Quantifying the impact of product changes on manufacturing performance

T. Dooper[1], L. F. P. Etman[2], A. A. Alblas[2]

[1]FruitPunch AI
[2]Eindhoven University of Technology

Abstract: Every adjustment to a physical product disrupts the manufacturing organization, requiring adaptation in tools and processes. The resulting disruption to manufacturing performance is poorly understood. We use design structure matrices and a complexity metric to quantify the complexity and change of product architecture in an explorative, small-scale experiment. Based on the results we develop two propositions to guide further research into the factors that affect the shape of consecutive learning curves upon product changes. The first proposition is that after product change, the complexity of the novel part of product architecture is responsible for the initial decrease in manufacturing performance. Second, we propose that the asymptote of a learning curve and the complexity of a product's architecture are inversely related.

Keywords: Design structure matrix, DSM, Complexity, Delta, Engineering change, Organizational learning, Learning curve, Manufacturing performance

1 Introduction

To keep up with market demands, companies continuously improve their products. Every adjustment to a physical product however, disrupts the manufacturing organization: Tools and processes have to be updated and possibly redesigned and employees need to adapt to the new situation (Gopal et al., 2013). Most companies have difficulty estimating the extent of this disruption, but the complexity of a change seems to be an important determining factor (Eckert et al., 2009; Jarratt et al., 2011).

Manufacturing performance has long been known to display learning effects: As an organization gains experience in producing a certain product, its performance improves (Wright, 1936; Muth, 1986; Argote and Miron-Spektor, 2011). By shaking up the manufacturing process, product changes may impede this ongoing learning process. Improved understanding of the impact of product changes on manufacturing performance could form the basis of better production planning, improved resource allocation, or market entry decisions.

In this study, we synthesize existing systems engineering methodologies to quantify product change and complexity in a small-scale experiment with individual participants. Based on the results, we develop two propositions for further research on the impacts of change complexity on manufacturing performance and offer new tools to model product changes. In doing so we continue research into the impacts of technological complexity on performance by McNerney et al. (2011); Sinha and De Weck (2012); Sinha et al. (2013); Rosiello and Maleki (2021). We also attempt to alleviate some of the difficulty in estimating the organizational impact of product changes described by Eckert et al. (2009) by building upon work by Smaling and De Weck (2007). Finally, we answer calls for research into the factors that affect organizational learning by Argote (2013) and Lapré and Nembhard (2011).

2 Background

Learning curves are an empirical phenomenon where performance increases with experience. The core idea is that with each repetition of an activity, learning takes place; knowledge generated is then employed and an increase in performance results. This phenomenon is apparent in individuals as well as organizations, with many similarities between the two (Thompson, 2012; Lapré and Nembhard, 2011). On an organizational level, different types of learning can occur (Dutton and Thomas, 1984; Argote and Miron-Spektor, 2011). For example, through repetition, employees might get better at their respective tasks, improvements could be made to the equipment, or organizations can adopt insights gathered by others, and invest in improvement programs.

Cumulative production volume is the most common indicator of organizational experience because it is relatively easy to measure and captures the idea of 'learning by doing'. This measure is typically plotted on the horizontal axis of a learning curve graph. Performance is then plotted along the vertical axis and can be measured in many ways (Lapré and Nembhard, 2011; Argote and Miron-Spektor, 2011). Learning curves are therefore also referred to as 'performance curves'. Cycle time—the time that it takes to produce a single unit—is a common measure for organizational performance.

While organizational learning curves have often been investigated, the factors that influence their shape are still poorly understood. Learning curves vary wildly among organizations, even those manufacturing similar products (Lapré and

Nembhard, 2011; Argote and Miron-Spektor, 2011). This has been an obstacle in developing forecasting models and applying them in organizational practice.

Intuitively, product complexity should be an important factor in organizational learning: learning something complex takes more time. However, its impact has hardly been researched empirically. The operationalization of a complexity variable in operations management research has often been subjective, through surveys (Simonin, 1999; Chapman and Hyland, 2004; Anzanello and Fogliatto, 2010), or through the judgement of the researchers themselves (Griffin, 1997a,b; Clift and Vandenbosch, 1999). Most studies that investigated product complexity specifically did not define it on a level of detail that would allow distinguishing two versions of a product. The only exception are the graph-based methods employed by Rosiello and Maleki (2021) and McNerney et al. (2011).

In systems engineering, a product is generally seen as a collection of components that have some type of connection (Simon, 1996) and a common purpose. Components are connected via proximity or flow of matter, energy or information (Blanchard and Fabrycky, 2011; Walden et al., 2015). Product complexity then arises from the interactions between these components (Maier and Rechtin, 2009). It is often described as the difficulty to predict system properties and behaviors from analysis of its parts (Anderson and Joglekar, 2012; Suh, 2005). So-called 'emergent properties' (Fisher, 2006) cannot be inferred by looking at the components in isolation. The interplay of all components together is responsible for a product's desired functionality, but unexpected emergent properties can undermine this intended purpose.

3. Methodology

When investigating the impact of product complexity, the operations management literature has used very general measures. Comparing two versions of a product, before and after change, requires a more detailed perspective and the design structure matrix (DSM) might be an excellent tool to that end, since it allows analysis of a system to an arbitrary degree of granularity. A DSM is analogous to a network graph and therefore also a promising method to investigate the relationship between product complexity and organizational learning: Learning has successfully been modeled as a network search process by Shrager et al. (1988); Fioretti (2007) and McNerney et al. (2011).

3.1 Capturing Product Architecture and Change

A design structure matrix (DSM) can be used to display a product as a network of components and interfaces (Eppinger and Browning, 2012). Interfaces between components can be mapped in a square matrix, where every row and column correspond to a particular system component and every cell corresponds to an interface between the two components corresponding to the row-column pair. If there is an interface, or dependency, between two components, the corresponding cells in the matrix are marked. This results in a symmetric matrix with marks in the off-diagonal cells, as seen in Figure 1.

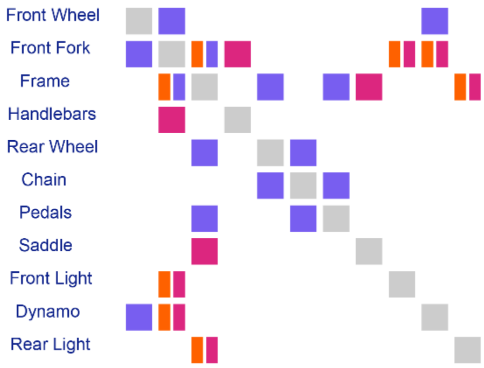
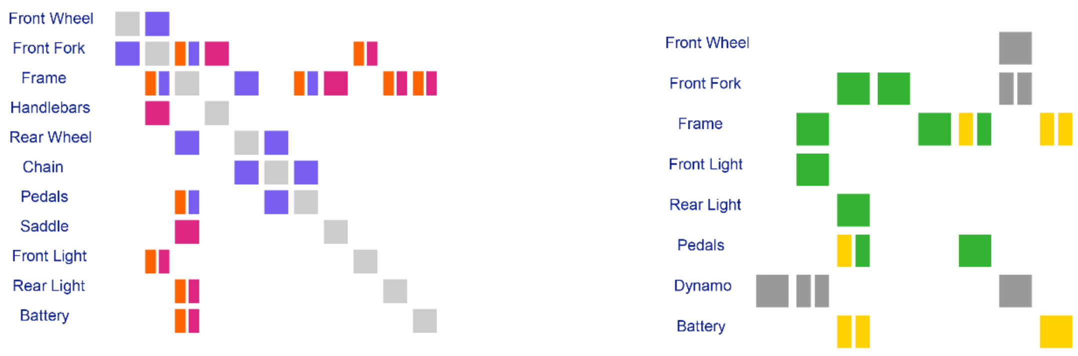


Figure 1: Example DSM of a bicycle. Purple matrix entries indicate a mechanical, moving interface between components; pink indicates a rigid mechanical interface; and orange an electrical interface. Some pairs of components have multiple types of dependencies between them. For example, there is a mechanical, moving connection between Front Fork and Frame, but also an electrical one: The Front Fork is able to rotate relative to the Frame, to allow steering. Since the Dynamo of the bike in this example is mounted on the Front Fork, current from the Dynamo travels via the Front Fork to the frame to reach the Rear Light, resulting in an electrical connection between Frame and Front Fork as well.

To quantify the difference between several alternative changes to a product's baseline design, Smaling and De Weck (2007) coined the idea of the Delta DSM. This Delta DSM captures changes, additions and removals of components and interfaces. It is constructed by starting with a DSM of the system before change and subsequently removing the marks from all cells. Then, new components are added to the matrix in the form of new rows and columns, resulting in a larger square matrix. Whether a component is removed, changed or added is indicated on the diagonal of the Delta DSM. Furthermore, every removed, changed and added interface is entered into off-diagonal cells of the matrix. Interfaces that remain unchanged don not appear in the Delta DSM. Figure 2 illustrates the Delta DSM concept by changing the bicycle of Figure 1 into an electrical bicycle. In the Delta DSM of Figure 2, we omit components if they remain unchanged and all of their interfaces remain unchanged as well. This eliminates empty rows and columns from the Delta DSM.



(a) DSM of an electrical bicycle. (b) Delta DSM of Figure 1 to 2a.

Figure 2: Representation of the change from the bicycle in Figure 1 to the electrical bicycle in Figure 2a. Figure 2b shows the Delta DSM for this change, with new components and interfaces marked in yellow, removals indicated in grey and changes in green. Components such as the Handlebars are omitted from the Delta DSM whenever they remain unchanged and all of their interfaces remain unchanged as well.

3.2 Quantifying Product Complexity

Based on the mathematical property of matrix energy (Li et al., 2012), Sinha (2014) developed a metric to quantify structural complexity of a system captured in an $n \times n$ DSM, which breaks down complexity into three parts:

$$C(\alpha, \beta, A) = C_1 + C_2 \cdot C_3 = \sum_{i=1}^{n} \alpha_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} \beta_{i,j,k} \cdot \frac{E(A)}{n} \tag{1}$$

The complexities of individual components, α_i are summed up to obtain C_1, the 'component complexity'. $\beta_{i,j,k}$ is the complexity of the one-on-one interface of type k between components i and j; C_2, the 'interface complexity', is obtained by summing up these pair-wise interactions. C_3, the 'architectural complexity', is the matrix energy of the adjacency matrix, denoted $E(A)$, divided by the number of components n. The adjacency matrix A is equivalent to the system's DSM with ones indicating interfaces and zeros in all other cells, including those on the diagonal. The matrix energy is then defined to be the sum of the singular values of A.

3.3 Research Design

Since the impact of product complexity and change on organizational learning curves has hardly been investigated, we took an exploratory approach. An experimental study was set up, to allow for easily manipulable levels of complexity, change and learning. Since organizations don not lend themselves well to experiments, the study was carried out with individual participants. Though not directly generalizable to organizations, individual learning shows many parallels with organizational learning (Thompson, 2012; Lapré and Nembhard, 2011) and an experiment with individuals was deemed appropriate for an initial exploration of the relationship between product changes and learning. This resulted in a small-scale, easily reproducible experiment.

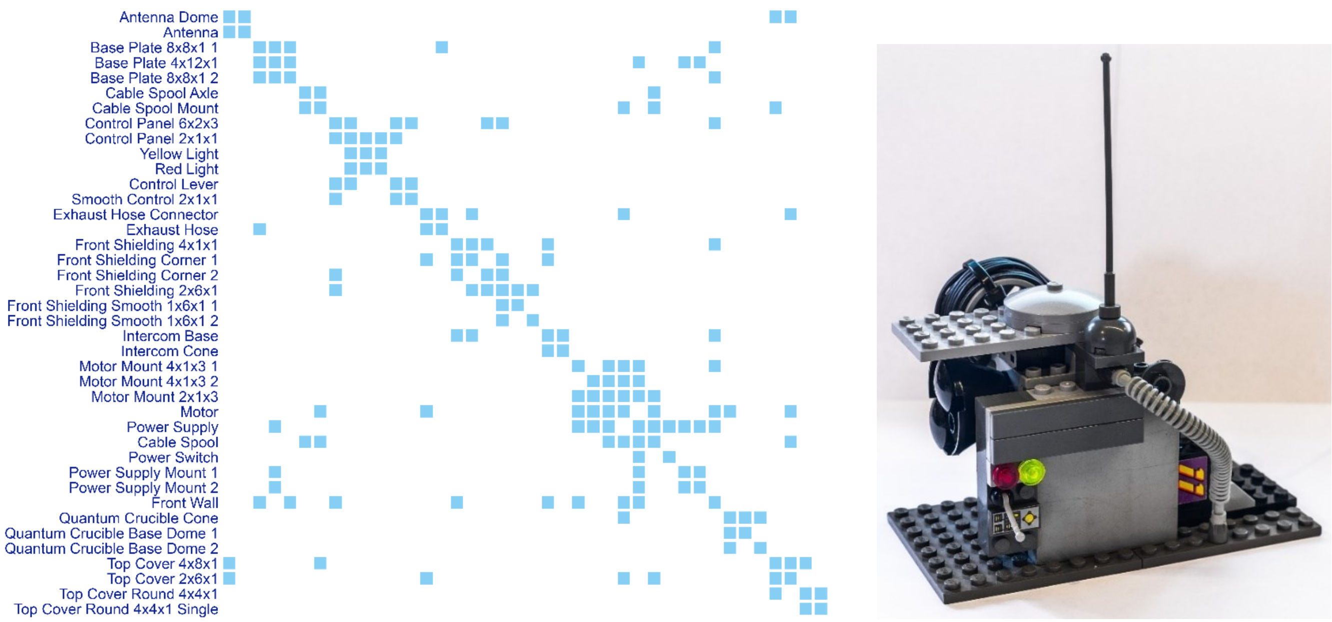
An assembly task was designed to elicit learning curves from individuals: Participants were asked to assemble two variants of a 'product' made out of Lego bricks, shown in Figures 3 and 4. After assembling one variant five times, participants would switch to the other, to mimic an engineering change which would cause a disruption in assembly performance. There was no control for participants' prior experience with Lego, but the machine was intentionally not modeled after any existing system to mitigate the influence of any prior engineering knowledge of the participants.

Half of the subjects built the simpler variant V_S first, the other half started with the complex variant V_C. The required Lego bricks were supplied in a slotted box before every attempt. Participants received pictures of the machine in various stages of assembly, with only minimal information about what was shown in every picture. They were intentionally not given step-by-step instructions for assembling the machine, rather they were left to figure out efficient assembly procedures themselves.

DSMs of both product variants are shown in Figures 3a and 4a. Components have an interface in the DSMs when they interlock (via studs or an axle), or when they are in physical contact. Assuming all components and interfaces are equally complex and assigning a complexity value of 1 to every component and interface, applying the metric by Sinha (2014) yields a complexity of 166 for product variant V_S and 244 for variant V_C. This represents a clear change in complexity switching between product variants. Furthermore, the two DSMs allow the derivation of two Delta DSMs representing the change from V_S to V_C and vice-versa.

(a) DSM of V_S. (b) V_S fully assembled.

Figure 3: The simple variant (V_S) of the machine that experiment participants assembled.



(a) DSM of V_C. (b) V_C fully assembled.

Figure 4: The complex variant (V_C) of the machine that experiment participants assembled.

4. Results

Eight people assembled both machines, four per experimental group. Two additional participants only constructed one of the two variants. Figure 5 shows the results for the 8 participants that constructed both variants. The pattern in Figure 5 of two consecutive learning curves, with a decrease in performance between them, corresponds to what is often observed in organizational practice. As is evident from Figure 5, there was large variability in performance between participants, likely due to their variable previous experience with assembling Lego models. Since the experimental groups were very small, this might have influenced the results.

One would expect mean assembly time for the complex product to be larger on average, regardless of order, than that of the simple product. In this small-scale experiment, the difference in assembly time was not significant though. However, the jump in cycle time when moving from V_S to V_C was significantly larger than from V_C to V_S. Also, practicing with V_C first significantly lowered the mean assembly time for V_S, but not the other way around: The group that made the V_S first did no better on the V_C than the other group, which started with the V_C.

An important qualitative observation was that most participants got the idea of turning over the slotted box with parts to speed up the assembly process. They felt that after assembling one machine variant a few times, the time spent on taking parts out of the box was limiting their performance. Some attempted to solve this problem by turning over the box and emptying it on the table at which they were sat. Most mentioned the idea when assembling V_S, only one person mentioned flipping the box while assembling V_C.

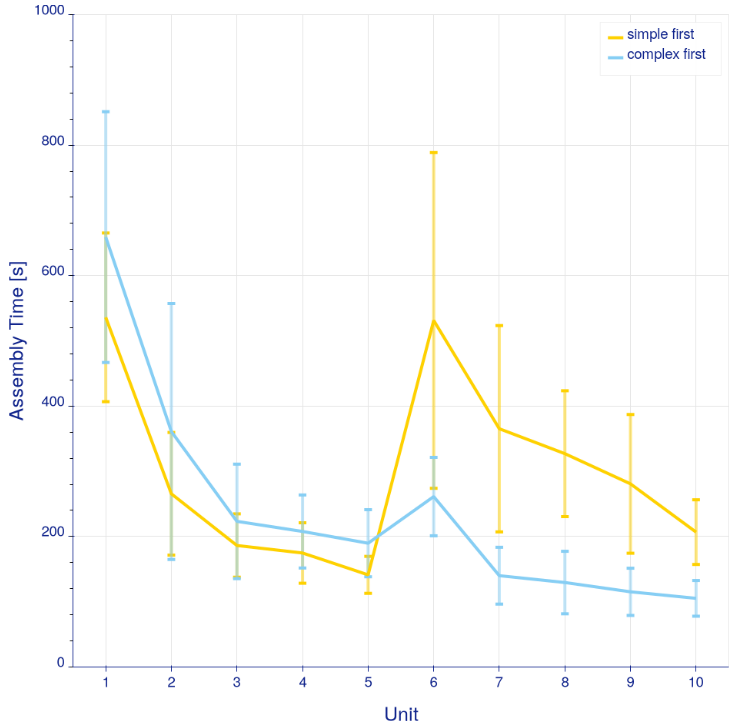


Figure 5: Mean and standard deviation for both groups. Yellow corresponds to the group that built V_S first, V_C second and blue to the other.

5. The Relationship between Complexity, Change and Learning Curves

The results of the experiments give some preliminary insight into the effect that a change in product architecture has on performance as captured in two consecutive learning curves.

5.1 Complexity and Performance

Sinha and de Weck (2016) found that cycle time related supra-linearly to complexity in an experiment where participants assembled molecule structures from a chemistry modeling kit: Higher complexity was associated with lower performance. In our exploratory experiment, the number of participants was not large enough to statistically validate this relationship between product complexity and performance in general. Nonetheless, several features of the observed learning curves seem to relate to product complexity in different ways.

5.2 Change and the Jump between Learning Curves

The experiment showed that practicing with the complex V_C reduced cycle time for the simple V_S significantly, when compared to building V_S without any previous experience. Interestingly, this effect was not found the other way around: Experience with V_S did not improve cycle time for V_C. An explanation can be sought in the fact that V_C was mostly an extension of V_S, adding components. Moving from V_S to V_C constituted an introduction of new parts and it took participants time to figure out how to assemble them. Going from V_C to V_S on the other hand, mostly meant a removal of components, having little impact on performance.

The study seems to suggest that the jump between the end-point of a learning curve and the starting point of its successor relates to the newness of the system after change. The more complex the part of a product's architecture that is new to the manufacturing party, the more effort required to get the product to perform as intended—decreasing performance. This causes a sudden dip in performance from one learning curve to the next. Complexity in novel architecture might make it more likely for unexpected problems to appear. This idea is consistent with systems engineering literature on complexity, which suggests that emergent system properties cause issues when complexity rises (Anderson and Joglekar, 2012; Suh, 2005; Fisher, 2006). We therefore propose:

P1: <u>Novel</u> product complexity decreases manufacturing performance directly after product introduction, causing a sudden dip in performance from one learning curve to the next.

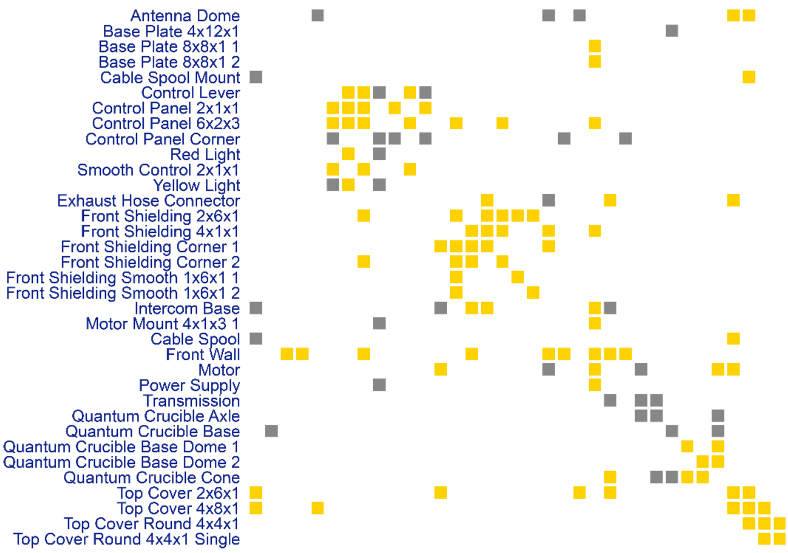
5.3 Complexity and the Asymptote of a Learning Curve

Participants mentioned that picking up parts became limiting to their performance after a few builds, especially for the simpler V_S. Once they had figured out an efficient assembly strategy, which might have been sooner for V_S since it was simpler, they appeared to run into physical limitations to the assembly process. The impact of novelty on manufacturing performance therefore seems to fade away as experience grows. When the emergent issues that arise with a new product introduction are dealt with and an assembly strategy has been figured out, performance moves to a lower bound asymptote value. This asymptote to performance might again relate to product complexity: Since complexity rises with the number of components and interfaces of a system, higher product complexity could mean more manufacturing actions, a larger risk of errors, higher costs and a longer cycle time. This is in accordance with the findings of Sinha and de Weck (2016) who saw performance decrease with complexity. While assembly time in the experiment did not significantly differ between variants when averaged over all attempts of all participants, the final (fifth) attempt to assemble V_C took more time on average than that of the V_S. This might indicate a higher asymptote to cycle time and hence a lower asymptote to performance for the more complex product. In other words: given an equal amount of manufacturing experience, performance is worse for a complex product than for a simple product and the maximum attainable performance is lower for complex products than for simpler ones. This would also agree with the findings of Sinha and de Weck (2016) who saw performance decrease with complexity. We therefore propose:

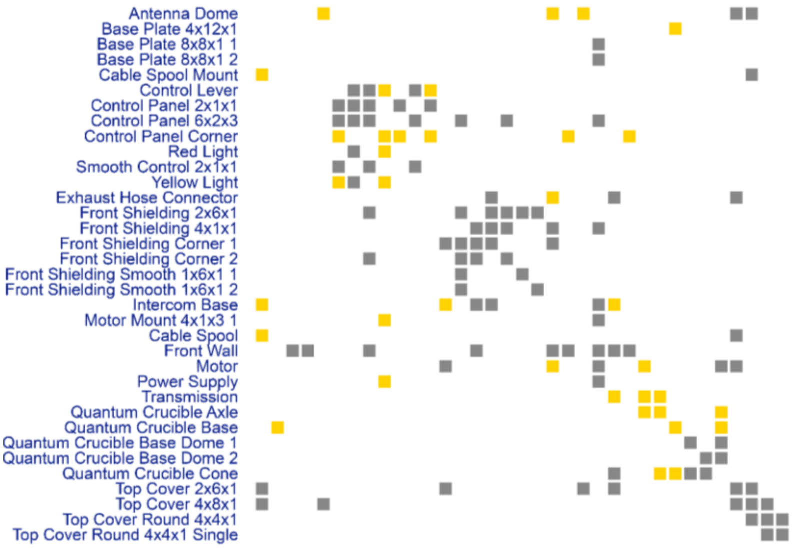
P2: Product complexity relates negatively to the lower bound asymptote of the learning curve in manufacturing performance.

6. Quantifying the Complexity of Change

Figure 6 shows the Delta DSMs of both variants of product change in the experiment. Since the Lego pieces were not physically modified, they contain no changed components or interfaces. The only difference between both Delta DSMs is that architecture that is new in 7a is removed in 7b and vice-versa. Weighing new and removed components and interfaces equally, the complexity of both DSMs is the same following the metric in equation (1). For example, assigning a complexity value of 1 to all components and interfaces, the Delta DSMs have a complexity of 186. The experimental results in Figure 5 however show a clear difference in performance between both experimental cases, indicating that removals impact manufacturing performance differently than additions or changes.



(a) V_S to V_C.



(b) from V_C to V_S.

Figure 6: Delta DSMs modeling the change from V_S to V_C (a) and vice-versa (b). New components and interfaces are shown in yellow and removed architecture in grey. There are no changed components.

Since new, removed, and common parts of a product's architecture after change all seem to impact manufacturing performance differently, we propose the 'Complement DSM', or '\-DSM' to solely capture the unique architecture of either of two products that have a degree of commonality between them. This Complement DSM allows separately quantifying the complexity of the unique parts of both versions of a product. Similarly, the common part of two products can be captured in an 'Intersection DSM' or '∩-DSM', including only those components and interfaces which two versions of a product share and which do not change. With two Complement DSMs and one Intersection DSMs, the respective complexity of old, new and common architecture can be distinguished for any product change. This allows separately investigating the influence of the complexity of each of these parts of product on learning curves in manufacturing performance.

Figure 7a shows the Complement DSM for the change from V_S to V_C in the experiment and Figure 7b shows the opposite case. Moving from the simpler V_S to the more complex V_C introduced more new components and interfaces than moving from V_C to V_S, as evident from the difference between 7a and 7b. Both contain only new and no changed components and interfaces, since no Lego components were physically altered. The Intersection DSM containing common components and interfaces between both V_S and V_C is shown in Figure 7c.

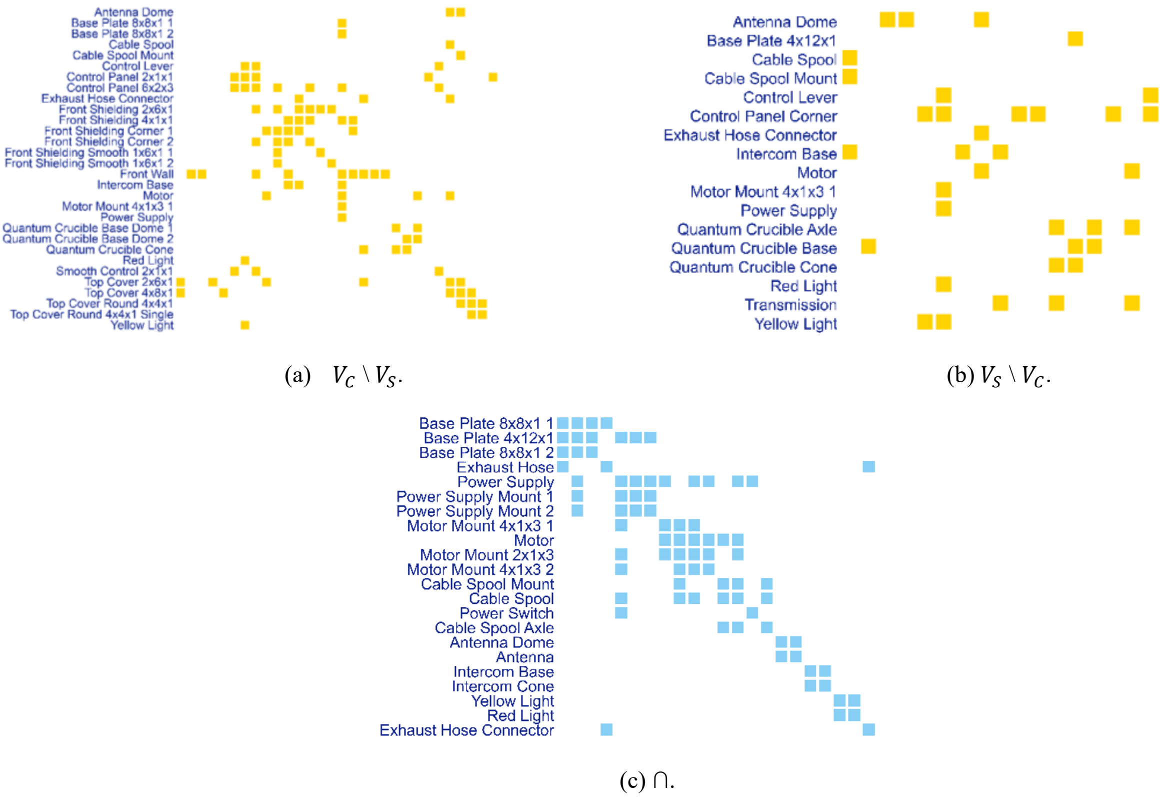


Figure 7: Complement DSMs showing only new components and interfaces switching from V_S to V_C (a) and vice-versa (b), and the Intersection DSM showing only components and interfaces common to both architectures (c).

Discussion & Future Research

In this paper, we investigate the relationship between learning curves and product changes by quantifying product complexity using DSM-based methods. This fits into a broader stream of research to identify the shape parameters of organizational learning curves. In particular, we considered product complexity and the overlap between products with regards to consecutive learning curves before and after a product change. We found that the complexity metric developed by Sinha (2014) is an effective measure to distinguish the complexity of two similar products.

The study takes an exploratory approach, using a small assembly experiment with individuals, resulting in two promising propositions to further investigate in follow-up work. Specifically, we propose that the complexity of the novel part of product architecture is responsible for the initial decrease in manufacturing performance after a product change. Second, we propose that the asymptote of a learning curve and the complexity of the DSM representing a product's architecture are inversely related: Manufacturing performance is worse for more complex products than for simple products, given an equal amount of experience. The framework based on these propositions might help estimate the extent of the disruption to manufacturing performance caused by product changes in organizational practice. An increased understanding of the impact of product changes might lead to better production planning, improved resource allocation, or market entry decisions by organizations.

While the Lego assembly experiment here was explorative and limited in its number of participants, the resulting patterns in performance are also observed in organizational practice. We conclude the findings merit further investigation and validation of the proposed ideas. Experiments similar to that of Sinha and De Weck (2016) and the one described here provide an excellent method for experimental research. Further research could separately vary the level of overlap and the difference of complexity between two product versions. It might also be fruitful to explore the relationship between product complexity and the steepness of learning curves, since this is an important characteristic of learning curves that we were not able to study.

Future case studies and natural experiments in organizational settings can help judge the generalizability of experimental research findings with individual participants to organizational learning. Research could also expand to include products with a large software component, to see whether the link to manufacturing performance is present there as well. Finally, while we focused on quantifying product complexity and change, the methods employed here might also enable research into the complexity of and change in requirements, processes, or organizations themselves.

References

Anderson, E. G. and Joglekar, N. R. (2012). The Innovation Butterfly. Springer. Anzanello, M. J. and Fogliatto, F. S. (2010). Scheduling learning dependent jobs in customised assembly lines. International Journal of Production Research, 48(22):6683– 6699.

Argote, L. (2013). Organizational Learning. Springer, New York, second edition.

Argote, L., Insko, C. A., Yovetich, N., and Romero, A. A. (1995). Group Learning Curves: The Effects of Turnover and Task Complexity on Group Performance. Journal of Applied Social Psychology, 25(6):512–529.

Argote, L. and Miron-Spektor, E. (2011). Organizational learning: From experience to knowledge. Organization Science, 22(5):1123–1137.

Blanchard, B. S. and Fabrycky, W. J. (2011). Systems Engineering and Analysis. Pearson, 5th edition.

Browning, T. R. (2016). Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. IEEE Transactions on Engineering Management, 63(1):27–52.

Browning, T. R. and Heath, R. D. (2009). Reconceptualizing the effects of lean on production costs with evidence from the F-22 program. Journal of Operations Management, 27(1):23–44.

Chapman, R. and Hyland, P. (2004). Complexity and learning behaviors in product innovation. Technovation, 24(7):553–561.

Clift, T. B. and Vandenbosch, M. B. (1999). Project complexity and efforts to reduce product development cycle time. Journal of Business Research, 45(2):187–198.

Dutton, J. M. and Thomas, A. (1984). Treating Progress Functions as a Managerial Opportunity. Academy of Management Review, 9(2):235–247.

Eckert, C., de Weck, O., Keller, R., and Clarkson, P. J. (2009). Engineering Change: Drivers, Sources and Approaches in Industry. In International Conference on Engineering Design, pages 47–58, Stanford, CA.

Eppinger, S. D. and Browning, T. R. (2012). Design Structure Matrix Methods and Applications. The MIT Press.

Fisher, D. (2006). An emergent perspective on interoperation in systems of systems. Technical Report March, Carnegie Mellon University.

Gopal, A., Goyal, M., Netessine, S., and Reindorp, M. (2013). The Impact of New Product Introduction on Plant Productivity in the North American Automotive Industry. Management Science, 59(10):2217–2236.

Griffin, A. (1997a). Modeling and measuring product development cycle time across industries. Journal of Engineering and Technology Management - JET-M, 14(1):1–24.

Griffin, A. (1997b). The effect of project and process characteristics on product development cycle time. Journal of Marketing Research, 34(1):24–35.

Jacobs, M. A. and Swink, M. (2011). Product portfolio architectural complexity and operational performance: Incorporating the roles of learning and fixed assets. Journal of Operations Management, 29(7-8):677–691.

Jarratt, T. A., Eckert, C. M., Caldwell, N. H., and Clarkson, P. J. (2011). Engineering change: An overview and perspective on the literature. Research in Engineering Design, 22(2):103–124.

Lapré, M. A. and Nembhard, I. M. (2011). Inside the organizational learning curve: Understanding the organizational learning process. Foundations and Trends in Technology, Information and Operations Management, 4(1):1–103.

Li, X., Shi, Y., and Gutman, I. (2012). Graph Energy. Springer.

Maier, M. W. and Rechtin, E. (2009). The art systems of architecting. CRC Press. McNerney, J., Doyne Farmer, J., Redner, S., and Trancik, J. E. (2011). Role of design complexity in technology improvement. Proceedings of the National Academy of Sciences of the United States of America, 108(22):9008–9013.

Muth, J. F. (1986). Search Theory and the Manufacturing Progress Function. Management Science, 32(8):948–962.

Pruett, M. and Thomas, H. (2008). Experience-based learning in innovation and production. R and D Management, 38(2):141–153.

Rosiello, A. and Maleki, A. (2021). A dynamic multi-sector analysis of technological catch- up: The impact of technology cycle times, knowledge base complexity and variety. Research Policy, 50(3):104194.

Shafiei-Monfared, S. and Jenab, K. (2011). Complexity analysis of an operation in demand-based manufacturing. International Journal of Production Research, 49(17):5303–5315.

Simon, H. A. (1996). The Sciences of the Artificial. MIT Press.

Simonin, B. L. (1999). Ambiguity and the process of knowledge transfer in strategic alliances. Strategic Management Journal, 20(7):595–623.

Sinha, K. (2014). Strucutal Complexity and its Implications for Design of Cyber-Physical Systems. PhD thesis, Massachusetts Institute of Technology.

Sinha, K. and De Weck, O. L. (2012). Structural complexity metric for engineered complex systems and its application. In Gain Competitive Advantage by Managing Complexity - Proceedings of the 14th International Dependency and Structure Modelling Conference, DSM 2012, number September, pages 181–192.

Sinha, K. and de Weck, O. L. (2016). Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems. Systems Engineering, 19(3):193–206.

Sinha, K., Omer, H., and De Weck, O. L. (2013). Structural complexity: Quantification, validation and its systemic implications for engineered complex systems. Proceedings of the International Conference on Engineering Design, ICED, 4 DS75-04(January):189– 198.

Smaling, R. and De Weck, O. L. (2007). Assessing Risks and Opportunities of Technology Infusion in System Design. Systems Engineering, 10(1):1–25.

Stuart, I., McCutcheon, D., Handfield, R., McLachlin, R., and Samson, D. (2002). Effective case research in operations management: A process perspective. Journal of Operations Management, 20(5):419–433.

Suh, N. P. (2005). Complexity. Oxford University Press, 1st edition.

Thompson, P. (2012). The relationship between unit cost and cumulative quantity and the evidence for organizational learning-by-doing. Journal of Economic Perspectives, 26(3):203–224.

Walden, D. D., Roedler, G. J., Forsberg, K. J., Hamelin, D. R., and Shortell, T. M., editors (2015). INCOSE Systems Engineering Handbook. INCOSE, 4 edition.

Wright, T. P. (1936). Factors Affecting the Cost of Airplanes. Journal of the Aeronautical Sciences, 3:122–128.
Yin, R. K. (2003). Case Study Research, volume 5. SAGE Publications Inc., third edit edition.
Zirger, B. J. and Hartley, J. L. (1994). A conceptual model of product development cycle time. Journal of Engineering and Technology Management, 11(3-4):229–251.

Contact: T. Dooper, FruitPunch AI, High Tech Campus 6a, 5656AE, Eindhoven, The Netherlands, +31649985539, tjommedooper@gmail.com, www.fruitpunch.ai

Tjomme Dooper

Tjomme holds two MSc degrees from Eindhoven University of Technology, on in mechanical engineering and the other in innovation management. He is currently the Head of Partnerships and Growth at FruitPunch AI. After obtaining his BSc in Mechanical Engineering, he worked in several foundations, associations and companies during his MSc program. From recruiter to board secretary, from business developer to marketeer, through many different part-time functions he gained an appreciation for the daily business of many types of organizations. During his time as a graduate student in the Control Systems Technology group at the faculty of Mechanical Engineering, he further broadened his knowledge base by following a dual degree program. At the faculty of Industrial Engineering and Innovation Sciences, he graduated in the Innovation, Technology, Entrepreneurship and Marketing group. He wrote his thesis bridging systems engineering and operational management domains, by investigating the complexity of engineering changes and their effect on organizational learning. After graduating, he started as Head of Partnerships and Growth at FruitPunch AI, an education and recruitment startup. There, he combines his knowledge of learning, innovation and complex technology with his experience in dealing with people and organizations of all sorts.

Pascal Etman

Pascal is an Associate Professor of Mechanical Engineering in the Control Systems Technology group at Eindhoven University of Technology (TU/e). His research field of interest is system design and optimization. Together with his co-workers he is developing methods and tools for model-based design and optimization of complex (mechanical) engineering systems. His scientific achievements include approximation methods for structural optimization, the Augmented Lagrangian Coordination (ALC) method for distributed optimum design, and the Effective Process Time (EPT) as aggregate modeling method for performance analysis of discrete manufacturing systems. His current research focuses on design optimization, design structure matrix modeling, and modeling concepts for systems engineering. Applications include infrastructure systems, mechatronic systems, and additive manufacturing systems. Pascal has (co-)authored journal publications in Structural and Multidisciplinary Optimization, ASME Journal of Mechanical Design, Research in Engineering Design, IEEE Transactions on Semiconductor Manufacturing, and International Journal for Numerical Methods in Engineering, among others. He teaches a graduate course on engineering optimization introducing numerical methods for design optimization, and a graduate course on design structure matrix modeling for analysis of integrated systems design. In the bachelor studies, Pascal is involved in a design project for third year undergraduate mechanical engineering students.

Alex Alblas

Alex is an Assistant Professor of Product and Process Innovation in the Innovation, Technology Entrepreneurship & Marketing (ITEM) group of the School of Industrial Engineering at Eindhoven University of Technology. Alex's research activity focused on how organizations can successfully manage the invention, development and launch of new products. His ambition is to contribute to the challenges and mechanisms related to enhancing and speeding up learning curves in new product development. In this quest he is currently investigating the learning curve effects of design change, caused by design iteration, design debugging, and evolving insights, and the resulting process changes. He has a special interest in the operational practices of new product development that lead to enhanced performance. In one research project he investigates the learning mechanisms in which agile operations and iterative product development can lead to performance improvement. In another big industry funded research project, of which Alex is the project leader, he investigates together with a team of researchers the practices in which product development and customer service processes enable service improvements.