by including cost and schedule margin in their project plans [1]. These margins can be used to mitigate the impact of rework and technical debt within a project. Love defines rework as the “unnecessary effort of re-doing a process or activity that was incorrectly implemented the first time” [2]. Kleinwaks, Batchelor, & Bradley define technical debt as “a metaphor reflecting technical compromises that can yield short-term benefit but may hurt the long-term health of a system” [3]. Within the context of a project, technical debt occurs when decisions made in the completion of one task negatively impact the ability to complete successor tasks on time and on budget. The impact of technical debt is not certain: the compromises made on one task may or may not impact

a successor task [4] and the compromises may proliferate throughout the system and cause significant issues [5]. Within this article, technical debt is distinguished from rework by asserting that rework is the result of the poor execution of defined processes and methods and technical debt is the result of shortcuts taken in the requirements development, design, and/or implementation in order to achieve a short-term benefit. Rework requires the repeated execution of existing process and unplanned iterations of existing tasks. Technical debt does not typically require the redoing of a specific task but instead technical debt makes completing successor tasks more complicated, costly, or time-consuming. If technical debt is not accounted for in project scheduling, then successor task duration may increase unexpectedly, resulting in late completion of tasks compared to stakeholder expectations. However, traditional schedule analysis techniques do not model changes in successor task durations based on the fidelity of predecessor task completions. This

The associate editor coordinating the review of this manuscript and approving it for publication was Zhongyi Guo<sup>1</sup>.

article provides a novel mechanism to address this gap and enable more realistic schedule assessments.

Properly assessing project schedules requires the ability to proactively predict risks associated with both technical debt and rework [6]. Monte Carlo simulation is often used to assign probabilistic durations to tasks, assuming that the task will be completed within the bounds of the assigned distribution. However, these simulations can overlook the costs associated with changes to the schedule [1] as a result of technical debt or rework [7].

Several authors have investigated the use of design structure matrices (DSM) to predict the impact of design iterations on project schedule [8], [9], [10], [11]. These techniques can be used to assess the probability of rework occurring within a project and the extensions to schedule that occur. However, they do not model the potential for technical debt. Ma, et al. [8] extend DSMs to include a probability of rework and its impact on future tasks in the context of design iterations. However, in many projects, iterations are not planned – the successor tasks must be extended or changed to address the shortcomings of the predecessor tasks. Furthermore, while modeling rework can account for project extensions, it is not the same as modeling technical debt. Rework results in repeated execution of the same tasks. Technical debt may result in longer durations of successor tasks and the potential need for unplanned effort to remove the debt from the system.

Maheswari and Varghese [11] provide a method to use DSMs to determine a project schedule accounting for overlapping tasks. By assessing the necessary condition of task overlap in a project, they demonstrate that tasks do not always abide by strict finish-to-start schedule relationships. However, their work assumes a fixed value of the overlap time and does not provide a mechanism to calculate when a task reaches that level of completion. Additionally, they assume that the work completes perfectly until the overlap time is reached without considering technical debt's task to task dependencies.

From this review, it is clear that additional techniques to handle the presence of technical debt within a project schedule are required. Failure to model technical debt can result in overly optimistic schedule estimates due to the failure to account for the cascading impact of technical debt interest. The technical debt incurred on one task can compound, impacting multiple successor tasks, resulting in significant delays and cost increases to the project.

In this article, we extend existing project schedule analysis methods to include technical debt analysis. The impact of technical debt incurred on one task on successor tasks is modeled through earned value computations. The earned value equations are inverted to estimate the duration of successor tasks subject to technical debt from their predecessors. With these equations, the impact of technical debt is then included in a Monte Carlo schedule analysis and the results compared to a traditional Monte Carlo schedule analysis. Various impacts of technical debt are explored by altering the

parameters in the analysis. This article answers the following research question:

*How can technical debt be accounted for within project scheduling activities?*

By answering this research question, this article presents a mathematical model that can be used by project managers and schedulers in Monte Carlo schedule analysis techniques. This model uses the technical debt formulation to compute increased duration of successor tasks, thereby providing a more realistic schedule analysis.

The rest of this article is structured as follows: first, overview of related work is provided. Next, the method used to account for technical debt within a schedule is described and is followed by an example application of the method within Monte Carlo schedule analysis. Finally, the results are discussed and opportunities for future work are presented.

## II. RELATED WORK

Earned value management expresses the project progress in terms of value created, where value is expressed in monetary terms. The creation of value is then used to predict both the project cost at completion and the schedule at completion through linear extrapolation of the current state [12]. While EVM is traditionally effective in cost management, its schedule management component is usually considered insufficient, especially since the schedule is expressed in cost parameters [13]. These weaknesses led to the development of earned schedule (ES) techniques [14]. ES techniques have been shown to be more accurate in predicting the schedule at completion [15] and can be more easily understood, as they measure the earned schedule in units of time (and not cost). Both EVM and ES use the same planned value and earned value curves, which take the form of an 'S-curve', shown in Figure 1. The planned value is based on the baselined project development plan, while the earned value is based on measured project progress. EVM techniques measure the difference between the two curves in the vertical direction while ES techniques measure the difference between the two curves in the horizontal direction.

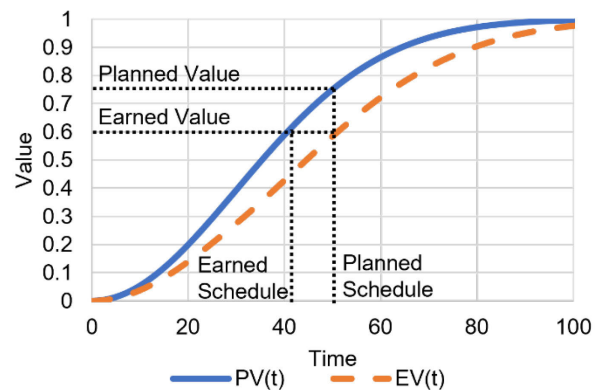


FIGURE 1. Planned and earned value 's-curves'.

Warburton [16] formulated (1) to (4) to represent planned value (PV) and earned value (EV) curves. Lowercase letters represent the instantaneous value and capital letters represent the cumulative value. Note that (4) in [16] contains an error where the negative sign on the first exponential was excluded. That error has been corrected in (4) in this article. The variables used in these equations are defined in Table 1.

$$pv(t) = \frac{Nt}{T^2} e^{-\frac{t^2}{2T^2}} \tag{1}$$

$$PV(t) = \int_0^t pv(s) ds = N \left[ 1 - e^{-\frac{t^2}{2T^2}} \right] \tag{2}$$

$$ev(t) = \begin{cases} (1-r)pv(t), & t \leq \tau \\ (1-r)pv(t) + r * pv(t-\tau), & t > \tau \end{cases}$$

$$ev(t) = \begin{cases} (1-r) \frac{Nt}{T^2} e^{-\frac{t^2}{2T^2}}, & t \leq \tau \\ (1-r) \frac{Nt}{T^2} e^{-\frac{t^2}{2T^2}} + r \frac{Nt}{T^2} e^{-\frac{(t-\tau)^2}{2T^2}}, & t > \tau \end{cases} \tag{3}$$

$$EV(t) = \begin{cases} EV_1(t) = \int_0^t ev(s) ds, & t \leq \tau \\ EV_2(t) = EV_1(\tau) + \int_\tau^t ev(s) ds, & t > \tau \end{cases}$$

$$EV(t) = \begin{cases} (1-r)N \left[ 1 - e^{-\frac{t^2}{2T^2}} \right], & t \leq \tau \\ N - N(1-r)e^{-\frac{t^2}{2T^2}} - rNe^{-\frac{(t-\tau)^2}{2T^2}}, & t > \tau \end{cases} \tag{4}$$

TABLE 1. Variables used in earned value equations.

Symbol	Definition
$N$	The number of tasks occurring in the project
$T$	The time at which the maximum value of the instantaneous planned value curve occurs
$t$	The independent time variable
$r$	Percentage of activities that require extra work
$\tau$	Delay introduced into the project due to extra work

Equation 1 defines the instantaneous planned value function. This equation models a project where the planned value achieved at each point in time, for example, work accomplished each day, initially increases until time  $T$ , which is the time at which the maximum instantaneous planned value is reached. After this point, the contributions to planned value in each time period steadily decrease. The cumulative planned value is calculated using (2). This equation, the integral of (1), produces the traditional S-curve, as shown in Figure 1. Equation 3 calculates the instantaneous earned value by assuming that a fraction of the tasks,  $r$ , are late by a time  $\tau$ , thereby delaying the accumulation of value. Equation 4 computes the cumulative earned value as the integral of the instantaneous earned value [16].

This related research forms the basis of the process for accounting for technical debt in the schedule analysis. Building off of the equations for earned value, the time at which a task reaches the necessary conditions for the successor task to start can be established. The  $r$  and  $\tau$  parameters allow for the modeling of delays introduced into a task from its predecessor tasks, a key component of technical debt. Attaching these equations to a Monte Carlo analysis allows for the modeling of the probabilistic aspects of technical debt interest.

### III. ACCOUNTING FOR TECHNICAL DEBT IN SCHEDULE ANALYSIS

Accounting for technical debt in schedule analysis starts with understanding how to measure task completion. Technical debt occurs when technical compromises are made in the execution of a task in order to achieve a short-term benefit [3]. The technical compromises may impact the scope of the task, resulting in reduced performance relative to its objectives, or in the quality of the task, resulting in lower maintainability, upgradability, sustainability, and other -ilities. These compromises may then impact the ability to complete future tasks on time, on budget, or to their performance specifications [17]. For example, technical debt is incurred when the documentation associated with a system component is reduced (technical compromise) to release on time (short-term benefit). The lack of documentation may make integration and testing of the component more time consuming and more costly (long-term impact). Kleinwaks, Batchelor, and Bradley conducted a survey on the presence of technical debt within systems engineering, concluding that, although the terminology of technical debt is not well used within systems engineering, the impacts of technical debt are substantial [17].

The size of the impact of technical debt, referred to as the interest amount within software engineering [4], is uncertain and dependent upon both the technical compromise and the interconnectedness of the task within the system context. The occurrence of the interest, defined as the interest probability [4], is uncertain – if no changes need to be made to the component carrying the technical debt, then no interest needs to be paid. Technical debt may remain hidden in a system and linger for extended periods of time, compounding the interest amount and resulting in more complicated, or even impossible, design changes.

#### A. UTILITY AS VALUE

When modeling project value for schedule analysis, the use of a monetary metric as the project value can confuse value and utility. While project duration ultimately relates to project cost, the ability of one task to begin work is not related to how much profit that predecessor task generates. Therefore, this article assumes that a value function can be formulated in terms other than financial terms [18]. Specifically, this article models value as the utility of a task to its successor tasks, where utility is measured as the completion percentage of the

predecessor task. A successor task may be able to begin work when a predecessor task is not complete (has a utility of less than one (1)), an implementation of the start-start relationship [12] of traditional project scheduling techniques. The value function is modeled as an S-curve, a relationship that has been shown to hold for task duration as well as cost [13] and which enables the time at which the task reaches a specified utility (value) to be found. Therefore, the start time of the successor tasks can be determined, leading to the calculation of the overall project duration.

**B. MODELING EARNED VALUE FROM MULTIPLE PREDECESSORS**

Modeling technical debt impacts requires the ability to determine both the interest amount and the interest probability and to account for their impacts on the value creation of a specific task. Since the interest could come from any of the predecessor tasks, it is necessary to determine the contributions to the value of a task that is derived from each of its predecessors. Adopting an S-curve formulation of the value function, modifications to Warburton’s equations can be made to calculate the earned value contributions from each predecessor task in turn. As written, Warburton’s equations assume that the earned value is contributed evenly from multiple predecessor tasks. The  $N$  parameter is used to represent the number of predecessor tasks, which changes the magnitude of the overall planned and earned value, but only in aggregation. Each predecessor task contributes the same portion of the value. This model is appropriate for planned value, which assumes perfect schedules. However, earned value, which attempts to model the actual value creation schedule, must account for the individual impacts of predecessor tasks on the earned value of the successor task. Equations 5 and 6 show the updated equations for earned value accounting for the impacts of the predecessors.  $N$  becomes a scaling variable applied evenly to all the predecessor tasks.

$$\begin{aligned}
 ev(t) &= \sum_{i=0}^n \begin{cases} (1 - r_i) \alpha_i p v_i(t), & t \leq \tau_i \\ (1 - r_i) \alpha_i p v_i(t) + r_i \alpha_i p v_i(t - \tau_i), & t > \tau_i \end{cases} \\
 ev(t) &= \sum_{i=0}^n \begin{cases} (1 - r_i) \frac{\alpha_i N t}{T^2} e^{-\frac{r^2}{2T^2}}, & t \leq \tau_i \\ (1 - r_i) \frac{\alpha_i N t}{T^2} e^{-\frac{r^2}{2T^2}} + r_i \frac{\alpha_i N t}{T^2} e^{-\frac{-(t-\tau_i)^2}{2T^2}}, & t > \tau_i \end{cases} \tag{5}
 \end{aligned}$$

$$\begin{aligned}
 EV(t) &= \sum_{i=0}^n \begin{cases} EV_{1i}(t) = \int_0^t ev_i(s) ds, & t \leq \tau_i \\ EV_{2i}(t) = EV_{1i}(\tau) + \int_{\tau_i}^t ev_i(s) ds, & t > \tau_i \end{cases} \\
 EV(t) &= \sum_{i=0}^n \begin{cases} (1 - r_i) \alpha_i N \left[ 1 - e^{-\frac{r^2}{2T^2}} \right], & t \leq \tau_i \\ \alpha_i N - \alpha_i N (1 - r_i) e^{-\frac{r^2}{2T^2}} - r_i \alpha_i N e^{-\frac{-(t-\tau_i)^2}{2T^2}}, & t > \tau_i \end{cases} \tag{6}
 \end{aligned}$$

In (5) and (6), it is assumed that each predecessor task independently impacts a portion of the successor’s task earned value. This portion is controlled by the  $\alpha$  variable, which is the percentage of the successor task’s earned value that is impacted by predecessor task  $i$ . The  $\alpha$  variables are constrained to add up to one, as shown in (7).

$$\sum_{i=0}^n \alpha_i = 1 \tag{7}$$

$\alpha_0$  is the percentage of the successor task’s earned value that is not impacted by any predecessor and can be calculated using (8).

$$\alpha_0 = 1 - \sum_{i=1}^n \alpha_i \tag{8}$$

Figure 2 depicts the contribution of multiple predecessors to the earned value of a successor task. In Case 1, each predecessor contributes equally to the earned value of Task D. In Case 2, the individual contributions are not equal, resulting in different values of  $\alpha$ . Changing the values of  $r$  and  $\tau$  for each predecessor task will change the overall earned value based on the values of  $\alpha$ , which is discussed in the next section.

**C. TECHNICAL DEBT AND EARNED VALUE**

Warburton’s equations can be used to model the impacts of technical debt interest on the system by redefining the variables  $r$  and  $\tau$ . Warburton defines  $r$  as the percentage of activities that require rework. Within the multiple predecessor and technical debt context,  $r$  is redefined as the percentage of the predecessor task’s impact on the successor task that is subject to a delay. This relationship is shown in

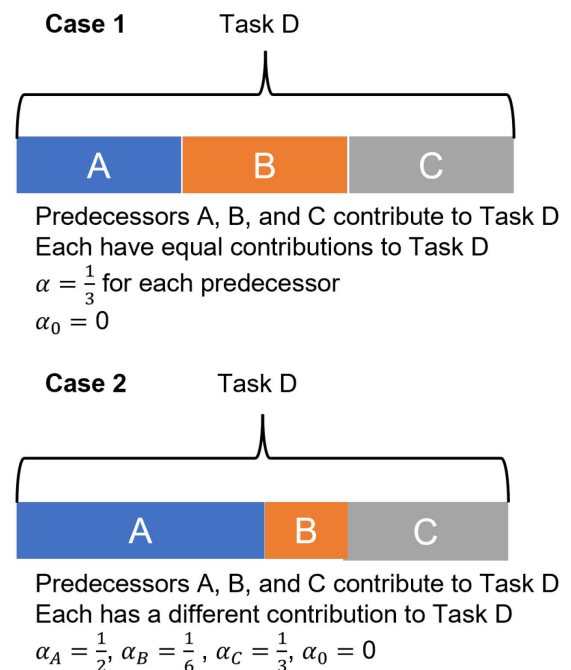
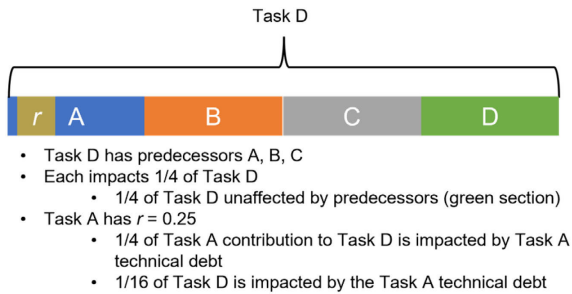


FIGURE 2. Multiple predecessor contribution to earned value.

Figure 3. In this figure,  $\alpha_A = 0.25$ : task A impacts 25% of the earned value of task D.  $r_A = 0.25$  and therefore 25% of task A's impact on task D is subject to technical debt interest from task A. Combined, 6.25% of task D is subject to delays due to technical debt interest from task A.

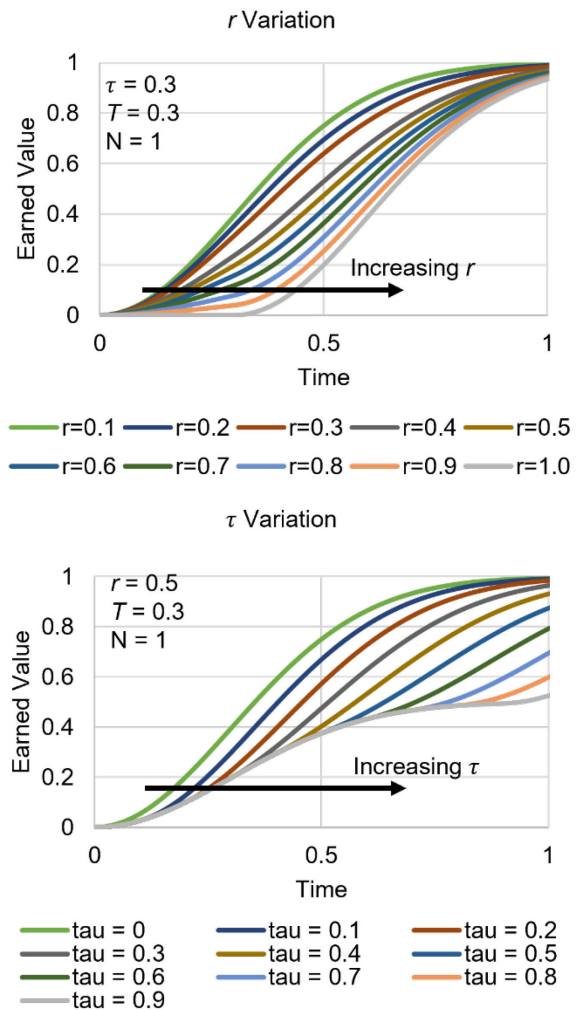


**FIGURE 3.** Definition of  $r$  parameter in the context of multiple predecessors and technical debt.

The definition of  $\tau$  is unchanged from Warburton – it is a measure of the delay introduced to the system due to technical debt interest. It measures how much longer a task takes to complete based on the technical debt introduced by a predecessor task. The impact of changing  $r$  and  $\tau$  is shown in Figure 4. Increasing  $r$  shifts the earned value curve to the right along the time axis but does not significantly change the slope – it changes the time at which the value is accumulated but not the rate. Changing  $\tau$  changes the slope of the earned value curve thereby affecting the rate of value accumulation along with the time at which the value is earned.

In terms of technical debt,  $r$  and  $\tau$ , when combined, represent the interest amount. The interest probability can be modeled through the specification of probability distributions for  $r$  and  $\tau$  and the use of Monte Carlo analysis. The impact of  $r$  and  $\tau$  and the relationship to technical debt is best understood through an example.

Williams [7] defines a schedule for the development of a test aircraft, including the expected duration for each of the tasks. This schedule will be used throughout the rest of this article as an example project. Williams provides a relevant example to technical debt through the discussion of the third management action in his aircraft example: if avionics production is delayed, then a temporary avionics kit may be installed in production aircraft. The technical compromise is to use a non-fully functional avionics kit to achieve the short-term benefit of meeting the task schedule. The long-term impact is the lack of fully functional aircraft and the potential for rework to retrofit the avionics kit. In Williams' example, 28% of the aircraft had the temporary kit installed, so  $r = 0.28$ . Williams does not provide the timeline to produce additional kits, but it is fair to estimate that it would be the same as the avionics production task and range between 12-18 months. Therefore,  $\tau$  could be estimated



**FIGURE 4.** Effect of changing  $r$  and  $\tau$  on earned value.

through a distribution that produces values in the range of 12-18 months.  $r$  and  $\tau$  would then be applied to the earned value equation for the aircraft assembly along with an estimate of the alpha value – the portion of the aircraft assembly affected by the avionics.

**D. COMPOUNDING TECHNICAL DEBT INTEREST**

One of the most pernicious qualities of technical debt is that the interest compounds. Technical debt may impact multiple successor tasks, may not appear until several successor tasks have completed, and it may grow in impact as it affects more tasks [5]. To model the compounding of technical debt interest, it is necessary to consider all the predecessor tasks as having some contribution to the earned value of the successor task. If direct predecessor tasks are the only ones considered, then there is a chance that the technical debt contribution is underestimated. For example, consider the development of a software interface with three tasks: development of the

interface control document (ICD), writing the software code, and integrating the software interface. An ICD may contain documentation debt [19], which includes the under specification of the interfaces. The software developer can take the ICD and perfectly implement it as written, and may not be aware that the interfaces were underspecified. It is not until the next task, the integration of the interface, that the technical debt in the ICD will appear, even though the ICD is not a direct predecessor of the integration task.

To model the compounding of technical debt interest, it is necessary to specify the  $\alpha$ ,  $r$ , and  $\tau$  values for each possible predecessor for every task. Figure 5 shows an example of how to specify the values using two design structure matrices (DSM), based on the aircraft project provided in Williams [7]. The dependency matrix, on the left of Figure 5, indicates the direct predecessor (value of 1) and indirect predecessor relationships (value of 2). The alpha matrix, on the right side of Figure 5, indicates the value of alpha for the relationships. These matrices are read like other DSMs, where a value in a cell indicates that the column contributes to the row. The dependencies of task 5, interim avionics, are found by reading down the column. This task has one immediate successor (task 7, assemble d/b aircraft) and several secondary successors. The value of alpha for the immediate successor is 0.1. To maintain the constraint identified in (7), the summation of the values in the rows of the alpha matrix must equal one (1). Similar DSMs could be created for  $r$  and  $\tau$ .

Dependencies																Alpha																	
ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0																	0																
1																	1																
2		1															0.4	0.6															
3		1															0.4		0.6														
4		1															0.7		0.3														
5		2	1														0.1	0.7	0.2														
6		2		1													0.1	0.7	0.2														
7		2	2	2	1	1	1										0.20	0.10	0.20	0.10	0.10	0.1											
8		1															0.7					0.3											
9		2						1									0.4					0.40	0.2										
10		2	1														0	0.7				0.3											
11		2	2								1						0	0.4				0.40	0.2										
12		2	2	2	2	2	2	1									0.20	0.2	0	0.20	0	0.10	0.1										
13		2	2	2	2	2	2	2	1								0.2	0	0.7	0	0	0	0				0	0.1					
14		2	2	2	2	2	2	2			2	1	1				0.2	0.7	0	0	0	0	0	0			0	0	0		0.1		
15		2	2	2	2	2	2	2	2	1	2	2	2	1	1		0.10	0.10	0.1	0	0	0	0	0	0.2	0	0	0	0.20	0.20	0.1		

FIGURE 5. Specification of compounding technical debt.

From these matrices, it can become clear which tasks may have larger impacts throughout the system. For example, summing the columns in the  $\alpha$  matrix will provide a total of the impact percentage of a specific task. Larger values will have higher potential for compounding technical debt interest.

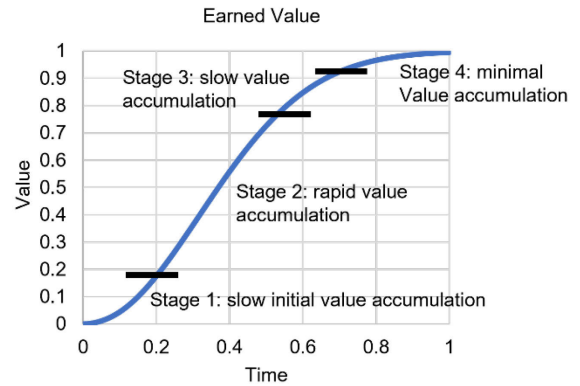


FIGURE 6. Stages of earned value S-curve.

E. CALCULATING TIME AT WHICH EARNED VALUE IS REACHED

Equation 6 models the earned value, in the presence of technical debt and multiple predecessors, as a function of time. Therefore, if this equation could be solved for  $t$ , then the time at which a specified earned value is reached could be analytically determined. However, this equation is a transcendental equation and is not analytically solvable, especially in the presence of an unknown number of predecessor tasks. Numerical techniques could be used; however, they do not lend themselves to easy application.

Examining the shape of the S-curve reveals that there are four distinct sections [20]:

- Stage 1: value accumulation starts out slowly, usually as the project is ramping up
- Stage 2: value accumulates rapidly as more resources are put into the project and work is delivered
- Stage 3: value accumulation slows down as the bulk of the work is completed and resource loading starts to reduce
- Stage 4: additional value accumulation is minimal as tasks are finalized and the project is concluded

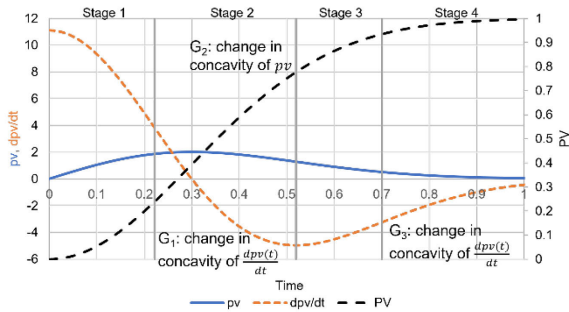
These stages are shown in Figure 6.

Between each of these stages, the concavity of the S-curve changes direction. A piece-wise linear function can be used to approximate the curve, with a separate line for each of the four sections [20]. Determining this piecewise function requires identifying the transition points between the changes in concavity.

The changes in concavity of the function are found by taking the derivatives of the function and setting those derivatives to zero. The derivatives of the earned value function are not directly solvable, due to the presence of multiple exponentials and the unknown number of predecessor tasks. However, the planned value function only contains a single exponential and does not depend on the number of predecessor tasks and therefore the transition points on the planned value curve can be found.

Figure 7 plots the cumulative planned value (PV), instantaneous planned value (pv), and the derivative of the

instantaneous planned value ( $\frac{dp}{dt}$ ). The solid black lines represent the transition points on the PV curve. The three transition points,  $G_1$ ,  $G_2$ , and  $G_3$ , can be found by applying successive derivatives of the pv curve.



**FIGURE 7.** Concavity changes indicating transition points between growth stages in planned value.

The first transition point to be found is  $G_2$  – the transition from stage 2 to stage 3 [21]. As seen in Figure 7, this transition point occurs where the concavity of pv changes. Candidates for changes in concavity are found by finding the roots of the second derivative of a function. Equation 9 shows the second derivative of the pv function. Therefore,  $G_2$  is found by solving (9) for  $t$ , which is shown in (10).<sup>1</sup> Only the positive roots are considered in this analysis.

$$\frac{d^2pv(t)}{dt^2} = \frac{N}{T^6}te^{-\frac{t^2}{2T^2}}(t^2 - 3T^2) \quad (9)$$

$$G_2 = \pm\sqrt{3}T \quad (10)$$

Transition points  $G_1$  and  $G_3$  occur when the concavity of  $\frac{dp}{dt}$  changes, as shown in Figure 7. Therefore, the second derivative of  $\frac{dp}{dt}$ , which is the third derivative of pv, needs to be found and solved. Equation 11 shows the third derivative of pv, and the solution is shown in (12). Again, only the positive roots are used in this analysis.

$$\frac{d^3pv(t)}{dt^3} = \frac{N}{T^8}e^{-\frac{t^2}{2T^2}}(t^4 - 6t^2T^2 + 3T^4) \quad (11)$$

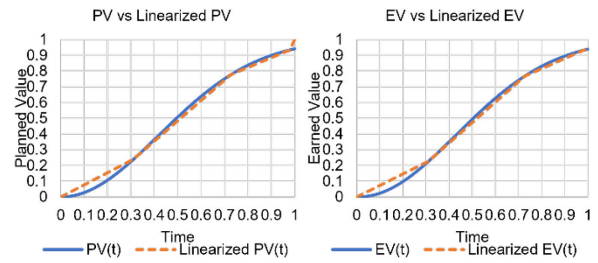
$$G_1 = \pm\sqrt{-(\sqrt{6}-3)T^2}, G_3 = \pm\sqrt{(\sqrt{6})T^2 + 3T^2} \quad (12)$$

With the transition points known, the piecewise linear equation for the planned value (LPV) can be found, as shown in (13).

$$m_1 = \frac{PV(G_1)}{G_1}$$

$$m_2 = \frac{PV(G_2) - PV(G_1)}{G_2 - G_1}$$

<sup>1</sup>Derivatives and solutions were checked using the Online Equation Solver from Wolfram Alpha, available at <https://www.wolframalpha.com/calculators/equation-solver-calculator/>



**FIGURE 8.** Linearized planned and earned value curves using the same transition points.

$$m_3 = \frac{PV(G_3) - PV(G_2)}{G_3 - G_2}$$

$$m_4 = \frac{1 - PV(G_3)}{1 - G_3}$$

$$LPV(t) = \begin{cases} m_1 * t, & t \leq G_1 \\ m_2 * (t - G_1) + PV(G_1), & G_1 < t \leq G_2 \\ m_3 * (t - G_2) + PV(G_2), & G_2 < t \leq G_3 \\ m_4 * (t - G_3) + PV(G_3), & G_3 < t \leq 1 \end{cases} \quad (13)$$

Equation 13 can be easily solved for  $t$  to determine the time at which a specific planned value occurs. Following the same process to linearize the earned value equations results in unsolvable derivative equations due to the combination of multiple exponentials and the unknown number of predecessor tasks. A possible solution is to use the transition points found in the planned value curve as the transition points of the earned value curve. The reuse of these points will induce error in the time dimension of the linearization, which needs to be characterized. The resulting linearization of planned value and earned value is shown in Figure 8. In this case, the linearized earned value plot underestimates the earned value in the stage 4 and overestimates the earned value in stage 1.

With the transition points set, the linearized earned value equations can be determined and then solved for  $t$ , as shown in (14) and (15), where  $V$  is the desired earned value. The impact of multiple predecessors is included in the linearization by using the complete earned value (EV) equation (6) at each of the transition points.

$$m_1 = \frac{EV(G_1)}{G_1}$$

$$m_2 = \frac{EV(G_2) - EV(G_1)}{G_2 - G_1}$$

$$m_3 = \frac{EV(G_3) - EV(G_2)}{G_3 - G_2}$$

$$m_4 = \frac{EV(1) - EV(G_3)}{1 - G_3}$$

$$LEV(t) = \begin{cases} m_1 * t, t \leq G_1 \\ m_2 * (t - G_1) + EV(G_1), G_1 < t \leq G_2 \\ m_3 * (t - G_2) + EV(G_2), G_2 < t \leq G_3 \\ m_4 * (t - G_3) + EV(G_3), G_3 < t \leq 1 \end{cases} \quad (14)$$

$$t = \begin{cases} \frac{V}{m_1}, V \leq EV(G_1) \\ \frac{\frac{m_1}{(V - EV(G_1))} + G_1, EV(G_1) < V \leq EV(G_2)}{\frac{m_2}{(V - EV(G_2))} + G_2, EV(G_2) < V \leq EV(G_3)} \\ \frac{\frac{m_3}{(V - EV(G_3))} + G_3, EV(G_3) < V \leq 1}{m_4} \end{cases} \quad (15)$$

$t$  represents the time at which the task reaches a particular earned value. Successor tasks may be able to start at  $t$ , however, the task is not necessarily complete at this point in time. The time of task completion is found by calculating when the earned value equals the total planned value for the task. The total planned value is input into (15) as  $V$  and then the task completion time is found. The difference between the task completion time and the original planned duration is the penalty on the task due to technical debt.

With (15), it is now possible to determine the time at which a task earns a particular value and the time at which it will finish in the presence of technical debt from multiple predecessors. The algorithm is as follows:

1. Set the values of  $\alpha$ ,  $r$ , and  $\tau$  for each predecessor task
2. Based on the value of  $T$  for the task, determine the transition points  $G_1$ ,  $G_2$ , and  $G_3$  using equations
3. Calculated the earned value at each transition point for each predecessor task using (6)
4. Given the desired earned value  $V$ , calculate  $t$  from (15)

An accuracy assessment of this method is provided in the appendix.

#### IV. APPLICATION TO MONTE CARLO SCHEDULE ANALYSIS

The prior analysis shows how to calculate the time at which a task reaches a desired earned value in the presence of technical debt. A Monte Carlo analysis can be used to determine the most likely duration of the entire project, accounting for technical debt along the way. Table 1 shows the parameters used in the analysis and recommended random and static variables. The random variables are assigned probability distributions, such as normal or triangular distributions and the accompanying distribution parameters are set as static variables. Static variables are held constant through each trial of the Monte Carlo analysis while random variables are resampled and changed with each Monte Carlo trial. Variables either apply to a singular task, such as the independent duration, or to a pair of tasks, such as  $r$  and  $U$ .

With the task duration,  $D$ , expressed as a random variable, it becomes simpler to express the time parameters

TABLE 2. Recommended random and static variables for monte carlo analysis.

Random Variables	Static Variables
$D$ – independent duration for each task	$\alpha$ – % of successor impacted by predecessor. Applies to task pairs $\alpha_0$ – % of successor that is not impacted by predecessors. Applies to each task
$r$ – % of $\alpha$ that is impacted by technical debt interest. Applies to task pairs. $r_0$ – % of $\alpha_0$ that is impacted by self-inflicted technical debt interest. Applies to each task	$U$ – earned value when successor task can start/finish (based on relationship). Applies to task pairs
$\tau$ – delay introduced into the successor task due to the technical debt interest on the predecessor task. Applies to task pairs. $\tau_0$ – self-inflicted delay on a task. Applies to each task	$T$ – the time of peak planned value. Applies to each task Earliest start time – defines the earliest that a particular task can start development Distribution parameters for $r$ , $D$ , $\tau$ will be static (e.g., mean and standard deviation for normal distributions)

( $T$ ,  $t$ , and  $\tau$ ) as percentages of the task duration, forcing them to have values between zero (0) and one (1). Setting the value of  $N$  to one (1) treats each task as a single activity. Then, the calculated earned value is the percentage of the planned value and the utility  $U$  is expressed as a percentage of planned value. This convention allows all the parameters in the Monte Carlo analysis, with the exception of the task duration to be on the same scale, from 0 to 1. It also enables automatic adjustments of the technical debt delay based on the duration of the task.  $\tau$  is expressed as the percentage of the successor task duration and therefore adjusts with the random selection of the task duration in the Monte Carlo analysis. The actual task duration is found by multiplying the time at which the utility threshold is reached by the duration. This method is shown in the following example.

Williams [7] performed a Monte Carlo analysis for a new airplane development project, including modeling management actions. The tasks, their sequence, and the parameters for the distributions of the task durations are shown in Figure 9. This analysis will serve as a test case for the method presented in this article, including updating the analysis to account for technical debt.

Williams assessed the project duration, found by determining the completion time of the ‘Ready to assemble’ task, in two cases: the baseline case which only uses the distribution of the durations and a case that represents the application of management actions that cause impacts to subsequent tasks such as “downstream quality issues” [7]. These impacts can



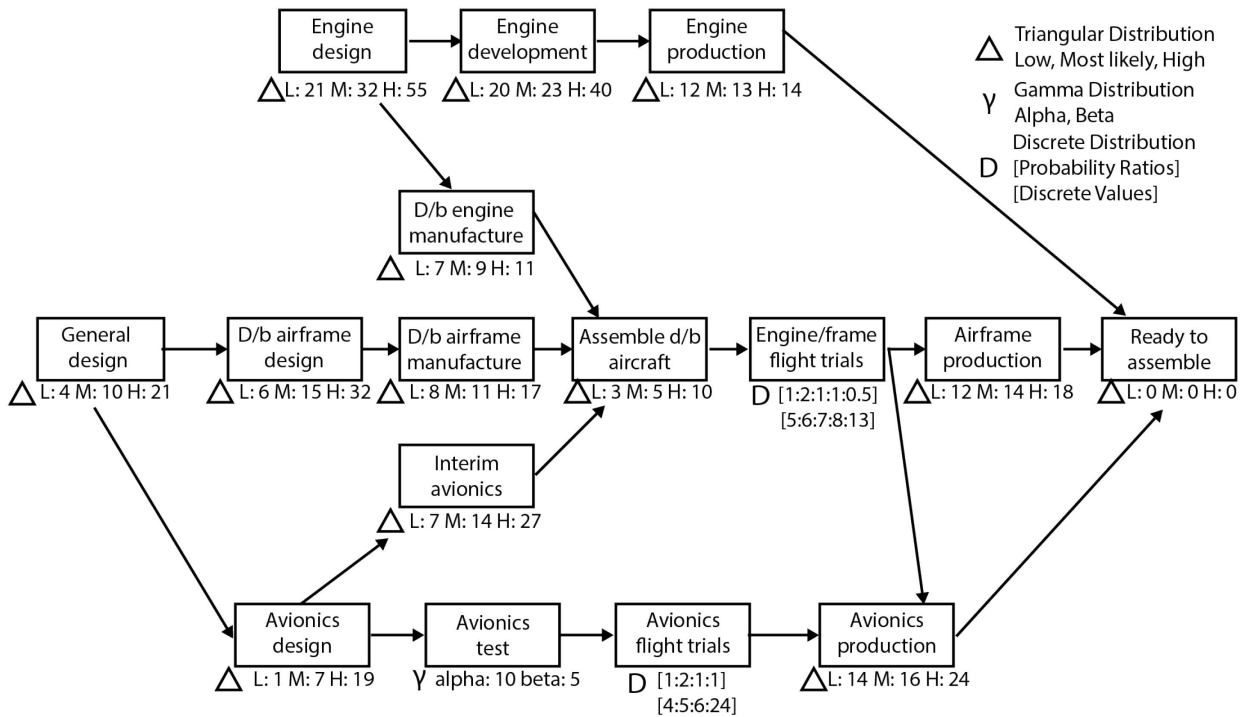


FIGURE 9. Project tasks, durations, and sequence, adapted from [7].

be interpreted as technical debt. Table 2 compares the mean duration of the project and the 90% point (the time at which 90% of the Monte Carlo trials show completion of the project) provided by Williams with those calculated using the method presented in this article. The parameters used in this method are also provided for each case. Since planned value curves for each task were not provided by Williams, the value of  $T$  used for all tasks was iteratively determined by running the algorithm with different values until results comparable to Williams was achieved. In cases where the planned value curves of each task are known,  $T$  would be determined as the point of maximum instantaneous planned value as defined by Warburton [16].

As can be seen in Table 2, the new method provides answers that are similar to those provided by Williams. Of note is that a custom distribution for duration had to be applied to account for the management actions associated with expediting the engine development to better map to the method used by Williams. The closeness of the results lends confidence to the baseline algorithms presented in this article.

### A. IMPLEMENTATION

The equations described in the previous sections can be implemented as part of a Monte Carlo schedule analysis. The algorithm requires the user to specify the task duration and technical debt parameters. Static variables, as defined in Table 1, have their specific values defined. Random variables,

as defined in Table 1, have the parameters of their associated probability distributions set. For this algorithm, it is assumed that the sequence of tasks is known. The algorithm is defined as follows:

1. Define the sequence of tasks and establish the predecessor-successor relationships
2. Define the parameters  $\alpha$ ,  $T$ , and  $U$
3. Define the distribution parameters for  $D$ ,  $r$ , and  $\tau$
4. For each trial in the Monte Carlo analysis:
  - a. Randomly set all values of  $D$ ,  $r$ , and  $\tau$  using the supplied distributions
  - b. For each task:
    - i. Determine the earned value at the transition points using (6)
    - ii. For all predecessors:
      1. Calculate  $t_u$ , the time at which the earned value threshold,  $U$ , is reached using (15). This value will be between zero and one
      2. Calculate the actual task duration,  $td$ , by multiplying  $t_u$  times  $D$
      3. Calculate the predecessor end time as predecessor start time plus  $t_u$
      4. Set the task start time to the maximum predecessor end time
    - c. Determine the completion time as the end time of the last task
5. Average the results of the Monte Carlo analysis to produce the results

TABLE 3. Comparison of results with [7].

Case	Values from [7]	Calculated Values	Parameters Used
Baseline analysis	Mean Duration: 90.5 90% Value Duration: 103	Mean Duration: 90.5 90% Value Duration: 100.5	$T = 0.395$ (all tasks) $U = 1$ (all tasks) $r, r_0 = 0$ (constant value, all tasks) $\tau, \tau_0 = 0$ (constant value, all tasks) $\alpha_0 = 1$ (all tasks) $\alpha = 0$ (all tasks)
Expediting engine design	Not provided	Mean Duration: 88.1 90% Value Duration: 95.7	Same as previous case except: <b>Engine Design</b> Duration: custom triangular distribution where the value is reduced by 1/3 if it exceeds 34, per [7] <b>d/b Engine Manufacture</b> for the dependency on engine design: $\tau$ : Uniform Distribution between [0, 0.2] $r$ : Normal Distribution with $\mu = 1, \sigma = 0$ $\alpha: 0.5$ $\alpha_0 = 0.5$ <b>Engine Development</b> for the dependency on engine design: $\tau$ : Uniform Distribution between [0, 0.05] $r$ : Normal Distribution with $\mu = 1, \sigma = 0$ $\alpha: 0.5$ $\alpha_0 = 0.5$
Increased parallelism	Mean Duration: 87.2 90% Value Duration: 95	Mean Duration: 88.2 90% Value Duration: 95.6	Same as previous case except: <b>Engine Production</b> Add a dependency on engine flight trials with the following parameters: $T = 0.395$ $U = 0.2$ $r = 0$ $\tau = 0$ $\alpha = 1/3$ For the dependency on engine development set: $\tau$ : Uniform Distribution between [0, 0.1] $r$ : Normal Distribution with $\mu = 1, \sigma = 0$ $\alpha: 1/3$ $\alpha_0 = 1/3$

V. DISCUSSION

Using the same example project provided in [7], the impact of technical debt and increased parallelism on the project schedule can be assessed by rerunning the Monte Carlo analysis for conditions assessing both technical debt and increased parallelism. Starting with the baseline analysis case, two different technical debt conditions were run: low technical debt and high technical debt. In the high technical debt case, the technical debt is assumed to affect a higher portion of the successor task and with a larger impact – both  $r$  and  $\tau$  are higher. The distributions used are listed in Table 3. The values for alpha were set using the values shown in Figure 5. The increased parallelism case sets the values for  $U$  to 0.9 for all task dependencies, indicating that a task

can start once all of its predecessors have reached at least 90% of their earned value. Figure 10 shows the cumulative distribution function for each of the cases. Note that it is possible to calculate durations that are of extreme length due to the probabilistic analysis. Outliers were defined as total project durations above 200 months and these outliers were removed from the results.

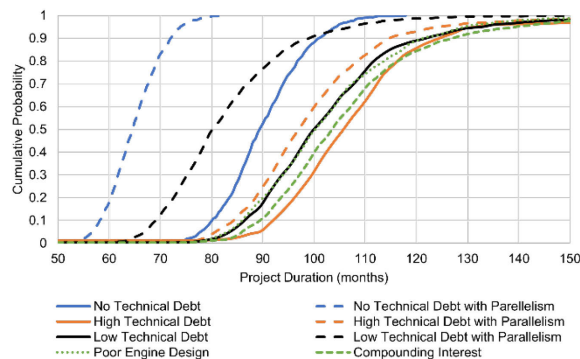
A. IMPACT OF INCREASED PARALLELISM ON PROJECT SCHEDULE

The third column and the second through fourth rows in Table 3 show the impact of assuming that tasks can start when their predecessors reach at least 90% of their value. Evaluating the start time of successor tasks based on the

**TABLE 4. Technical debt and increased parallelism impacts on the airplane project.**

Technical Debt Condition	Case	
	No Parallelism ( $U = 1$ )	Increased Parallelism ( $U = 0.9$ )
No Technical Debt: $r$ : Normal Distribution with $\mu = 0, \sigma = 0$ $\tau$ : Normal Distribution with $\mu = 0, \sigma = 0$	Mean duration 90.5 90% duration 101.0	Mean duration 65.2 90% duration 72.1
Low Technical Debt: $r$ : Normal Distribution with $\mu = 0.2, \sigma = 0.05$ $\tau$ : Normal Distribution with $\mu = 0.2, \sigma = 0.05$	Mean duration 92.0 90% duration 102.0	Mean duration 68.9 90% duration 76.9
High Technical Debt: $r$ : Normal Distribution with $\mu = 0.5, \sigma = 0.2$ $\tau$ : Normal Distribution with $\mu = 0.5, \sigma = 0.2$	Mean duration 108.2 90% duration 123.5	Mean duration 100.7 90% duration 114.5
	No compounding interest	Compounding interest
Low Technical Debt, Compounding Interest: $U = 1$ for all task $R$ and $\tau$ : same as the low technical debt case except for engine design task. All dependencies on engine design have the following distributions $r$ : Normal Distribution with $\mu = 0.5, \sigma = 0.2$ $\tau$ : Normal Distribution with $\mu = 0.8, \sigma = 0.1$	Mean duration 104.4 90% duration 123.2	Mean duration 108.0 90% duration 128.4

accumulation of value can significantly decrease the subsequent start time of each task and therefore decrease the overall project duration. Conceptually, this conclusion follows from the evaluation of an earned value curve, such as shown in Figure 1, where accumulating the last 10% of the project value can take over 20% of the time. This last 10% of value often does not add value to the successor tasks, and therefore, by starting earlier, the entire project can be accelerated. For example, a software interface between two separate components is typically defined by an interface control document (ICD). To start developing the software interface, it is necessary to have the majority of the ICD complete, but the final version, which may include non-technical aspects such as formatting and obtaining signatures, is not required.



**FIGURE 10. Cumulative probabilities of completing the aircraft project under various technical debt and parallelism assumptions.**

**B. IMPACT OF TECHNICAL DEBT ON PROJECT SCHEDULE**

The third and fourth rows in Table 3 show the impact of technical debt on the project. In both cases, the mean duration of the project increased when technical debt is assumed to occur on each task. In the ‘high technical debt’ case increasing the parallelization is not sufficient to bring the schedule back to the original baseline. These results model the impact that technical debt can have on a schedule and highlight one of the deficiencies of traditional Monte Carlo schedule analysis. Every task carries some risk of creating technical debt for its successor tasks, either through design or implementation deficiencies or through a change in the context of the system. Traditional methods add margin for the duration of impacted tasks without actually assessing the downstream impacts. The method presented in this article allows for the project manager to assess both increases in task duration and different levels of impact through setting the distribution and technical debt parameters. By varying these assumptions on individual tasks, the project manager can determine which tasks carry the largest risk associated with technical debt. Evaluating these risks allows a project manager to determine the likelihood that a task moves onto the critical path due to technical debt.

**C. IMPACT OF COMPOUNDING TECHNICAL DEBT INTEREST**

The last row of Table 3 shows the impact of compounding technical debt interest. In this scenario, the tasks all demonstrate low technical debt, except for the engine design task. The engine design task is modeled as completing with exceptionally high technical debt, resulting in high values of  $r$  and  $\tau$ . In the second column of Table 3, it is assumed that the technical debt interest does not compound, and that the technical debt accrued in the engine design task only affects its direct successor. In the third column of Table 3 the technical debt from the engine design task affects all of the possible successors. The results show that compounding the technical debt interest causes increased delays to the project: a 3.5% increase in mean project duration and a 4.2% increase in the 90% point. Figure 11 shows the

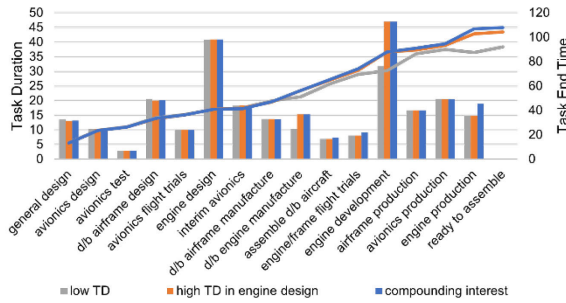


FIGURE 11. Effect of compounding interest on task duration and end time.

average duration and completion time for each task in the low technical debt case with no parallelism, the low technical debt case with a high technical debt engine design and no compounding interest, and the compounding interest case. The task with high technical debt, engine design, does not suffer from significant delays. The technical debt of the engine design impacts the d/b engine manufacture task and the engine development task directly. In the compounding case, additional delays are seen in the engine production, assemble d/b aircraft, and engine/frame trials since the additional dependencies on the technical debt from the engine design are modeled.

D. QUANTIFYING TECHNICAL DEBT INTEREST

As defined in (6), this method allows for the quantification of technical debt interest. The technical debt interest amount is represented by  $r$  and  $\tau$  and the interest probability is represented through the distribution parameters selected for the Monte Carlo analysis. For each task, the interest amount can be evaluated by assessing the delay in task completion due to the technical debt of the task predecessors. Using the normalized parameter representation, the task is complete when the earned value reaches a value of one (1). This time,  $t_c$ , can be found by calculating  $t$  using (15), with  $V = 1$ . The interest amount,  $i_A$ , is expressed as a percentage of the task duration and is calculated using (16):

$$i_A = t_c - 1 \tag{16}$$

$i_A$  can be multiplied by the task value to convert it to the value units. If this value is also tracked through the Monte Carlo analysis, then the results of the analysis can be used to predict the expected value of the technical debt interest. Figure 12 shows the cumulative probability of the interest amount for the ‘engine/frame flight trials’ task for the low technical debt with no parallelism, the high technical debt with no parallelism, and the compounding interest cases found in Table 3. This task depends on several other tasks with both primary and secondary dependencies, as seen in Figure 9.

The low technical debt case has a small standard deviation and does not compound the interest; therefore, the predicted interest amount is relatively static. The compounding interest

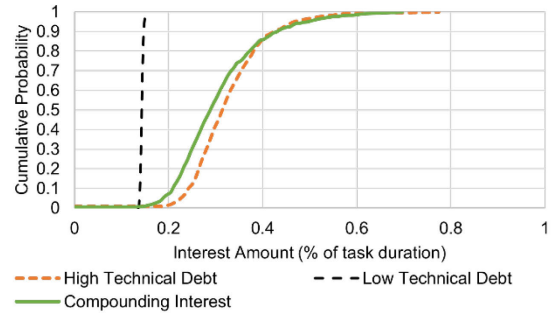


FIGURE 12. Interest amount for ‘engine/frame flight trials’.

case, which propagates the effects of a single task with large technical debt, incurs close to the same level of interest as the high technical debt case, where all tasks carry technical debt. This result highlights how technical debt can permeate through the system – a single task can cause cascading delays throughout the rest of the project.

E. COMPARISON TO EXISTING METHODS

Using the equations and processes defined in this article, it is possible to model increases to the durations of successor tasks based on technical debt introduced in predecessor tasks. This technique is important to schedule analysis as it highlights which tasks need more effective process control methods to prevent the entire project from being delayed.

Compared to existing methods of Monte Carlo schedule analysis, the method presented in this article adds additional capability to evaluate technical debt and its impacts. This method leverages the existing approaches and adjust the duration calculation for each task based on the technical debt parameters. While requiring a larger upfront investment of effort to determine the parameters, the method adds minimal runtime to the analysis, yet produces leading indicators for the project manager.

VI. LIMITATIONS AND FUTURE WORK

While providing a novel approach to including technical debt contributions in a Monte Carlo schedule analysis, this work is not without its limitations, which can be explored through future efforts. This work assumes that the technical debt parameters remain constant between predecessor and successor task pairs. However, it is likely that the potential impact of technical debt could change based on the state of the predecessor task. This dynamic model could be implemented in future versions of the algorithm. The linearization of the earned value equations introduce error into the analysis, as shown in Appendix A. These equations can be refined and better solutions found to reduce the error. Finally, the major limitation in the work is reliability of the input parameters and estimates. In any schedule analysis, the output is only as good as the original estimates. The same principle holds with this approach – the overall fidelity of the assessment is based on the accuracy of the input task durations and

technical debt parameters. Future work can explore relationships between different task types to established guidelines for the parameters to be used. Additional future work includes verification and validation of the method through application to real project development. These applications will reveal the success of the method in predicting technical debt impacts and the cost-benefit tradeoff of early introduction of technical debt reduction efforts.

**VII. CONCLUSION**

Monte Carlo schedule analysis provides a probabilistic estimate of the duration for completing a project. However, traditional techniques do not consider the impact of the quality of each task on the ability to complete the successor task on schedule. They also tend to assume finish-to-start relationships, which do not accurately represent task sequencing, especially in high level schedules. This article provides a novel method to assess the technical debt of each task and its impact on successors by modeling technical debt contributions and impacts on successor tasks. It also allows for the modeling of relationships where a task starts once its predecessor reaches a specified percentage of its final value. This combination allows for more accurate schedule modeling early in projects based on real world conditions and for the inclusion of technical debt effects. By estimating technical debt impacts on successor tasks, the project manager has the ability to evaluate leading indicators of future delays. Leading indicators provide project managers with time to implement corrective actions, such as increased quality control, while the cost to do so is low. Regularly updating the schedule analysis based on the evaluated technical debt of tasks in progress can identify the risk of delays to future tasks, and therefore the entire project. Identification of these risks enable project managers to introduce proper mitigation strategies before the risks become issues.

**APPENDIX A  
ACCURACY ASSESSMENT**

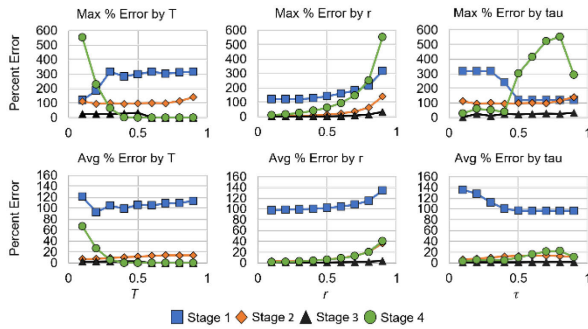
Given the piecewise nature of the linearization function, it is beneficial to look at the accuracy in each of the four sections. An exhaustive analysis was done examining the linearized earned value functions for values of  $T$ ,  $r$ , and  $\tau$  for the single predecessor case. All three parameters were varied from 0.1 to 0.9 in steps of 0.1. For all cases,  $N = 1$  to enable consistent scaling. The maximum absolute error and the maximum percent error were calculated for each of the four linearization stages for each combination of input parameters. The maximum and average values found are shown in Table 4, showing that while the percent errors are large in some cases, the absolute errors are of similar magnitudes for each case. Therefore, the linearization can be considered a valid approximation to the true function.

The earned value function itself is piecewise, changing equations when  $t = \tau$ . Therefore, rows have been added to Table 4 showing the results for the cases where  $t \leq \tau$  and where  $t > \tau$ . The largest percent error values are for

**TABLE 5. Accuracy assessment of earned value linearization.**

Output Parameter	Input condition	Stage 1	Stage 2	Stage 3	Stage 4
Maximum absolute error	All	0.057	0.079	0.034	0.579
Maximum percent error	All	318%	141%	33%	554%
Average absolute error	All	0.032	0.022	0.011	0.095
Average percent error	All	107%	11%	3%	25%
Maximum absolute error	$t \leq \tau$	0.052	0.025	0.016	0.042
Maximum percent error	$t \leq \tau$	187%	5%	2%	5%
Average absolute error	$t \leq \tau$	0.028	0.014	0.009	0.021
Average percent error	$t \leq \tau$	98%	4%	2%	2%
Maximum absolute error	$t > \tau$	0.057	0.079	0.034	0.579
Maximum percent error	$t > \tau$	318%	141%	33%	554%
Average absolute error	$t > \tau$	0.049	0.026	0.013	0.098
Average percent error	$t > \tau$	148%	15%	3%	26%

stage 1. This section of the linearization curve applies when the calculated earned values are small which can lead to large discrepancies in percent error. The magnitude of the absolute error, while higher than the other sections, is still in the same general range. Figure 13 plots the maximum and average percent errors for each analyzed value of  $T$ ,  $r$ , and  $\tau$ . From these plots, it can be clearly seen that large values of  $r$  (center plots) consistently lead to higher error values, while the largest values of the other parameters do not exhibit consistent behavior. Therefore, it can be inferred that the  $r$  parameter drives the errors when it gets large. The impact of  $r$  is to shift the earned value plot to the right. Large values decrease the similarity that was assumed when



**FIGURE 13.** Maximum and average percent error of linearization of earned value sliced by  $T$ ,  $r$ , and  $\tau$ .

reusing the inflection points from the planned value curve. Figure 13 shows that the linearization accuracy is within 10% on average for the final three linearization stages when  $T$ ,  $r$ , and  $\tau$  are all less than or equal to 0.5. Note that values of zero in the plot indicate cases that were not realized. For example, high values of  $T$  did not enter the limited growth phase in the cases tested.

Although the linearization produces some areas of large percent error, these errors are low in absolute magnitude. Additionally, these errors are likely to be smaller than any errors introduced through the initial estimation of the task duration. Therefore, it can be concluded that the linearization does not cause a significant impact on the overall accuracy of the schedule assessment.

The transition points in the linearization are controlled by the value of  $T$ .  $T$  changes as the shape of the planned value curve changes and therefore the transition points will change. As seen in the first column of Figure 13, the percent error in the analysis is relatively constant across different values of  $T$ , except for the first and last stage. Therefore, values of  $T$  that produce longer first or last stages would produce additional errors.

## APPENDIX B COMPUTATION ENVIRONMENT

The Monte Carlo analysis in this article was conducted using Python 3.9.7 scripts executed within the Spyder integrated development environment (version 5.1.5). The software was executed on a Dell Vostro 15 7510 computer running 64-bit Windows 11 Pro with a dual 2.30 GHz 11<sup>th</sup> Gen Intel® Core™ i7-11800H processor and 16.0 GB of RAM. All cases in this article were executed for 1000 trials and the execution time was between 2.3 and 2.6 seconds.

## REFERENCES

- [1] P. E. D. Love, F. Ackermann, J. Smith, Z. Irani, and D. J. Edwards, "Making sense of rework causation in offshore hydrocarbon projects," *Project Manage. J.*, vol. 47, no. 4, pp. 16–28, Aug. 2016.
- [2] P. E. D. Love, "Influence of project type and procurement method on rework costs in building construction projects," *J. Construct. Eng. Manage.*, vol. 128, no. 1, pp. 18–29, Feb. 2002.

- [3] H. Kleinwaks, A. Batchelor, and T. H. Bradley, "Technical debt in systems engineering—A systematic literature review," *Syst. Eng.*, vol. 26, no. 5, pp. 675–687, Sep. 2023.
- [4] C. Seaman, Y. Guo, N. Zazworka, F. Shull, C. Izurieta, Y. Cai, and A. Vetro, "Using technical debt data in decision making: Potential decision approaches," in *Proc. 3rd Int. Workshop Manag. Tech. Debt (MTD)*, Jun. 2012, pp. 45–48.
- [5] A. Martini and J. Bosch, "The danger of architectural technical debt: Contagious debt and vicious circles," in *Proc. 12th Work. IEEE/IFIP Conf. Softw. Archit.*, May 2015, pp. 1–10.
- [6] R. L. Yim, J. M. Castaneda, T. L. Doolen, I. Y. Tumer, and R. Malak, "Exploring the relationship between rework projects and risk indicators," *Project Manage. J.*, vol. 46, no. 4, pp. 63–75, Aug. 2015.
- [7] T. Williams, "Why Monte Carlo simulations of project networks can mislead," *Project Manage. J.*, vol. 35, no. 3, pp. 53–61, Sep. 2004.
- [8] G. Ma, J. Jia, T. Zhu, and S. Jiang, "A critical design structure method for project schedule development under rework risks," *Sustainability*, vol. 11, no. 24, p. 7229, Dec. 2019.
- [9] V. Krishnan, S. D. Eppinger, and D. E. Whitney, "A model-based framework to overlap product development activities," *Manage. Sci.*, vol. 43, no. 4, pp. 437–451, Apr. 1997.
- [10] R. P. Smith and S. D. Eppinger, "A predictive model of sequential iteration in engineering design," *Manage. Sci.*, vol. 43, no. 8, pp. 1104–1120, Aug. 1997.
- [11] J. U. Maheswari and K. Varghese, "Project scheduling using dependency structure matrix," *Int. J. Project Manage.*, vol. 23, no. 3, pp. 223–230, Apr. 2005.
- [12] *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 6th ed., Newtown Square, PA, USA: Project Manag. Inst., 2017.
- [13] H. Khamooshi and H. Golafshani, "EDM: Earned duration management, a new approach to schedule performance management and measurement," *Int. J. Project Manage.*, vol. 32, no. 6, pp. 1019–1041, Aug. 2014.
- [14] W. Lipke, "Schedule is different," *Measurable News*, vol. 31, no. 4, pp. 1–29, Mar. 2003.
- [15] S. Vandevoorde and M. Vanhoucke, "A comparison of different project duration forecasting methods using earned value metrics," *Int. J. Project Manage.*, vol. 24, no. 4, pp. 289–302, May 2006.
- [16] R. D. H. Warburton, "A time-dependent earned value model for software projects," *Int. J. Project Manage.*, vol. 29, no. 8, pp. 1082–1090, Dec. 2011.
- [17] H. Kleinwaks, A. Batchelor, and T. H. Bradley, "An empirical survey on the prevalence of technical debt in systems engineering," *INCOSE Int. Symp.*, vol. 33, no. 1, pp. 1640–1658, Aug. 2023.
- [18] T. R. Browning, "Planning, tracking, and reducing a complex project's value at risk," *Project Manage. J.*, vol. 50, no. 1, pp. 71–85, Feb. 2019.
- [19] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," *J. Syst. Softw.*, vol. 101, pp. 193–220, Mar. 2015.
- [20] K. Fordyce. (Apr. 1, 2020). *Some Basics on the Value of S Curves and Market Adoption of a New Product*. Arkieva. Accessed: Mar. 26, 2023. [Online]. Available: <https://blog.arkieva.com/basics-on-s-curves/>
- [21] E. W. Weisstein. *Inflection Point From MathWorld—A Wolfram Web Resource*. Accessed: Mar. 26, 2023. [Online]. Available: <https://mathworld.wolfram.com/InflectionPoint.html>
- [22] F. Caron, F. Ruggeri, and A. Merli, "A Bayesian approach to improve estimate at completion in earned value management," *Project Manage. J.*, vol. 44, no. 1, pp. 3–16, Feb. 2013.
- [23] Y. H. Kwak and F. T. Anbari, "History, practices, and future of earned value management in government: Perspectives from NASA," *Project Manage. J.*, vol. 43, no. 1, pp. 77–90, Feb. 2012.
- [24] L. Leach, "Schedule and cost buffer sizing: How to account for the bias between project performance and your model," *Project Manage. J.*, vol. 34, no. 2, pp. 34–47, Jun. 2003.
- [25] S. Blumberg, R. Das, J. Lansing, N. Motsch, B. Munstermann, and R. Patenge, "Demystifying digital dark matter: A new standard to tame technical debt," McKinsey Digital, Jun. 2022. Accessed: Mar. 26, 2023. [Online]. Available: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/demystifying-digital-dark-matter-a-new-standard-to-tame-technical-debt>

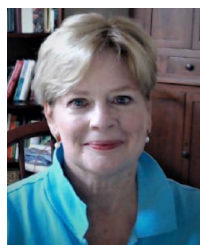


**HOWARD KLEINWAKS** received the B.Sc. and M.Sc. degrees in aerospace engineering from MIT. He is currently pursuing the Doctor of Engineering degree in systems engineering with Colorado State University, Fort Collins, CO, USA. He conducts research on the management of technical debt in iterative systems development. He is also the Program Director with the Strategic Space Business Unit, Modern Technology Solutions Inc. (MTSI), and has over 15 years of experience in the aerospace and systems engineering fields. He is a Project Management Professional (PMP), an Associate Systems Engineer (ASEP), and a Professional Scrum Master.



**THOMAS H. BRADLEY** (Member, IEEE) received the B.S. degree in mechanical engineering from the University of California at Davis, and the Ph.D. degree in mechanical engineering from the Georgia Institute of Technology. He is a Woodward Foundation Professor and the Department Head of the Department of Systems Engineering, Colorado State University. He conducts research and teaches a variety of courses in system engineering, multidisciplinary optimization, and design. His research interests include applications in automotive and aerospace system design, energy system management, and lifecycle assessment. He is a member of INCOSE, SAE, ASME, and AIAA.

...



**ANN BATCHELOR** is a Professor of systems engineering with Colorado State University, teaching engineering project and program management, systems requirements engineering, and engineering risk assessment. She has more than 20 years of industrial experience in technical management, systems engineering, production, manufacturing, lean engineering, life cycle management, test and analysis, proposal and project management, and technical writing. She has held roles, including a chief scientist, a systems engineer, and the director of engineering and program management. She is a past Certified Program Management Professional (PMP), a Military Sensing Fellow (DOD Informational and Analysis Center for Military Sensing), and a former President-Elect of the INCOSE Atlanta Chapter.