Shane K. Curtis
Research Assistant
e-mail: shanekcurtis@gmail.com

Stephen P. Harston
Ph.D. Candidate
e-mail: sharston@gmail.com

Christopher A. Mattson[1]
Associate Professor
e-mail: mattson@byu.edu

Department of Mechanical Engineering,
Brigham Young University,
Provo, UT 84602

Characterizing the Effects of Learning When Reverse Engineering Multiple Samples of the Same Product

Reverse engineering is the process of extracting information about a product from the product itself. An estimate of the barrier and time to extract information from any product is useful for the original designer and those reverse engineering, as both are affected by reverse engineering activities. The authors have previously presented a set of metrics and parameters to estimate the barrier and time to reverse engineer a product once. This work has laid the foundation for the developments of the current paper, which address the issue of characterizing the reverse engineering time and barrier when multiple samples of the same product are reverse engineered. Frequently in practice, several samples of the same product are reverse engineered to increase accuracy, extract tolerances, or to gather additional information from the product. In this paper, we introduce metrics that (i) characterize learning in the reverse engineering process as additional product samples are evaluated and (ii) estimate the total time to reverse engineer multiple samples of the same product. Additionally, an example of reverse engineering parts from a control valve is introduced to illustrate how to use the newly developed metrics and to serve as empirical validation. [DOI: 10.1115/1.4007918]

Keywords: reverse engineering, barrier to reverse engineering, product imitation, time estimation

1 Introduction and Literature Survey

Reverse engineering, or the process of extracting information about a product from the product itself, is a common industry practice that has a noticeable impact on the global market. For many companies, reverse engineering is synonymous with competitive benchmarking—a legitimate and profitable business norm. For other companies, however, reverse engineering represents a real threat to their competitive advantage, as imitators can use reverse engineering to create imitation products or to extract valuable trade secrets [1–5]. As a result, reverse engineering can potentially inhibit growth and innovation [6]. To combat this effect, companies can design products that are harder to reverse engineer, which reduces the incentive for competitors to imitate their products [7,8]. Therefore, strategic design approaches that increase the barrier or time [9,10] to reverse engineer a product are worth developing and can have a significant, positive impact on a product's return on investment [11].

The authors have previously presented a set of metrics and parameters to estimate the barrier and time to reverse engineer a product *once* [12]. However, frequently in industry multiple samples of the same product are reverse engineered to increase data accuracy, gather statistical information, estimate tolerances, or gather additional information that was initially overlooked. Much like other repetitive tasks, the time required to reverse engineer any product sample after the first is likely to decrease due to learning that occurs from sample to sample. Thus, the time predicted by the previous metrics, if multiplied by the number of times that a product is reverse engineered, will overestimate the total time to reverse engineer that product because the previous metrics do not account for the effects of learning. The principle purpose of this paper is to present additional metrics that enable a design engineer to estimate the time and barrier to reverse engineer multiple samples of the same product by characterizing and accounting for learning that occurs when reverse engineering.

As stated above, one purpose for reverse engineering multiple samples of the same product is to improve data accuracy. When one product is reverse engineered, the part, in most cases, is just a single member of a distributed population, where variation is undoubtedly present [13]. The precision of measuring devices, such as digital calipers or coordinate measuring machines, also introduces uncertainty into extracted product data [14]. As a result, an appropriate statistical analysis needs to be performed in order to test hypotheses on the true nominal values of information contained by a product. This involves determining an adequate product sample size to be reverse engineered based on a predetermined confidence level and acceptable error [15]. As the number of available parts for the sample size increases, so does the accuracy of the extracted data [16].

Another purpose for reverse engineering multiple samples of the same product is to extract geometric tolerance data. Reverse engineering with the intent of reconstructing a product for future manufacturing is incomplete until tolerances are allocated to the product. If the dimensions of a product vary more than the allowable tolerances, then the probability of the product failing to assemble or function correctly increases. This will inevitably lead to costly repairs, poor performance, and dissatisfied customers, all of which diminish the product's effectiveness [17]. Optimally allocating tolerances is a typical, yet challenging task in engineering design. An overview of the many methods used to allocate tolerances when designing a product can be found in Ref. [18]. When reverse engineering, the process for allocating tolerances becomes more difficult [19], as it requires a significant amount of skill and experience to match the original tolerances of a product. As a consequence, various methods have been presented in the literature to help approximate dimensional and geometric tolerances when reverse engineering [20–22]. One in particular involves

[1]Corresponding author.

Contributed by the Design Innovation and Devices of ASME for publication in the Journal of Mechanical Design. Manuscript received November 1, 2011; final manuscript received September 12, 2012; published online November 21, 2012. Assoc. Editor: Janis Terpenny.

performing dimensional analysis on multiple samples of the same product and comparing the results to discover possible manufacturing variations as an aid to establishing tolerances [22].

Regardless of the reason for reverse engineering multiple samples of the same product, the person or team reverse engineering is likely to learn as they repeat the reverse engineering process. Learning can be defined here as *change in behavior that occurs as a result of experience* [23] and is typically accompanied by improved performance [24]. This improvement in performance, and the rate at which it occurs, has led several researchers to investigate the learning curve for various industrial settings and determine the factors that influence learning [25,26]. However, research has not been published regarding learning during reverse engineering.

In this paper, we introduce parameters and metrics that (i) characterize a person's learning capability in the context of reverse engineering and (ii) predict the total barrier and time to reverse engineer multiple samples of the same product. The remainder of this paper is organized as follows: In Sec. 2, we present a brief overview of the pertinent reverse engineering metrics developed previously by the authors. The additional metrics—capable of estimating the time and barrier of tolerance extraction—are then introduced in Sec. 3, followed by a case study to illustrate the validity and limitations of the newly developed model in Sec. 4. Concluding remarks are provided in Sec. 5.

2 Technical Preliminaries—Metrics for Reverse Engineering Only One Sample of a Product

In this section, we provide the technical background for the new developments in this paper. Harston and Mattson [12] create an analogy between the extraction of information from a product during reverse engineering and a simple resistor–capacitor circuit. In so doing, they successfully used Ohm's law to predict the time to reverse engineer the first sample of a product with an average error of 12.2%. An overview of the pertinent metrics and parameters in their model is presented below and summarized in Table 1. A familiarity with these metrics and definitions is requisite for understanding the developments of the current paper.

2.1 Unit of Information. (K)—The quantity of unextracted, pertinent information remaining in a product at any time. For example, in the case of reverse engineering a product's geometry, K could refer to the number of geometric dimensions that have not been measured yet. K_0 is the amount of information initially contained by a product.

2.2 Information Flow Rate. (F)—The rate at which information remaining in a product is extracted.

2.3 Power. (P)—Effort per time exerted by a reverse engineering team to extract information. The lower bound of P is zero and represents no effort being put forth to reverse engineer a product. The upper bound, one, signifies full effort at maximum efficiency.

Table 1 Reverse engineering parameters and metrics summary [12]

#	Parameter or metric	Electrical analogy	Relationships
1	Unit of information (K)	Charge (Q)	$0 < K \leq K_0$
2	Information flow rate (F)	Current (I)	$F = dK/dt$
3	Power (P)	Power (P)	$0 < P \leq 1$
4	Barrier (B)	Resistance (R)	$B = P/F^2$
5	Storage ability (S)	Capacitance (C)	$S = (KF)/P$
6	Time (T)	Time (T)	$T = -BS\ln(K/K_0)$

2.4 Barrier to Reverse Engineering. (B)—Anything that impedes reverse engineering [27]. As indicated by the relationship in Table 1, the barrier is a function of both the product being reverse engineered and the team reverse engineering it.

2.5 Information Storage Ability. (S)—The capability of a product to store information.

2.6 Time to Reverse Engineer. (T)—The total required man-time to reverse engineer a product.

3 Metrics Development for Characterizing the Effects of Learning When Reverse Engineering

In this section, we develop the metrics for predicting the time and barrier to reverse engineer multiple samples of the same product. The presentation of the metrics is divided into three main parts. In Sec. 3.1, we discuss how the flow rate of information changes during the process of reverse engineering multiple samples of the same product. In Sec. 3.2, we develop and present the metrics. Finally, in Sec. 3.3, we explain how to use the metrics to estimate the time to reverse engineer multiple samples of the same product.

3.1 The Behavior of Information Flow Rates When Reverse Engineering. When we reverse engineer a product, we extract information from that product (see definition in Sec. 1). Typically, these pieces of information are discrete in nature; thus, it is advantageous to look at K and F at discrete values of K, which we call *unextracted information levels*. Additionally, the values of F will vary depending on the reverse engineering sample (i.e., how many samples of the product have been reverse engineered). Therefore, we will use the subscripts $[\ \ ]_{k,s}$ to distinguish unextracted information level, k, and reverse engineering sample, s. For example, the information flow rate when three dimensions still need to be measured on the fourth product sample would be denoted $F_{3,4}$.

The way in which the flow of information varies when reverse engineering is illustrated in Fig. 1, which plots the amount of unextracted information in a product as a function of time for several reverse engineering samples of the same product. The first curve, labeled c_1, represents the first product sample that is reverse engineered. This curve resembles an exponential decaying relationship and is derived from Table 1, relationship 6. When $K = K_0$, the slope of c_1 is relatively steep, which means the extraction of information per unit time, or information flow rate,

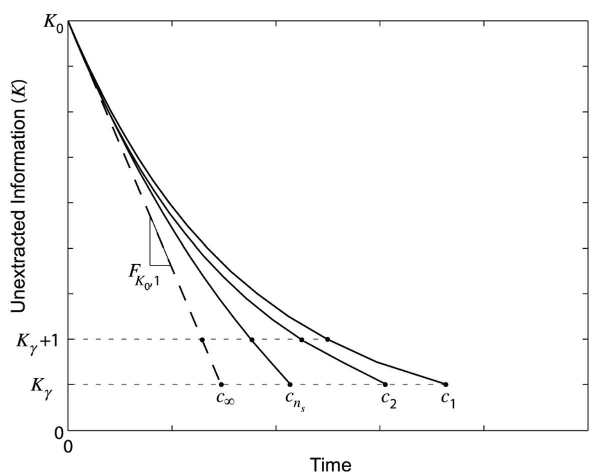

Fig. 1 Unextracted information in a product as a function of time—the curves for multiple reverse engineering samples are compared

is large in comparison to the slope of c_1 when $K = K_\gamma$, where K_γ is the lowest unextracted information level of interest.

This variation in information flow rate can be credited to the fact that reverse engineering encompasses more than just measuring dimensions, for example. It includes secondary procedures such as deciding which dimensions are pertinent, finding the dimensions in the product, documenting or recording the dimensions on a hand drawing or in a computer aided design (CAD) system, and verifying that all the needed dimensions have been extracted. When all of the aforementioned steps are performed, the flow rate of information is low, in comparison to when none or few of the secondary procedures are necessary for information extraction. This implies that the fastest, or largest, flow rate occurs when information is simply extracted without utilizing any secondary procedures.

When a person reverse engineers a second sample of a product, he or she utilizes some of the knowledge gained while reverse engineering the product the first time, obviating some of the steps of the reverse engineering process. For example, if someone is reverse engineering the geometry of a piston for the second time, the locations of the pertinent dimensions on the piston have already been determined during the first reverse engineering sample, as well as an appropriate documentation procedure. This is characterized in Fig. 1, where the slopes along the curve for the second product sample, labeled c_2, are generally steeper than those of c_1, resulting in less total time to reverse engineer the product. Reverse engineering additional samples, denoted by c_{n_s} in the plot, will yield similar results—flow rates will continue to increase and the reverse engineering time will continue to decrease. If the reverse engineering sample size is sufficiently large, then the sample curves will approach the dashed line in the plot, marked as c_∞. This line represents the reverse engineering process when it is performed at maximum efficiency.

The slope of c_∞ in Fig. 1 is the *initial flow rate*, $F_{K_0,1}$, and is described in detail in Ref. [12]. We also assume that the initial flow rate is the initial slope of each sample curve, or

$$F_{K_0,s} = F_{K_0,1}, \forall s \in \{1, 2, \ldots, n_s\} \quad (1)$$

where n_s is the number of reverse engineering samples. Based on this assumption, the initial flow rate remains the same for an individual, regardless of reverse engineering sample. Moreover, information requiring less extraction time is extracted from a product *first*, followed sequentially by units of information requiring more time. While this may or may not happen in practice, when empirical data gathered by the authors is rearranged according to the time to extract each unit of information—with the shortest times placed first—this relationship generally holds true (see Sec. 4).

The horizontal dashed lines in Fig. 1 help to visually track unextracted information levels along different sample curves. The lowest unextracted information level in the plot is K_γ, which is the closest discrete value for which the following is approximately true

$$K_\gamma = 0.05 * K_0 \quad (2)$$

This is the value typically used for K in relationship 6 in Table 1 to predict the total time to reverse engineer a product once [12]; the value $0.05 * K_0$ is used instead of $0.0 * K_0$ to ensure that relationship 6 yields a finite quantity of time. If the information flow rates of different samples at any unextracted information level, K_0 through K_γ, are compared to one another, we assume the following to be true

$$|F_{k,1}| \le |F_{k,s}| \le |F_{k,n_s}|, \quad \begin{cases} \forall k \in \{K_\gamma, K_\gamma + 1, \ldots, K_0\} \\ \forall s \in \{1, 2, \ldots, n_s\} \end{cases} \quad (3)$$

In other words, the flow rate at a particular unextracted information level k is bound by the flow rate of the first reverse engineering sample, $F_{k,1}$, and the flow rate of the last reverse engineering sample, F_{k,n_s}. Additionally, as stated above, if the product sample size is sufficiently large, the curves in the plot approach a linear prediction with the slope of the initial flow rate, $F_{K_0,1}$, or

$$\lim_{s\to\infty} F_{k,s} = F_{K_0,1}, \quad \forall k \in \{K_\gamma, K_\gamma + 1, \ldots, K_0\} \quad (4)$$

This suggests that as an individual learns while reverse engineering multiple samples of the same product, they drive the flow of information towards maximum efficiency.

The question remains as to how quickly (in terms of reverse engineering samples) information flow rates approach the initial flow rate. Some individuals are fast learners with regards to reverse engineering, while others are not. In Sec. 3.2, we introduce a parameter, reflective of the rate at which a person can learn, to help characterize this behavior.

3.2 Metrics for Reverse Engineering Multiple Samples of the Same Product. The metrics for reverse engineering multiple samples of the same product are derived from the first order response of a simple resistor–inductor circuit [28,29]. Thus, we are extending the electrical analogy created by Harston and Mattson [12] to include inductance, for two reasons: (i) we have observed that when reverse engineering multiple samples of the same product, the flow of information at an unextracted information level behaves like the first order response of electrical current in a resistor–inductor circuit and (ii) the inductance in a circuit provides an applicable parameter that can be used to characterize how quickly a person learns while reverse engineering.

The flow rate of information for any sample and unextracted information level, $F_{k,s}$, is calculated as

$$F_{k,s} = F_{k,1} e^{-(s-1)*B/Z} + F_{K_0,1}\left(1 - e^{-(s-1)*B/Z}\right) \quad (5)$$

where B is the barrier to reverse engineering, as defined in Table 1, and Z is termed the *learning factor*. The flow rate from the first reverse engineering sample at any unextracted information level, k, is denoted by $F_{k,1}$. This value is determined by solving for K from the relationship for T in Table 1, and substituting into the relationship for F in Table 1, which yields

$$F_{k,1} = \frac{-K_0}{BS} e^{-T/(BS)} \quad (6)$$

This equation can be further simplified by substituting the relationships for B, S, and T in Table 1, allowing $F_{k,1}$ to be rewritten as

$$F_{k,1} = \frac{F_{K_0,1} * k}{K_0} \quad (7)$$

where k has been substituted for K because we are interested in the flow rate at discrete, unextracted information levels. In this form, the first flow rate for each unextracted information level can easily be calculated and used in Eq. (5). Notice that when $s = 1$ in Eq. (5), the second term on the right hand side of the equation drops out and Eq. (5) simplifies to $F_{k,s} = F_{k,1}$. On the other hand, as s approaches infinity, Eq. (5) simplifies to $F_{k,s} = F_{K_0,1}$. This is the same behavior for information flow rates that we described in Sec. 3.1.

The learning factor, Z, is a measure of a person's ability to learn while reverse engineering multiple samples of the same product, given a particular measurement tool. A large Z indicates a high resistance to change in information flow rates, or in other words, it is difficult for the individual/team performing the reverse engineering to utilize information gained during previous iterations of the process. A small Z may indicate that the process is nearly automated, meaning that secondary reverse engineering procedures do not need to be repeated after the first reverse engineering sample. An example would be using a coordinate measuring machine to automatically scan geometry or using a scanning electron microscope to extract the material microstructure from several samples—the set up procedure is only done once, and then the process is automated. If all other parameters are equal, a person with a smaller Z will reverse engineer multiple samples of the same product quicker than someone with a larger Z.

The learning factor is calculated as

$$Z = \frac{B\left(F_{K_0,1} - F_{k,s}\right)}{dF_{k,s}/ds} \tag{8}$$

where $dF_{k,s}/ds$ indicates the change in information flow rate, $F_{k,s}$, per reverse engineering sample, s, for any flow rate besides the initial flow rate, $F_{K_0,1}$. The parameters that comprise Z are experimentally determined for an individual; more information on how this is done is provided in Sec. 3.3. A similar equation to Eq. (8) exists for inductance in a simple resistor–inductor circuit. In fact, the learning factor is analogous to inductance in an electrical circuit—both measure resistance to change in flow rates (electrical current or information flow rates). We note that Z does not change as information flow rates increase, nor is it dependent on the unextracted information level or reverse engineering sample—we assume that a person's *aptitude* to learn remains constant during the reverse engineering process. Again, this is similar to an inductor in an electrical circuit, where the inductance value remains constant, regardless of the electrical current flowing through it.

With a relationship defined for how the flow rate of information during reverse engineering changes, we can now calculate the total time to reverse engineer multiple samples of the same product as

$$T = -BS\ln\left(K_\gamma/K_0\right) + \sum_{(s=2)}^{n_s}\sum_{(k=K_\gamma+1)}^{K_0}\frac{1}{F_{k,s}} \tag{9}$$

where the 1 in the numerator represents one unit of information, ensuring that T has units of time. The first term on the right hand side of Eq. (9), $-BS\ln\left(K_\gamma/K_0\right)$, represents the time to reverse engineer the first product sample, as given by the relationships in Table 1. The second term of the equation accounts for all remaining samples; thus, the outer summation is initialized at $s=2$ and continues until n_s. For each sample, the reciprocal of $F_{k,s}$ is summed for all unextracted information levels starting with $k = K_\gamma + 1$ up through K_0. The flow rates at the unextracted information level K_γ are not included because this is a forward difference approximation, and inclusion of the flow rates at the lowest unextracted information level would overestimate the total time. The parameters and metrics that make up Eq. (9) can easily be calculated for any individual or product. As a result, the task of accurately estimating the time to reverse engineer a product becomes simple and straightforward. More information on how this is to be done is included in Sec. 3.3.

The time required for each individual reverse engineering sample (beyond the first) can also be determined by modifying Eq. (9) to get

$$\hat{T}_s = \sum_{k=K_\gamma+1}^{K_0}\frac{1}{F_{k,s}} \tag{10}$$

where the subscript s distinguishes the reverse engineering sample in question and the 1 in the numerator signifies one unit of information. It is important to note that the metrics developed here use discrete unextracted information levels to determine F, using Eq. (7). Because discrete points are used to characterize the entire curve, approximation error is introduced into the model. Therefore, to maintain a higher degree of accuracy, it is more appropriate to use the relationship for T in Table 1 for the first reverse engineering sample.

Up until this point, the metrics introduced have not considered the type of information being extracted from a product. Information type is a significant factor in reverse engineering, as the barrier and time for reverse engineering depend on the type of information that is contained in a product [12]. Each information type has a distinct initial flow rate, $F_{K_0,1}$ and learning factor, Z. Therefore, every information type needs to be considered separately. This will result in a different time to reverse engineer each information type. The total time to reverse engineer all the information types in a product, T^*, is calculated as

$$T^* = \sum_{i=1}^{n_I} T^i \tag{11}$$

where the superscript $[\ \ ]^i$ is used to distinguish information type, making T^i the time to reverse engineer one type of information as calculated with Eq. (9), and n_I is the total number of information types contained in the product.

The barrier to reverse engineer multiple samples of the same product is the same barrier that has been defined in Table 1. Each information type has a unique barrier; however, this barrier does not change with additional reverse engineering samples, despite the fact that information flow rates do increase. This is similar to a resistor in a resistor–inductor circuit—the value of its resistance remains the same, even though the current passing through it can change. Therefore, the effective barrier to reverse engineer multiple samples of an entire product is still calculated using the relationships presented in Ref. [12].

3.3 How to Use the Metrics. In this section, we explain how to use the metrics that were presented in Sec. 3.2 to estimate the time to reverse engineer multiple samples of the same product. This could be done in industry by original designers who are trying to protect their products, or by those performing benchmarking activities. The process is described by the flow chart in Fig. 2. To start the process, one must determine the number of information types, n_I, that are needed to reverse engineer the product. The index to count the number of information types, i, is initialized at 1.

Step 1 is to experimentally determine the initial flow rate, $F^i_{K_0,1}$, for a particular information type i [30]. This is done by using a uniform dimension extraction test. The goal of the test is to measure the average rate at which a person can extract information from a product when no secondary reverse engineering procedures are performed. In the test, an individual is asked to familiarize themselves with a particular dimension on a product. After this is done, the individual receives a measurement tool and the time is then recorded for them to measure the dimension. The process is repeated for many different dimensions of the same information type and the extraction rates are averaged to determine $F^i_{K_0,1}$. The resulting $F^i_{K_0,1}$ determined by the test can be used to calculate the metrics in step 4 for any product that contains the appropriate information type. In practice, the test only needs to be done once and then the value for $F^i_{K_0,1}$ can be reused for information type i, or a generic database containing the initial flow rates for typical measurement tools and operator skill levels could be developed.

Step 2 is to experimentally determine and calculate the learning factor, or Z^i. It is calculated using Eq. (8), which requires a flow rate, $F^i_{k,s}$, other than the initial flow rate, and its associated derivative with respect to reverse engineering sample, $dF^i_{k,s}/ds$. These values are determined for an individual using a uniform dimension extraction test, similar to how $F^i_{K_0,1}$ is determined. However, in this test the person must extract dimensions from multiple product samples. During the test, the person is handed a product and asked to extract several difficult pieces of information. Then, they are asked to repeat the measurements on a new sample of the same product. $F^i_{k,s}$ and $dF^i_{k,s}/ds$ are recorded, and Z^i is calculated with Eq. (8). It is important that the information in this test be difficult to extract, so as to emphasize differences in flow rates between

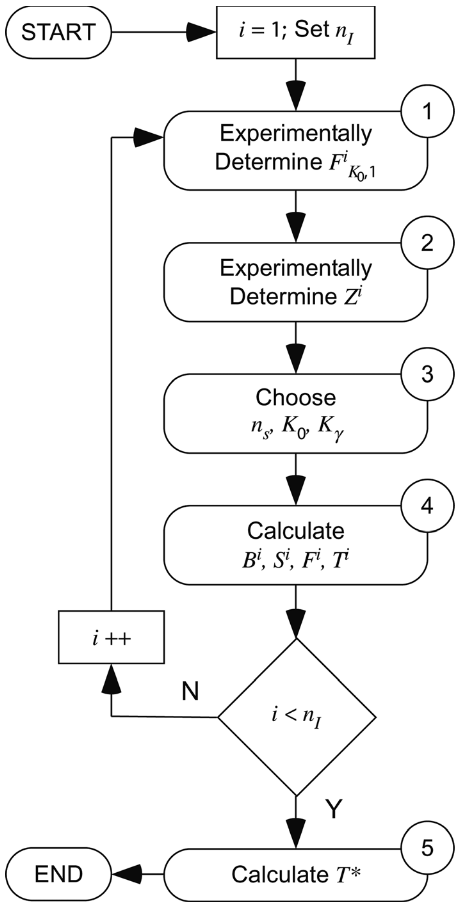

Fig. 2 The process for predicting the time to reverse engineer multiple samples of the same product

samples due to actual learning that occurs in the process, and not natural human variation.

Step 3 is to choose the number of samples, n_s, and the total amount of unextracted information, K_0. With K_0 defined, K_γ can be calculated using Eq. (2). In practice, when a person reverse engineers a product, n_s will be known initially, while K_0 will not. Otherwise, for the person using these metrics to predict the reverse engineering time of their competitors, n_s must be estimated, but K_0 will be known. Accurately predicting n_s can be a challenging task, as the number of samples used for reverse engineering will vary for different products and companies; however, an estimate can be made based on a statistical analysis to determine an adequate sample size. Additionally, if the price is significantly high for one product, then n_s will likely be small.

Step 4 is to calculate the barrier, B^i, storage ability, S^i, the flow rates, F^i, and time, T^i, with relationships 4 and 5 in Table 1, and Eqs. (5), and (9), respectively. The P, F, and K typically used to calculate B^i and S^i are $P = 1$, $F = F^i_{K_0,1}$, and $K = K_0$. Steps 1–4 are then repeated for each information type of interest in the product, after which the total time to reverse engineer the product, T^*, is then calculated in step 5 with Eq. (11). Thus, the time to reverse engineer a product can be estimated, without having to actually reverse engineer the product. In Sec. 4, we discuss the accuracy and limitations of the model with a case study.

4 Case Study and Validation

In this section, we present an empirical study with the purpose of showing that the time to reverse engineer multiple samples of the same product can be estimated by the relationships presented in this paper. For this study, only geometric information was extracted and analyzed. Two individuals were asked to reverse engineer multiple samples of a spool valve block and its associated spool from a flowserve digital positioner seen in Fig. 3. According to sources at Flowserve, the spool valve block and spool have been reverse engineered and imitated by competitors of Flowserve; therefore, these parts merit our attention in this study on reverse engineering.

Before beginning the reverse engineering process, the initial flow rate and learning factor of both individuals in the study were determined as described in steps 1 and 2 from Sec. 3.3. The individuals were then instructed to extract and record geometric dimensions using digital calipers with enough detail that the product could be recreated if needed. Multiple samples (between 10 and 30) of both parts were analyzed by the individuals while the time to reverse engineer was recorded.

Independently, the number of samples, n_s, and the total amount of unextracted information, K_0, were chosen. This enabled the calculation of the barriers to reverse engineering, storage abilities, information flow rates, and times to reverse engineer the product samples. These values, excluding the information flow rates to preserve clarity in presentation, are located in Table 2. The actual extraction times along with the errors are also listed in Table 2. As shown, the total errors ranged from −10.7% to 6.8%.

For comparison purposes, we will look at several models for predicting the time to reverse engineer multiple samples of the same product. Each model is described below and the total absolute error, $|\varepsilon|$, for each model is compared in Table 3.

4.1 Linear Model. —The time for one sample predicted by this model is calculated as $\hat{T}_{\text{lin}} = (K_0 - K_\gamma)/F_{K_0,1}$. In other words,

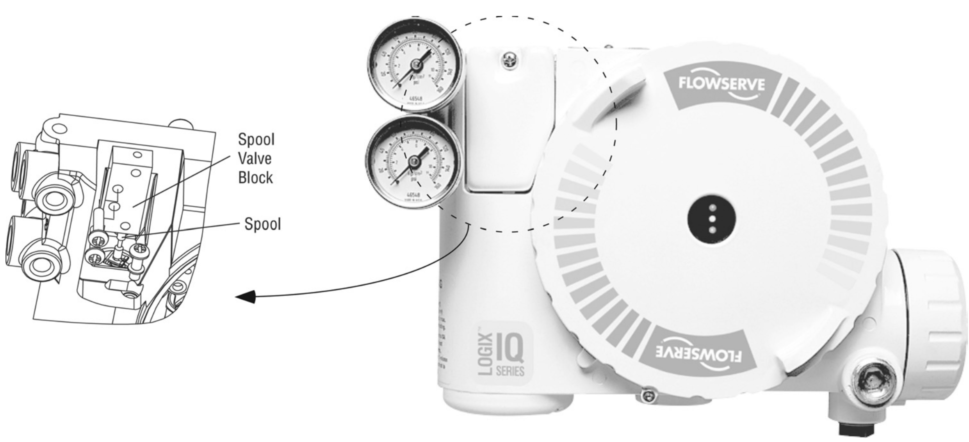

Fig. 3 Flowserve 3400IQ digital positioner with spool block valve and spool shown—image adapted from Ref. [31]

Table 2 Reverse engineering parameters and metrics for geometric information of Flowserve digital positioner spool and spool block valve

Individual	Part	$F_{K_0,1}$ (dim/s)	Z	n_s	K_0	K_γ	B	S	$T^*(s)$	Actual $T^*(s)$	ε (%)
#1	Spool	0.065	2127	30	24	1	236.7	1.560	13,613	14,611	6.8
#2	Spool	0.057	2912	10	23	1	307.8	1.311	6606	5966	−10.7
#1	Block	0.065	2127	10	34	2	236.7	2.210	8179	7630	−7.2
#2	Block	0.057	2912	29	35	2	307.8	1.995	21,668	21,791	0.6

Table 3 Comparison of different models to predict the time to reverse engineer multiple samples of the same product

Individual	Part	Linear model $\lvert\varepsilon\rvert$ (%)	Exponential model $\lvert\varepsilon\rvert$ (%)	Combined model $\lvert\varepsilon\rvert$ (%)	Learning model $\lvert\varepsilon\rvert$ (%)
#1	Spool	27.3	140.9	21.7	6.8
#2	Spool	35.3	112.1	20.6	10.7
#1	Block	35.5	94.2	22.5	7.2
#2	Block	23.0	133.9	17.5	0.6
Average		30.3	120.3	20.6	6.3

to calculate the total time for one sample, the quantity of information contained in a product is divided by the initial flow rate. This results in a linear relationship between K and T. The total time, T^*_{lin}, is then calculated as $T^*_{\mathrm{lin}} = n_s * \hat{T}_{\mathrm{lin}}$. This is the simplest model, and does not account for any variation in information flow rates. As shown in Table 3, the average absolute error when using this model was 30.3%.

4.2 Exponential Model. —For this model, the total time, T^*_{exp}, is calculated as $T^*_{\mathrm{exp}} = n_s * \hat{T}_{\mathrm{exp}}$, where $\hat{T}_{\mathrm{exp}}$ is determined from relationship 6 in Table 1. This is the time that is predicted using the previous metrics [12], where learning is not accounted for. It is called the exponential model because when K is plotted against T, it resembles an exponentially decaying relationship. As stated in Sec. 1, this model will typically overestimate the time to reverse multiple samples of the same product. This is especially evident for this case study, where the average $\lvert\varepsilon\rvert$ shown in Table 3 for the exponential model was 120.3%.

4.3 Combined Model. This model is a combination of the linear and exponential models. The total time is determined as

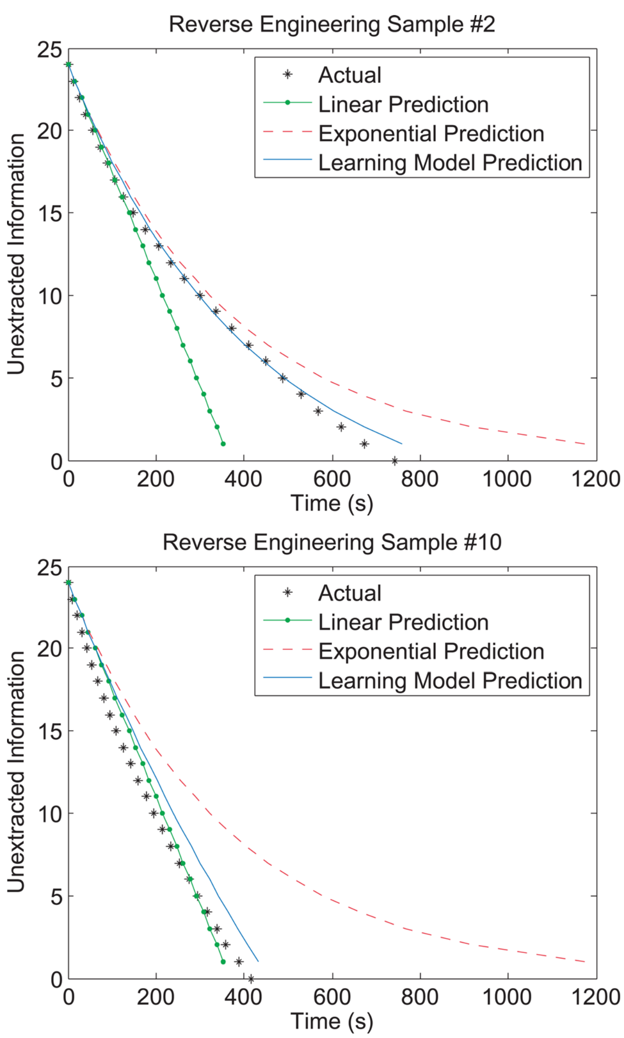

Fig. 4 Unextracted geometric dimensions of the spool as a function of time for individual # 1—samples 2, 5, 10, and 30 are shown

$T^*_{\text{com}} = T^*_{\text{exp}} + (n_s - 1) * T^*_{\text{lin}}$. We note that the exact same result will occur if Z is nearly zero in the learning model. In this case study, the combined model had the second best time prediction in Table 3, with an average error of 20.6%.

4.4 Learning Model. —This is the model developed in this paper. The total time is calculated here with Eq. (9), because we are only dealing with one information type. Without exception, the learning model outperforms the other models for predicting the time to reverse engineer multiple samples of the same product. The learning model predicted the times with an average absolute error across all tests of 6.3% (see Table 3). The stark contrast in accuracy between the learning model and the other models suggests that learning plays an important role in reverse engineering when using manual equipment such as digital calipers to extract information from a product.

The plots in Fig. 4 display the actual results and model predictions for spool samples 2, 5, 10, and 30 of individual #1. While the plots are for a single individual and product, they are representative and consistent with other tests that we have performed. The combined model is not explicitly called out in the plots, because it is only a combination of the linear and exponential models, both of which are shown. As the sample number increases, the data, which is marked by the asterisks in the plot, moves away from the exponential prediction towards the linear prediction. Likewise, the learning model curve, begins at the exponential curve and moves toward the linear curve at a rate that closely matches the real data.

We note that the data in the plots have been rearranged according to the time to extract each dimension - with the shortest times plotted first - and are not plotted in the order of dimension extraction. According to our assumptions given by Eqs. (1), (3), and (4), the flow rates should never be larger than $F_{K_0,1}$, which is the slope of the linear prediction in Fig. 4; however, some of the flow rates for reverse engineering sample #30 are clearly larger than $F_{K_0,1}$. This is explained by how we obtained $F_{K_0,1}$—by averaging the quickest times to extract several simple dimensions from an arbitrary product (see Sec. 3.3 step 1). Since the $F_{K_0,1}$ used here is an average, it is likely that some information will be extracted quicker due to natural variation. If the data is not rearranged according to the time to extract each dimension for sample #30, with the shortest times plotted first, the actual data appears more linear in nature and strongly correlates with the linear prediction.

5 Concluding Remarks

In this paper, we have presented general metrics for evaluating the time to reverse engineer multiple samples of the same product, which is a continuation of the research previously done by Harston and Mattson in Ref. [12]. An exponential decay function adequately describes the relationship between unextracted information remaining in a product and time for the first reverse engineering product sample. With subsequent samples, the relationship becomes more linear, due to changes in the flow of information. We have introduced supporting metrics that characterize this change in information flow rates due to learning.

A study involving multiple product samples of a spool and a spool valve block from a flowserve digital positioner has been offered to both demonstrate the use of the metrics and serve as empirical validation. The study confirms that as reverse engineering samples increase, the flow rates at all unextracted information levels increase toward the same asymptotical limit—the theoretical fastest flow rate, much like the response of electrical current in a resistor–inductor circuit. Moreover, the example suggests that if certain information is known about a product, the person reverse engineering, and the product sample size, then the metrics can be used to accurately estimate the total time needed to reverse engineer the geometry of a product, and, in this case with an average absolute error of 6.3%. Although this paper focuses on geometric information, the metrics defined here can also apply to other information types such as electrical conductivity, elasticity, tensile strength, or even color.

Acknowledgment

This research was partially supported by National Science Foundation Grant No. CMMI-0800904 for Christopher A. Mattson and Brent L. Adams. The authors also thank Flowserve for providing the parts used in the case study.

Nomenclature

B = barrier to extract information about a product from the product itself
F = estimated or actual rate at which information is extracted from a product
K = estimated or actual quantity of information remaining in a product
k = an unextracted information level
K_0 = the quantity of information initially contained by a product
K_γ = the lowest unextracted information level of interest
n_I = total number of information types
n_s = total number of reverse engineering samples
P = estimated power—effort per time—exerted to extract information contained by a product
S = a measure of a product's ability to store information
s = reverse engineering sample
T = estimated time to extract information from a product
Z = learning factor, a measure of a person's resistance to change in information flow rate

Subscripts, Superscripts, and Other Indicators

$[\ ]^*$ = indicates $[\ ]$ pertains to the product as a whole
$\widehat{[\ ]}$ = indicates $[\ ]$ pertains to a single reverse engineering sample
$[\ ]_{k,s}$ = indicates $[\ ]$ is evaluated at unextracted information level k and reverse engineering sample s
$[\ ]^i$ = indicates $[\ ]$ is evaluated for information type i

References

[1] Schnaars, S. P., 1994, Managing Imitation Strategies: How Later Entrants Seize Markets From Pioneers, The Free Press, NY.
[2] Zhou, K. Z., 2006, "Innovation, Imitation, and New Product Performance: The Case of China," Ind. Mark. Manage., **35**, pp. 394–402.
[3] Urban, G. L., Carter, T., Gaskin, S., and Mucha, Z., 1986, "Market Share Rewards to Pioneering Brands: An Emperical Analysis and Strategic Implications," Manage. Sci., **32**, p. 6.
[4] Greenstein, S., 2004, "Imitation Happens," Technical Report, IEEE Computer Society.
[5] Kim, L., 1997, Imitation to Innovation: The Dynamics of Korea's Technological Learning, Harvard Business School Press, Boston, MA.
[6] McLoughlin, I., 2008, "Secure Embedded Systems: The Threat of Reverse Engineering," ICPADS'08: Proceedings of the 2008 14th IEEE International Conference on Parallel and Distributed Systems, IEEE Computer Society, pp. 729–736.
[7] Shapiro, C., 1985, "Patent Licensing and R & D Rivalry," Am. Econ. Rev., **75**, pp. 25–30.
[8] Nelson, R., and Winter, S., 1982, *An Evolutionary Theory of Economic Change*, Belknap Press, Cambridge, Massachusetts.
[9] Macmillan, I., McCaffery, M. L., and van Wijk, G., 1985, "Competitors' Responses to Easily Imitated New Products-Exploring Commercial Banking Product Introductions," Strategic Manage. J., **6**(1), pp. 75–86.
[10] Mansfield, E., Schwartz, M., and Wagner, S., 1981, "Imitation Costs and Patents: An Empirical Study," Economic J., **91**(364), pp. 907–918.
[11] Knight, D., 2011, "Return on Investment Analysis for Implementing Barriers to Reverse Engineering and Imitation," Master's thesis, Brigham Young University, Provo, UT.
[12] Harston, S. P., and Mattson, C. A., 2010, "Metrics for Evaluating the Barrier and Time to Reverse Engineer a Product," ASME J. Mech. Des., **132**, p. 041009.
[13] Varady, T., Martin, R. R., and Cox, J., 1997, "Reverse Engineering of Geometric Models—An Introduction," Comput.-Aided Des., **29**(4), pp. 255–268.
[14] Thompson, W. B., Owen, J. C., de St. Germain, H. J., Stark, S. R., and Henderson, T. C., 1999, "Feature-Based Reverse Engineering of Mechanical Parts," IEEE Trans. Rob. Autom., **15**(1), pp. 57–66.

[15] Montgomery, D. C., Runger, G. C., and Hubele, N. F., 2006, Engineering Statistics, 3rd ed., John Wiley & Sons, New York.
[16] Kaisarlis, G. J., Diplaris, S. C., and Sfantsikopoulos, M. M., 2007, "Position Tolerancing in Reverse Engineering: the Fixed Fastener Case," Proc. Inst. Mech. Eng. Part B. J. Eng. Manuf., **221**, pp. 457–465.
[17] Chase, K. W., and Parkinson, A. R., 1991, "A Survey of Research in the Application of Tolerance Analysis to the Design of Mechanical Assemblies," Res. Eng. Des., **3**, pp. 23–37.
[18] Chase, K. W., 1999, "Tolerance Allocation Methods for Designers," Technical Report, ADCATS Report No. 99–6.
[19] Kaisarlis, G. J., Diplaris, S. C., and Sfantsikopoulos, M. M., 2007, "Geometrical Position Tolerance Assignment in Reverse Engineering," Int. J. Comput. Integr. Manuf., **21**(1), pp. 89–96.
[20] Jamshidi, J., Mileham, A. R., and Owen, G. W., 2006, "Dimensional Tolerance Approximations for Reverse Engineering Applications," Proceedings of the 9th International Design Conference (Design 2006), The Design Society, pp. 855–862.
[21] Kaisarlis, G. J., Diplaris, S. C., and Sfantsikopoulos, M. M., 2000, "A Knowledge-Based System for Tolerance Allocation in Reverse Engineering," Proceedings of the 33rd International Matador Conference.
[22] Anonymous, 2006, *U.S. Army Reverse Engineering Handbook* (Guidlines and Procedures), Department of Defense, Washington DC.
[23] Prerau, M. J., Smith, A. C., Eden, U. T., Kubota, Y., Yanike, M., Suzuki, W., Graybiel, A. M., and Brown, E. N., 2009, "Characterizing Learning by Simultaneous Analysis of Continuous and Binary Measures of Performance," J. Neurophysiol., **102**, pp. 3060–3072.
[24] Usher, M., and McClelland, J., 2001, "The Time Course of Perceptual Choice: The Leaky, Competing Accumulator Model," Psychol. Rev., **108**(3), pp. 550–592.
[25] Adler, P. S., and Clark, K. B., 1991, "Behind the Learning Curve: A Sketch of the Learning Process," Manage. Sci., **37**(3), pp. 267–181.
[26] Argote, L., 1999, Organizational Learning: Creating, Retaining, and Transfering Knowledge, Kluwer Academic, Boston.
[27] Curtis, S. K., Harston, S. P., and Mattson, C. A., 2011, "The Fundamentals of Barriers to Reverse Engineering and Their Implementation Into Mechanical Components," Res. Eng. Des., **22**(4), pp. 245–261.
[28] Rizzoni, G., 2004, Principles and Applications of Electrical Engineering, McGraw-Hill, New York.
[29] Fogiel, M., 2002, Basic Electricity, Research and Education Association, Piscataway, NJ.
[30] Anderson, N., 2011, "Characterization of the Initial Flow Rate of Information During Reverse Engineering," Master's thesis, Brigham Young University, Provo, UT.
[31] Flowserve Corporation, 2006, 3400IQ Digital Positioner FCD LGENIM3401-00 – 06/06 User Instructions.