

MIT Open Access Articles

The Role of Design Complexity in Technology Improvement

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: McNerney, J. et al. "Role of design complexity in technology improvement." Proceedings of the National Academy of Sciences 108.22 (2011): 9008-9013.

As Published: <http://dx.doi.org/10.1073/pnas.1017298108>

Publisher: National Academy of Sciences (U.S.)

Persistent URL: <http://hdl.handle.net/1721.1/67448>

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Role of design complexity in technology improvement

James McNerney^{a,b}, J. Doayne Farmer^{a,c}, Sidney Redner^{a,b,d}, and Jessika E. Trancik^{a,e,1}

^aSanta Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501; ^bDepartment of Physics, Boston University, Boston, MA 02215; ^cLibera Università Internazionale degli Studi Sociali, Guido Carli, Viale Pola 12, 00198 Rome, Italy; ^dCenter for Polymer Studies, Boston University, Boston, MA 02215; and ^eEngineering Systems Division, Massachusetts Institute of Technology, Cambridge, MA 02139

Edited by Hans-Joachim Schellnhuber, Potsdam Institute for Climate Impact Research (PIK), Potsdam, Germany, and approved March 8, 2011 (received for review November 22, 2010)

We study a simple model for the evolution of the cost (or more generally the performance) of a technology or production process. The technology can be decomposed into n components, each of which interacts with a cluster of $d - 1$ other components. Innovation occurs through a series of trial-and-error events, each of which consists of randomly changing the cost of each component in a cluster, and accepting the changes only if the total cost of the cluster is lowered. We show that the relationship between the cost of the whole technology and the number of innovation attempts is asymptotically a power law, matching the functional form often observed for empirical data. The exponent α of the power law depends on the intrinsic difficulty of finding better components, and on what we term the design complexity: the more complex the design, the slower the rate of improvement. Letting d as defined above be the connectivity, in the special case in which the connectivity is constant, the design complexity is simply the connectivity. When the connectivity varies, bottlenecks can arise in which a few components limit progress. In this case the design complexity depends on the details of the design. The number of bottlenecks also determines whether progress is steady, or whether there are periods of stasis punctuated by occasional large changes. Our model connects the engineering properties of a design to historical studies of technology improvement.

design structure matrix | experience curve | learning curve | performance curve

The relation between a technology's cost c and the cumulative amount produced y is often empirically observed to be a power law of the form

$$c(y) \propto y^{-\alpha}, \quad [1]$$

where the exponent α characterizes the rate of improvement. This rate is commonly termed the progress ratio $2^{-\alpha}$, which is the factor by which costs decrease with each doubling of cumulative production. A typical reported value (1) is 0.8 (corresponding to $\alpha \approx .32$), which implies that the cost of the 200th item is 80% that of the 100th item. Power laws have been observed (or at least assumed to hold), for a wide variety of technologies (1–3), although other functional forms have also been suggested and in some cases provide plausible fits to the data*. We give examples of historical performance curves for several different technologies in Fig. 1.

The relationship between cost and cumulative production goes under several different names, including the “experience curve,” the “learning curve,” or the “progress function.” The terms are used interchangeably by some, whereas others assign distinct meanings (1, 4). We use the general term performance curve to denote a plot of any performance measure (such as cost) against any experience measure (such as cumulative production), regardless of the context. Performance curve studies first appeared in the 19th century (5, 6), but their application to manufacturing and technology originates from the 1936 study by Wright on aircraft production costs (7). The large literature on this subject spans engineering (8), economics (4, 9), management science (1), organizational learning (16), and public policy (17). Performance

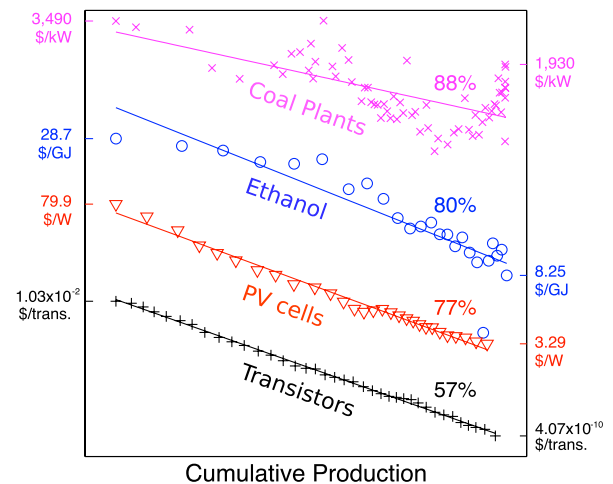


Fig. 1. Four empirical performance curves. Each curve was rescaled and shifted to aid comparison with a power law. The x - and y -coordinates of each series i were transformed via $\log x \rightarrow a_i + b_i \log x$, $\log y \rightarrow c_i + d_i \log y$. The constants a_i , b_i , c_i , and d_i were chosen to yield series with approximately the same slope and range, and are given in *SI Text*. Tick marks and labels on the left vertical axis show the first price (in real 2000 dollars) of the corresponding time series, and those of the right vertical axis show the last price. Lines are least-squares fits to a power law. Percentages are the progress ratios of the fitted power laws. Source: coal plants (10), ethanol (11), photovoltaic cells (12, 13, 14), transistors (15).

curves have been constructed for individuals, production processes, firms, and industries (1).

The power law assumption has been used by firm managers (18) and government policy makers (17) to forecast how costs will drop with cumulative production. However, the potential for exploiting performance curves has so far not been fully realized, in part because there is no good theory explaining the observed empirical relationships. Why do performance curves tend to look like power laws, as opposed to some other functional form? What factors determine the exponent α , which governs the long-term rate of improvement? Why are some performance curves steady and others erratic? By suggesting answers to these questions, the theory we develop here can potentially be used to guide investment policy for technological change.

Author contributions: J.M., J.D.F., S.R., and J.E.T. performed research and wrote the paper.

The authors declare no conflict of interest.

This article is a PNAS Direct Submission.

Freely available online through the PNAS open access option.

*Koh and Magee (35, 36) claim an exponential function of time (Moore's law) predicts the performance of several different technologies. Goddard (34) claims costs follow a power law in production rate rather than cumulative production. Multivariate forms involving combinations of production rate, cumulative production, or time have been examined by Sinclair et al. (38) and Nordhaus (37).

¹To whom correspondence should be addressed. E-mail: trancik@mit.edu.

This article contains supporting information online at www.pnas.org/lookup/suppl/doi:10.1073/pnas.1017298108/-DCSupplemental.

An example of the possible usefulness of such a theory is climate change mitigation. Good forecasts of future costs of low-carbon energy technologies could help guide research and development funding and climate policy. Our theory suggests that based on the design of a technology we might be able to better forecast its rate of improvement, and therefore make better investments and better estimates of the cost of achieving low-carbon energy conversion.

There have been several previous attempts to construct theories to explain the functional form of performance curves (19–21). Muth constructed a model of a single-component technology in which innovation happens by proposing new designs at random (21). Using extreme value theory he derived conditions under which the rate of improvement is a power law. An extension to multiple components, called the production recipe model, was proposed by Auerswald et al. (19). In their model each component interacts with other components, and if a given component is replaced, it affects the cost of the components with which it interacts. They simulated their model and found that under some circumstances the performance curves appeared to be power laws. Other models include Bendler and Schlesinger, who derive a power law based on the assumption that barriers to improvement are distributed fractally (22), and Huberman, who represents the design process as a random evolving graph (20). More recently Frenken has used the Auerswald model to interpret and address questions such as the efficacy of outsourcing (23, 24). Other related models that use random search to model technological progress (but which do not directly address performance curves) are those of Silverberg and Verspagen (25, 26) and Thurner et al. (27).

In this paper we both simplify and extend the production recipe model of Auerswald et al. (19). The simplifications allow us to derive the emergence of a power law, and most importantly, to derive its exponent α . We find that $\alpha = 1/(\gamma d^*)$, where γ measures the intrinsic difficulty of finding better components and d^* is what we call the design complexity. When the connectivity of the components is constant the design complexity d^* is equal to the connectivity. When connectivity is variable, the complexity can also depend on the detailed properties of the design, in ways that we make clear. We also show that when costs are spread uniformly across a large number of components, the whole technology undergoes steady improvement. In contrast, when costs are dominated by a few components, the total cost undergoes erratic improvement. Our theory thus potentially gives insight into how to design a technology so that it will improve more rapidly and more steadily.

We should emphasize that many factors besides design can affect costs—for example, the cost of input materials or fuels may change due to market dynamics rather than technology design (10). Furthermore, design is generally focused not just on reducing costs, but also on improving other properties such as environmental performance or reliability. The variable “cost” in the theory here can be interpreted as any property that depends on technology design.

The Model

The production design consists of n components, which can be thought of as the parts of a technology or the steps in an industrial process¹. Each component i has a cost c_i . The total cost κ of the design is the sum of the component costs: $\kappa = c_1 + c_2 + \dots + c_n$. A component's cost changes as new implementations for the component are found. For example, a component representing the step “move a box across a room” may initially be implemented by a forklift, which could later be replaced by a conveyor belt.

¹The original production recipe model (19) contained 6 parameters. We eliminated four of them as follows: length of production run $T \rightarrow \infty$; output-per-attempted-recipe-change $B \rightarrow 1$; available implementations per component $s \rightarrow \infty$; search distance $\delta \rightarrow 1$.

Cost reductions occur through repeated changes to one or more components.

Components are not isolated from one another, but rather interact as parts of the overall design. Thus changing one component not only affects its cost, but also the costs of other dependent components. Components may be viewed as nodes in a directed network, with links from each component to those that depend on it. The relationship between the nodes and links can alternatively be characterized by an adjacency matrix. In systems engineering and management science this matrix is known as the design structure matrix (DSM) (28–30). A DSM is an $n \times n$ matrix with an entry in row i and column j if a change in component j affects component i (Fig. 2). The matrix is usually binary (31, 32); however, weighted interactions have also been considered (33). DSMs have been found to be useful in understanding and improving complex manufacturing and technology development processes.

The model is simulated as follows:

1. Pick a random component i .
2. Use the DSM to identify the set of components $\mathcal{A}_i = \{j\}$ whose costs depend on i (the outset of i).
3. Determine a new cost c'_j for each component $j \in \mathcal{A}_i$ from a fixed probability distribution f .
4. If the sum of the new costs, $a'_i = \sum_{j \in \mathcal{A}_i} c'_j$, is less than the current sum, a_i , then each c_j is changed to c'_j . Otherwise, the new cost set is rejected.

This process is repeated for t steps. The costs are defined on $[0,1]$. We assume a probability density function that for small values of c_i has the form $f(c_i) \propto c_i^{\gamma-1}$; i.e., the cumulative distribution $F(c_i) = \int_0^{c_i} f(c)dc \propto c_i^\gamma$. The exponent γ specifies the difficulty of reducing costs of individual components, with higher γ corresponding to higher difficulty. This functional form is fairly general in that it covers any distribution with a power-series expansion at $c = 0$.

Independent Components

We first consider the simple but unrealistic case of a technology with n independent components. This generalizes the one component case originally studied by Muth (21). The cost of a given component at time t is equivalent to the minimum of t independent, identically distributed random variables. In *SI Text*, we use extreme value theory to show that to first order in n/t the expected cost $E[\kappa(t)]$ is

$$E[\kappa(t)] = \Gamma\left(1 + \frac{1}{\gamma}\right) \left(\frac{t}{n}\right)^{-1/\gamma}, \quad [2]$$

where $\Gamma(a)$ is Euler's gamma function.

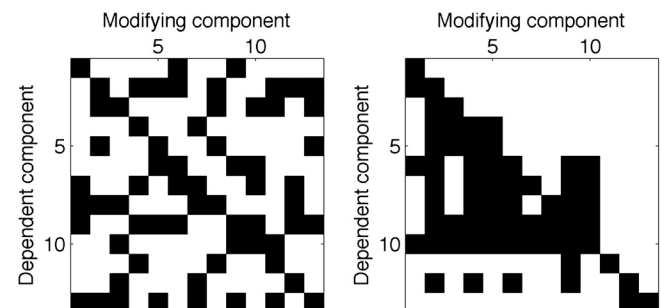


Fig. 2. Example design structure matrices (DSMs) with $n = 13$ components. Black squares represent links. The DSM on the left was randomly generated to have fixed out-degree for each component. The DSM on the right represents the design of an automobile brake system (31). All diagonal elements are present because a component always affects its own cost.

of the distribution vs. time. In every case the top envelope follows a straight line throughout most of the time range. The behavior of the bottom envelope is more complicated; in many cases, such as Fig. 5 Left, this bottom envelope also follows a straight line, but in others (for example, Fig. 5 Center) the bottom envelope changes slope over a large time range. A more precise statement can be made by following the contour corresponding to a given quantile through time. All quantiles eventually approach a line with slope $1/d^*$. However, the upper quantiles converge to this line quickly, whereas in some cases the lower quantiles do so much later. This slower convergence stems from the difference in improvement rates of different components. Whenever there is a dramatic improvement in the slowest-improving component (or components), there is a period where the next slowest-improving component (or components) becomes important. During this time the lower d_i^{\min} value of the second component temporarily influences the rate of improvement. After a long time the slowest-improving component becomes more and more dominant, large updates become progressively more rare, and the slope becomes constant.

The model therefore suggests that properties of the design determine whether a technology's improvement will be steady or erratic. Homogeneous designs (with constant out-degree) are more likely to show an inexorable trend of steady improvement. Heterogeneous designs (with larger variability in out-degree) are more likely to improve in fits and starts. High variability among individual trajectories can be interpreted as indicating that historical contingency plays an important role. By this we mean that the particular choice of random numbers, rather than the overall trend, dominates the behavior. In this case progress appears to come about through a series of punctuated equilibria.

To summarize, in this section we have shown that the asymptotic magnitude of the fluctuations is determined by the number of bottlenecks; i.e., the number with $d_i^{\min} = d^*$. The fluctuations decrease as the number of bottlenecks increases. In the constant out-degree case all of the components are equivalent, and this number is just n . In the variable out-degree case, however, this number depends on the details of the DSM, which influence d_i^{\min} .

Testing the Model Predictions

Our model makes the testable prediction that the rate of improvement of a technology depends on the design complexity, which can be determined from a design structure matrix. The use of DSMs to analyze designs is widespread in the systems engineering and management science literature. Thus, one could potentially examine the DSMs of different technologies, compute their corresponding design complexities, and compare to the value of α based on the technology's history[‡]. Thus we are able to make a quantitative prediction about learning curves. This is in contrast to previous work, which did not make testable predictions about α [§]. This test is complicated by the fact that α also depends on γ , which describes the inherent difficulty of improving individual components, which in turn depends on the inherent difficulty of the innovation problem as well as the collective effectiveness of inventors in generating improvements. The exponent γ is problematic to measure independently. Nonetheless, one could examine a collection of different technologies and either assume that γ is constant or that the variations in γ average out. Subject to these caveats, the model then predicts that the design complexity of the DSM should be inversely proportional to the estimated α of the historical trajectory. A byproduct of

[‡]One problem that must be considered is that of resolution. As an approximation, a DSM can be constructed at the coarse level of entire systems (e.g., "electrical system," "fuel system") or it can be constructed more accurately at the microscopic level in terms of individual parts. In general these will give different design complexities.

[§]A possible exception is Huberman (20), who presents a theory in terms of an evolving graph, and gives a formula that predicts power law scaling as a function of "the number of new shortcuts." It is not clear, however, whether this could ever be measured.

such a study is that it would yield an estimate of γ in different technologies.

To compare the model predictions to real data one must relate the number of attempted improvements to something measurable. It is not straightforward to measure the effort that has gone into improving a technology, and to compare to real data one must use proxies. The most commonly used proxy is cumulative production, but other possibilities include cumulative investment, installed capacity, research and development expenditure, or even time. The best proxy for innovation effort is a subject of serious debate in the literature (34–38).

Possible Extensions to the Model

There are a variety of possible ways to extend the model to make it more realistic. For example, the model currently assumes that the design network described by the DSM is constant through time, but often improvements to a technology come about by modifying the design network. One can potentially extend the model by adding an evolutionary model for the generation of new DSMs.

The possibility that the design complexity d^* changes through time suggests another empirical prediction. According to our theory, when d^* changes, α changes as well. One can conceivably examine an historical sequence of design matrices, compute their values of d^* , and compare the predicted $\alpha \sim 1/d^*$ to the corresponding observed values of α in the corresponding periods in time. Our theory predicts that these should be positively correlated.

We have assumed a particular model of learning in which improvement attempts are made at random, with no regard to the history of previous improvements or knowledge of the technology. An intelligent designer should be able to do (as well or) better, drawing on his or her knowledge of science, engineering, and present and past designs. (We note that for particularly complex design problems, random search may be computationally the most efficient option.) The model we study here can be viewed as a worst case, which should be an indicator of the difficulty of design under any approach: A problem that is harder for a designer to solve under random search is also likely to be more difficult to solve with a more directed search.

Discussion

We have developed a model that both simplifies and generalizes the original model of Auerswald et al. (19), which predicts the improvement of cost as a function of the number of innovation attempts. Whereas we have formulated the model in terms of cost, one could equally well have used any performance measure of the technology that has the property of being additive across the components. Our analysis makes clear predictions about the trajectories of technology improvement. The mean behavior of the cost is described by a power law with exponent $\alpha = 1/(\gamma d^*)$, where d^* is the design complexity and γ describes the intrinsic difficulty of improving individual components. In the case of constant connectivity (out-degree) the design complexity is just the connectivity, but in general the complexity can depend on details of the design matrix, as spelled out in Eq. 8. In addition, the range of variation in technological improvement trajectories depends on the number of bottlenecks. This number coincides with the total number of components n in the case of constant connectivity, but in general the number of bottlenecks is smaller, and depends on the detailed arrangement of the interactions in the design.

Many studies in the past have discussed effects that contribute to technological improvement, such as learning-by-doing, research and development, or capital investment. Our approach here is generic in the sense that it could apply to any of these mechanisms. As long as these mechanisms cause innovation attempts that can be modeled as a process of trial and error,

