

DESIGN FOR ASSEMBLY LINE PERFORMANCE: THE LINK BETWEEN DFA METRICS AND ASSEMBLY LINE PERFORMANCE METRICS

Marcos Esterman Jr.

Raghava Gujja

Ruben Proano

Krishna Kamath

Andres Carrano

Rochester Institute of Technology

Industrial and Systems

Engineering

Rochester, NY, USA

ABSTRACT

While DFA has been a useful design tool, it is not explicitly linked to actual manufacturing line performance. The motivation for this research came from the desire to link DFA directly to line balance and cycle time performance. This paper will build on previous work to link DFA and manufacturing analysis methods so that the manufacturing parameters can be estimated and used to guide the design of a product. Our current work to estimate the impact that design changes could have to reduce cycle time and improve line balancing performance will be presented.

1. INTRODUCTION

1.1. Background

In the past, design and manufacturing were often treated as two independent areas of operation within a company. Today, the practice of concurrent engineering is more prevalent, and the proverbial "wall" between designers and the factory floor is less common. Industry has realized that in order to improve productivity they need to bridge the gap between the two areas. This thinking has given rise to methodologies like Design for Assembly (DFA) and Design for Manufacturing, which are two of the earliest known tools in the Design for X toolset. DFA is an approach to design products that increase the ease of assembly of a given product, resulting in faster and more cost efficient assembly processes. This in turn adds value for the customer and results in higher profit for the manufacturer.

DFA tools breakdown the assembly into discrete operations where the parts, the handling, the insertion, and the processing activities are evaluated according to stability, directionality, manipulability and other difficulties [1]. Vincent and Filippo [2] define DFA as "*a process for improving product design for easy and low-cost assembly, focusing on functionality and on assemblability concurrently.*"

1.2. Motivation

Researchers are constantly trying to generate new ideas to bridge the gap between design and manufacturing. Early works include Pahl and Bietz [3] and Bralla [4]. The field has grown rapidly and is broadly known as Design for X, or DFX, where

the X represents distinct life-cycle considerations, such as environment, supply-chain and reliability. Though DFA was one of the earliest tools developed, researchers continually work to improve the tool (e.g. [5, 6]).

This research is focused on exploring the link between DFA metrics and actual assembly line performance. The inspiration for this research came from the observation that DFA does not explicitly consider the issues of line balancing and cycle time. The natural question arose as to whether these issues could be considered at the design stage by utilizing the metrics that were derived from a DFA analysis. From the literature review in the next section it is observed that there is relatively little work that has been done on this topic. However, it is known that the time required to assemble a product can be estimated from the DFA analysis. The time required to manufacture a product can also be estimated from a manufacturing analysis. It would be interesting to see if the manufacturing parameters could be estimated from the DFA analysis. In turn, this would allow assembly line performance issues to be considered during the design stage of the product.

2. LITERATURE REVIEW

The literature in the area of DFA is vast but most of the work is focused on either modifying or improving the existing DFA methods. This section provides an overview of the different DFA methods. A section on the research related to DFA and product development has also been included, as has related research area on the linking of DFA to manufacturing analysis.

2.1. DFA Methods

The development of DFA started in the early 1960's to help designers consider the assembly problems at the design stage of the product [7]. During the 1980 to the 1990 period there were many variations proposed to the then existing DFA methodologies, namely, the Westinghouse method and several others which were based on the original DFA method [7].

Boothroyd and Dewhurst

Boothroyd and Dewhurst [8] have developed two forms of DFA. The first is the DFA handbook which contains charts and

worksheets [8]. The second codifies this knowledge into a commercial software package [9]. The two approaches use the same calculation methodology to derive the DFA metrics.

The Boothroyd and Dewhurst DFA technique employs a DFA handbook, which gives equations and the extensive data necessary to estimate manufacturing and assembly cost during product design. This method is based on two principles: the application of criteria to each part to determine if it should be separate from all other parts and the estimation of the handling and assembly costs for each part using the appropriate assembly process. Tables and charts are used to estimate the part handling and part insertion time. These tables are based on a two-digit code that is in turn based on a part's size, weight, and geometric characteristics.

Hitachi Assembly Evaluation Method (AEM)

Ohashi et al. [10] state that the Hitachi AEM is the first method for assembly-productivity evaluation and that it has been used to achieve great cost reductions. Using this method, in the early design stage product design quality is analyzed quantitatively and the weaknesses in the design's assembly producibility are highlighted. In addition, the effects of design improvements are confirmed with respect to assembly cost.

This method is based on the principle of "one motion for one part." For more complicated motions, a point-loss standard is used and the ease of assembly of the whole product is evaluated by subtracting points lost. The method was originally developed in order to rate assemblies for ease of automatic assembly.

Assembly time (AT) is measured in T-downs. One T-down is the time taken for one downward movement with a part. This method helps in an accurate understanding and comparison of assembly time and cost. Also, the AEM score is not closely related to the estimated part attachment operation cost.

Westinghouse DFA

The Westinghouse DFA method has been widely used, as it considers many factors and their interaction while analyzing the time required for assembly. The simplified Westinghouse Method involves the following steps [11]:

- Step 1: Identify each part in the assembly and document the functions of each part.
- Step 2: Disassemble the product and carefully record the sequence of disassembly.
- Step 3: Identify the parts that provide the functions detailed in Step 1. Explain how each function is achieved, using small sketches when needed to provide additional clarity.
- Step 4: Create a fishbone diagram to provide a visual representation of assembly subassembly sequences.
- Step 5: Determine the assembly times for each subassembly sequence and for the final assembly sequence.
- Step 6: Produce a Pareto chart of assembly times for all operations and the cumulative percentage of assembly time.

Step7: Identify the areas of redesign, focusing on parts integration.

The Westinghouse DFA is particularly useful to help reduce part count and to reduce time to assemble, thereby reducing assembly costs. In addition, it improves design features which make it easier to grasp, move, orient and insert parts. The reduction of the number of parts in an assembly has the added benefit of generally reducing the total cost of parts in the assembly.

2.2. Advanced DFA Research

Coma et al. [12] establishes a new concept called "FuzzyDFA" based upon the fuzzy logic principles that deal with the uncertainties of a designer. This technique may be used at early design phase where the product design can be optimized using the DFA methodology. It implements geometric algorithms to evaluate the assembly process automatically.

Wu and Xie [13] proposed to create a linkage between design data in CAD with assembly operations in CAM. They introduced Open Structured Assembly Coding System (OSACS) a virtual coding system in order to reduce assembly costs. This system uses a virtual environment such as a CAD file in order to identify various features of the CAD model and visualize the part mating operations. The extractor is used to identify specific model codes to represent assembly operations in CAM. The proposed design depicts significant cost savings and also connects the CAD/CAM phases.

Additional useful insights come from various works like Stone et al. and Gupta and Okudan [5, 14]. They introduced the idea of integrating conceptual design and DFA. These product architecture-based conceptual DFA analyses can be used to accelerate the rate of product improvement, or perhaps achieve a fully mature design in a first product offering.

2.3. Research Linking DFA and Manufacturing

The literature in the area of linking DFA to manufacturing analysis is fairly limited. This field is of significant interest today, as design engineers and manufacturing engineers have realized that bridging the gap between design and manufacturing will yield the best results.

Caputo and Pelagagge [15] had a vision to evaluate the effect of product features on the presentation of manufacturing lines. They provided a specific rating to assembly components with respect to an assembly line. Caputo and Pelagagge [15] selects four distinct product features, mainly:-

1. The number of assembly tasks to be performed;
2. The average number of DOF in the assignment of a task to a station given the precedence constraints in the assembly sequence;
3. The ratio of average task time to the maximum task time, t_{avg}/t_{max} ;
4. The ratio of the maximum task time to the cycle time t_{max}/t_c .

To generate various scenarios, Caputo and Pelagagge [15] manipulated each of the four features at various levels. The scenarios led them to suggest guidelines for the designers based on the four features. This paper only considers average degrees of freedom so it qualitatively rates each product feature and suggests design recommendations. Also, it does not focus on finding the best solution for line balancing and cycle time simultaneously.

The Designers' Sandpit is a research project that aims to address these issues by developing an environment for "Assembly Oriented Design," incorporating methods for the generation and evaluation of concept design ideas, assembly planning and design advice [16]. The project is a collaborative effort between the Universities of Hull and Cranfield and a CAD software developer, Radan Computational Ltd. Their DFA research is focused on the following areas:

- *Proactive DFA* - Integration of tools into the design process to enable product developers to be proactive rather than reactive.
- *Concept design* - The integration of DFA principles into the earliest stages of product development.
- *Geometric Reasoning* - Reduction of time and subjectivity by the automatic extraction of data already available within the CAD model of a product design.
- *Product complexity* - Goal is to provide product developers with information about product complexity.

Lambert [17] focuses on the importance of line balancing problems and the planning of assembly sequence. According to Lambert, sequence planning means developing the various steps that lead to assembly of a product. He establishes the use of precedence graphs for line balancing. These precedence graphs help to plan the assembly sequence and to select the optimum assembly sequence.

While there has been significant advancement of the DFA methodology over the years, no explicit link has been established between this DFA analysis and manufacturing analysis, particularly line balancing and cycle time.

3. PREVIOUS WORK

3.1. Research Vision

The original objective of this research was to take the DFA analysis a step further to consider throughput or cycle time and line balancing. The following research questions were explored:

- Can an explicit link between DFA and assembly line performance be made?
- If so, can this link be leveraged to provide a method to aid product development practitioners make development decisions?
- What kind of design actions can be taken to optimize the cycle time given an initial design candidate?

3.2. Overview

This section will provide a prescriptive summary of the steps involved in the first attempt to propose a methodology.

- STEP 1. *Representation of the design candidate*: a detailed enough representation of the design needs to be generated so that components and the precedence relations of these components can be understood. Note that this does not necessarily imply that a detailed design needs to be in place. As in the work of Stone et al. [5], a functional module could be defined as a 'component' that is yet to be defined.
- STEP 2. *Fishbone diagram*: from the representation in step 1, an assembly fishbone diagram is created to show the overall assembly structure of the product and its precedence relationships.
- STEP 3. *DFA analysis*: the actual DFA analysis is performed which will generate the following information: Total assembly time; Time for each operation; Number of repetitions of each operation; Repetition time
- STEP 4. *Manufacturing analysis*: these times and operations from the DFA analysis become inputs to a simulation model. Also, the precedence constraints from the design stage are inputs to the model. In the simulation stage the line is balanced such that the number of workstations is minimal while maintaining the production rate (takt time) and precedence constraints. This is done by software using the COMSOAL algorithm (computer method for sequencing operations for assembly lines) [18]. This serves as a baseline measure of the design.
- STEP 5. *Relaxing precedence relationships*: the precedence constraints are systematically relaxed. During this step, a line balance index and cycle time index are developed to identify components that are candidates for re-design.
- STEP 6. *Redesign action*: once possible opportunities are identified an improved design can be generated.

In order to illustrate the methodology, this process is implemented on a brake assembly case study which is described in the next section.

3.3. Case Study: Brake Assembly

Step 1: Design Candidate: Brake Assembly

The exploded view of the brake assembly is shown in Figure 1. The assembly has 12 unique parts 23 total parts.

Steps 2 & 3: Generate Fishbone Diagram and DFA Analysis

Figure 2 shows the assembly sequence for the brake assembly that was used in the DFA analysis. It also lists the necessary operations needed for assembly which is useful when completing the DFA analysis. For space reasons, the details of the DFA table are not shown, but the assembly times for each

component as well as the total assembly time (132.4 sec.) were derived for this assembly with the aid of this fishbone.

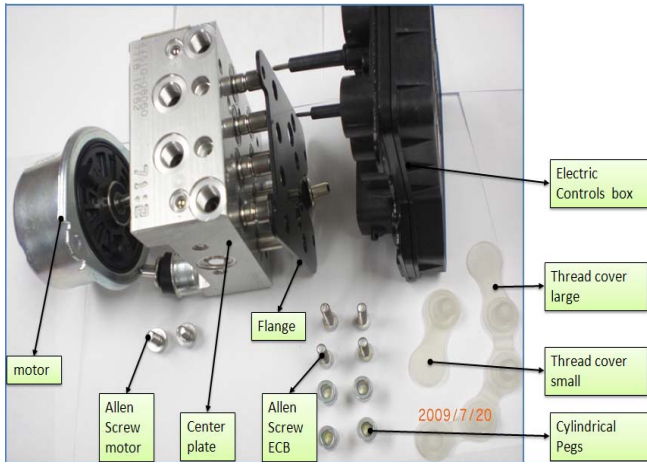


Figure 1: EXPLODED VIEW OF THE BRAKE ASSEMBLY

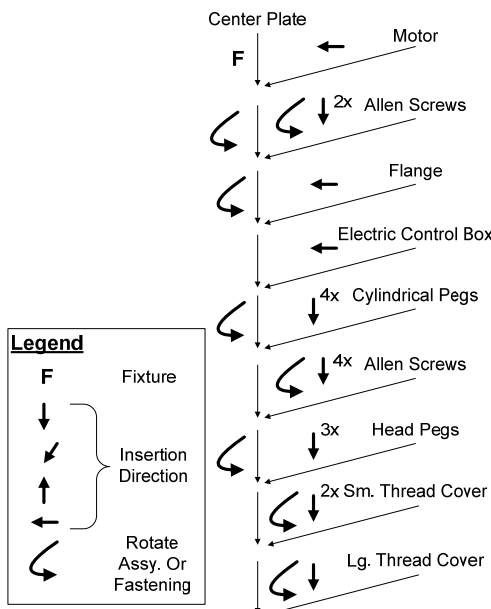


Figure 2: FISHBONE DIAGRAM OF BRAKE ASSEMBLY

Step 4: Manufacturing Analysis

After the DFA analysis phase, the assembly steps are distributed among the workstations depending on the takt time using the COMSOAL algorithm. In this case the takt time was assumed to be 32 seconds. The Flexible Line Balancing software (demonstration version) [19] was used for the purpose of line balancing, which uses the COMSOAL algorithm to balance the line. The output from this program is shown in Figure 3.

The number of operations from DFA, the operation time and the precedence constraints are entered into the Flexible Line Balancing software. The algorithm determines the

optimum assembly sequence based on the inputs and the optimum line balance is obtained according to takt time.

No	Workstation	Elem Task Description	ST	Elem Task #	Prec
1		motor	6.6	1	
2	1	centerplate	0.5	2	
3		allen screw	24.2	3	1,2
4		electric controls box	3.2	5	2,4
5	2	flange	4.9	4	2
6		cylindrical pegs	14.4	6	1,2,3,4,5
7	3	allen screw ECB	24.2	7	2,4,5
8	4	allen screw ECB	24.2	8	2,4,5
9		head pegs	14.4	9	1,2,3,4,5,6,7,8
10	5	thread cover small	9.2	10	1,2,3,4,5,6,7,8,9
11		thread cover large	6.6	11	1,2,3,4,5,6,7,8,9

Figure 3: FLEXIBLE LINE BALANCING SOFTWARE

When the model is run with the takt time of 32 seconds the Line Balancing output that is obtained is shown in Figure 4.

- Number of workstations = 5
- Takt time = 32seconds
- Neck time = 31.3 seconds
- Number of operators = 5 (The algorithm assumes one operator per workstation.)

It is interesting to note that the optimum assembly sequence in this case is:

1->2->3->5->4->6->7->8->9->10->11

Also, the task distribution at each workstation is shown in Figure 4. At workstation 1 component 1; component 2 and component 3 are performed. Similarly, the other tasks are distributed among the workstations as shown in Figure 4. The line balance efficiency, which is calculated using the neck time, is 84.6%, but is not a metric used in this work.

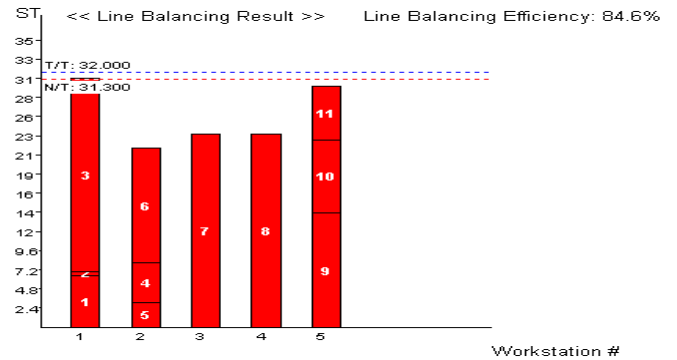


Figure 4: LINE BALANCING OUTPUT OF BRAKE ASSEMBLY

Thus, after balancing the line, the following optimum outputs are generated according to the takt time:

1. The number of workstations
2. Time at each workstation
3. Optimum sequence of the assembly
4. Distribution of task at each workstation.

Step 5: Relaxing Precedence Relationships

Given a baseline design, it would be useful to have a set of metrics that would identify areas in the existing design that would lead to improved manufacturing performance, specifically line balancing and cycle time. To accomplish this, a line balancing index and a cycle time index were developed. The basic idea behind these indices is that the precedence

constraints of the existing design would be relaxed in a systematic manner. However, with each systematic change, a comparison to the baseline is made so that metrics can be compiled to identify the assembly operations that would have the greatest likelihood of improving either cycle time or line balancing performance, should a design change be feasible.

Thus, the development indices are derived as follows:

Cycle Time Index

From the definition of Cycle time:

$$CT = NT \times Num_{workstations} \tag{1}$$

where,

NT = Neck Time = workstation which takes longest time

Num_{Workstation} = number of workstations

Hence the Cycle Time Index is calculated as follows:

$$CTI = (1 - (\frac{CT_p}{CT_{Baseline}})) \tag{2}$$

where,

CTI = Cycle Time Index

CT_p = Cycle time of a given Permutation

CT_{Baseline} = Cycle time of assembly line considering all components of the original design

The higher the index the more the improvement in cycle time meaning a reduction in cycle time. Note that the cycle time index may be negative. A negative index indicates an increase in cycle time as compared to the baseline. Thus this new alternative is worse than the original design, which is not of interest. When there is an improvement in the cycle time relative to the original design, then the cycle time index will be bounded between 0 and 1. It is these cases which are of most interest.

Line Balance Index

From the Line Balancing result,

$$SSTKE = \sum_{i=1}^n (TT - Wkstn_i)^2 \tag{3}$$

where,

SSTKE = Sum of Squares of the Takt Time Error
= the sum of squares of the difference between the work station time and the takt time.

TT = Takt Time

Wkstn_i = ith workstation assembly time

$$SST = \sum_{i=1}^n (Wkstn_i)^2 \tag{4}$$

where,

SST = Total Sum of Squares

Wkstn_i = ith workstation assembly time

Thus, the line balance index is as follows:

$$LBI = 1 - \frac{SSTKE}{SST} \tag{5}$$

where,

LBI = Line Balance Index

SSTKE = Sum of Squares of the Takt Time Error

SST = Total Sum of Squares

These calculations are done for each of the tasks by relaxing the precedence constraint on one task at a time. For a given precedence constraint, there is a component at the receiving end at the supplying end. This leads to two approaches for generating the permutations for relaxing the precedence constraints: row permutations and the column permutations respectively. A row permutation relaxes the constraint on the receiving end of the component or any dependencies that component may have. In a column permutation, the precedence constraint of a component supplying a constraint is relaxed on all components. A more detailed description of the calculations and permutations can be found in [20]. In order to perform systematic row and column analyses, a C++ program was developed to automate the index calculations.

Step 6: Redesign Action

Analyzing the Data

The brake assembly has a total of 11 tasks, which means that all combinations of these tasks result in 2¹¹ combinations of the tasks and similarly 2¹¹ combinations of the precedence constraints. Hence there are a total of 2048 possible ways in which the precedence constraints can be relaxed for the brake assembly. The responses that are tracked as each of the combinations are generated are the Line Balancing and Cycle Time Indices.

The resulting data are then analyzed in the statistical software package of Minitab to determine the significant factors and their interactions for each case, i.e., the Row Permutation case and the Column Permutation cases. The significant effects plot for both of these cases (the row permutation and the column permutation) is generated for the Line Balancing Index and Cycle Time Index.

Thus there are four plots of significant effects as follows:

Row Permutation:

- Significant effects plot of Line Balance Index
- Significant effects plot of Cycle Time Index

Column Permutation

- Significant effects plot of Line Balance Index
- Significant effects plot of Cycle Time Index

The significant effects plot of the row and column permutations help to identify the components and the interactions of the components that are the most significant. The significant effects plot for the response of the Line Balance Index is shown in Figure 5. The effects and interactions that are indicated in red are the “significant” factors in that they deviate from what would be expected from random variations. Strictly speaking this is not a valid statement as the results of this analysis are not generated from random processes, but this step does serves as screening step to reduce the number of component and component interactions to study to the ones

with the largest effects. These include KL; F and J interactions, which are most significant.

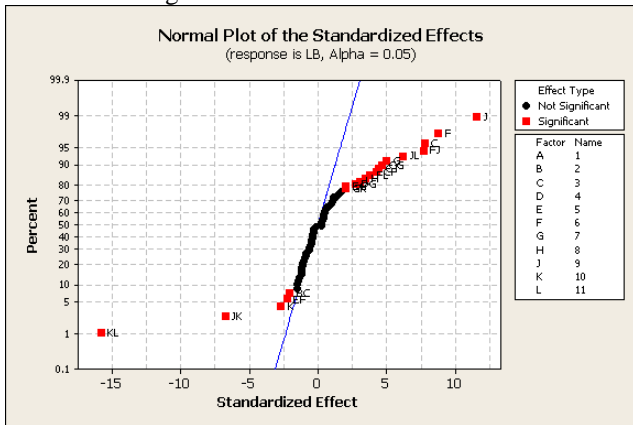


Figure 5: SIGNIFICANT EFFECTS PLOT OF LBI FOR ROW PERMUTATION

Identifying Components for Redesign

To identify the components for redesign, the effects of LBI are plotted versus the effects of the CTI. This is achieved by the following:

- Coefficients of significant effects of LBI.
- Coefficients of significant effects of CTI.

This is done for each case of the row permutation and the column permutation. Figure 6 shows this plot for the row permutation. Note that the red data points are high impact in both row and column. The yellow are high impact in the row permutations and the green are high in the column permutations only. This allows the factors of interest to be identified.

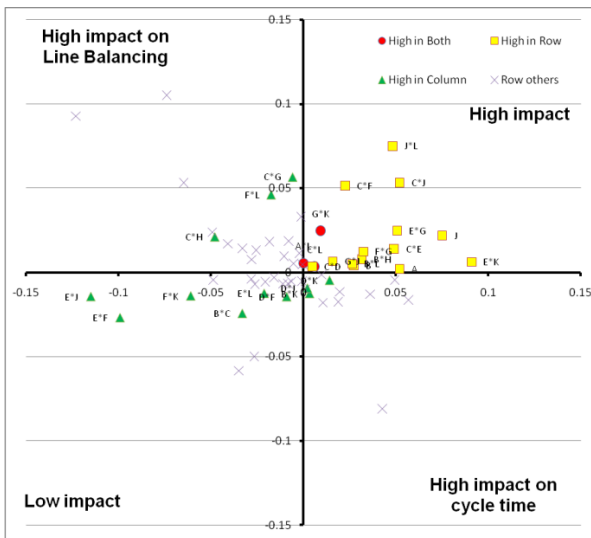


Figure 6: LBI VERSUS CTI FOR ROW PERMUTATION

In order to identify candidates for redesign the following steps are needed:

1. Prioritize the parts.
2. Draw the input/ output diagram of the most critical part.

3. Analyze this diagram for under constrained and over constrained precedence.
4. Refer to action table for design recommendations.

For the brake assembly the interaction G*K is selected since it is significant in both analyses. Looking at one component, G shown in Figure 7, one can analyze the inputs and outputs of the task. The input precedence constraints for the task G, i.e., Allen screw ECB is center plate, Flange and the Electric Controls box (ECB). The output constraints of task G are head pegs, thread cover small and thread cover large.

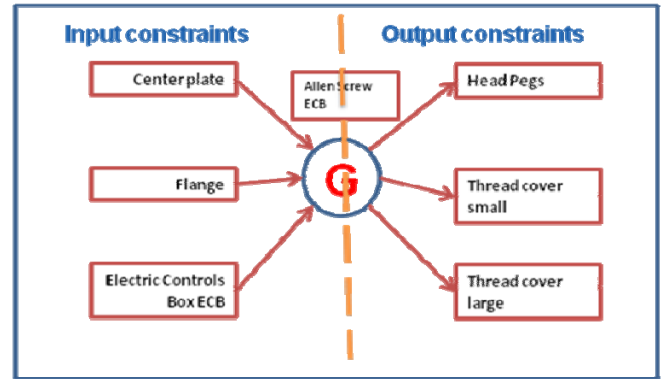


Figure 7: INPUT / OUTPUT PRECEDENCE CONSTRAINT FOR "G"

3.4. Lessons Learned

In this work, there was good progress made towards the initial objectives.

- A link between DFA and assembly line performance has been established.
- An analysis procedure was developed to systematically identify redesign components so that line balance and cycle time performance could be improved during the design stage.
- The potential utility of the approach was demonstrated through a case study.

While the above progress is encouraging, there are a number of problems that were identified that must be solved before this approach can be used on more realistic design problems. These problems suggest some research directions to be pursued to make such an approach feasible. These include:

- The use of a more efficient and effective line balancing algorithm
- A more efficient and effective search process to identify redesign candidates
- More systematic guidance on redesign actions based on the analysis results
- The validation of the methodology on more realistic case studies.

Each of these will be discussed in more detail in the next section.

4. CURRENT WORK & FUTURE RESEARCH DIRECTIONS

Based on the previous work and the opportunities discussed above, the main objective of the next phase of this research is to improve efficiency and effectiveness. To that end, this can be accomplished by two actions: (1) an improved line balancing algorithm; (2) reformulation of the optimization problem and search process so that the candidates for redesign can be more efficiently identified.

4.1. Line Balancing

The objective of line balancing is to minimize the idle time and to distribute the tasks to each workstation such that the pace at which goods are produced allows satisfying the product demand in a required takt time. Consequently, each workstation has a fixed amount of time, T , to complete all the tasks assigned to it. The assembly sequence and allocation of the tasks to the workstations is done ensuring any existing precedence relationships among the assembly tasks. As the tasks cannot be sub-divided, a single task can be assigned to only one particular station; yet tasks can be bundled as a sub-assembly component and treated as a single task. We propose using a line balancing model proposed by Thangavelu and Shetty [21].

Consider a set of tasks $J=\{1,2,3..n\}$ and a set of workstations $K=\{1,2,3..n\}$, where the time for completion of each task j is t_j , and each task j has a set of predecessor p_j , which must be performed in advance. Then the decision variable for this problem is given by

$$X_{jk} = \begin{cases} 1, & \text{if task } j \text{ is assigned to station } k \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

The maximum number of stations required for the assembly of a given product cannot exceed the number of tasks required in such process. Thus the maximum number stations to use is bounded by the number of tasks. This formulation assumes that setting a new station is expensive, and hence the main purpose of line balancing problem is to minimize the number of stations required to assemble the product. To induce this effect, the Thangavelu and Shetty propose a cost structure in which the cost of assigning a task to a station progressively increases as the station number increases. Thus, the formulation assumes that the cost of assigning a task to a station k , C_k , is the same for any task; and that this costs increase with the station number such that $nC_k \leq C_{k+1}$.

Therefore the resulting line balancing problem is given by:

$$\text{Minimize } Z = \sum_{i=1}^n \sum_{j=1}^n C_j X_{ij} \quad (7)$$

s.t.

$$\sum_{k=1}^n X_{jk} = 1, \quad \forall j \in J \quad (8)$$

$$\sum_{i=1}^n t_j X_{jk} \leq T, \quad \forall k \in K \quad (9)$$

$$X_{j_2 k_1} \leq \sum_{k=1}^{k_1} X_{j_1 k} T, \quad \forall k_1 \in K \quad (10)$$

$j_1 \in P_{j_2}, \text{ and } j_2 \in J$

$$X_{jk} \in \{0,1\} \quad \forall j \in J \text{ and } k \in K \quad (11)$$

The objective function (7) minimizes the total cost of assigning tasks to stations, and penalizes using large number of stations. Constraint (8) ensures that each task is assigned to only one station. Constraint (9) ensures that the time for processing all tasks assigned to each station does not exceed the maximum takt time T . Constraint (10) ensures the implementation of the precedence constraints. This constraint ensures that any task j_2 is assigned to a station k_1 only if all its predecessors have been already assigned to a previously opened station, or to the same station k_1 .

The above model was used to find the minimum number of stations required to assemble the brake assembly that was discussed in section 3, considering the following modifications:

- takt time was assumed to be 50 seconds
- the insertion of the allen screws was broken into 4 separate tasks

The tasks and precedence constraints are summarized in Table 1.

Table 1: Tasks and Precedence for Brake Assembly

Element Task number	Task decription	Time for each task	Precedence
1	Motor	6.6	-
2	Center plate	0.5	-
3	Allen screw	12.1	1,2
4	Allen screw	12.1	1,2
5	Allen screw	12.1	1,2
6	Allen screw	12.1	1,2
7	Flange	4.9	2
8	Electric control box	3.2	2,4
9	Cylindrical pegs	14.4	1,2,3,4,5
10	Allen screw motor	24.2	2,4,5
11	Head pegs	14.4	1,2,3,4,5,6,7
12	Thread cover small	9.2	1,2,3,4,5,6,7,8
13	Thread cover large	6.6	1,2,3,4,5,6,7,8

The proposed formulation was coded in AMPL and solved using a Gurobi solver. The optimal task-to-station assignment requires a minimum of 3 workstations, where tasks 1,2,3,4,5,8 are assigned to station 1, tasks 6,7,11,12,13 to station 2 and the tasks and tasks 9,10 are assigned to station 3. Table 2 shows the resulting processing time at each workstation.

Table 2- Time taken by each workstation

Workstation	Time Taken (sec)
1	43.4
2	47.2
3	38.6

The cycle time of the brake assembly was determined as follows:

$$\text{Cycle Time (CT)} = \text{Neck Time (NT)} \times \text{Number of workstations}$$

The neck time corresponds to the time at the workstation, which takes the maximum time (Workstation 2 in this example). Thus, the cycle time for this example is 141.6 seconds.

The advantage of this approach over the COMSOAL, a heuristically-based search, is that this approach leads to an optimum solution for the minimum number of workstations.

4.2. Design Candidate Search

The starting point for redesigning candidates begins with the results of the previous step, which identified the optimum number of workstations, given a takt time. For this redesign candidate search, the focus shifts to two fundamental questions:

1. How do we efficiently and effectively identify the precedence constraints that should be explored for elimination (by taking a redesign action)?
2. How do we efficiently and effectively identify assembly tasks that should be split-up (also through redesign action)?

These two questions arise from the fact that by eliminating precedence constraints, and by further sub-dividing assembly tasks, it should make the balanced assignment of tasks to particular workstations easier.

We propose tackling this problem by implementing a procedure based on the solution of a sequence of optimization problems. The first problem determines the minimum number of workstations for a given takt time, and the assignment of tasks to stations. The outputs of this problem will be then considered inputs in a modified optimization problem, where the objective function not only account for processing costs, but also accounts for other metrics of interest for the assembly and design process. This new problem will determine additional task-to-station assignments, given that the number of stations is now limited to minimum previously determined. It is precisely the formulation of the second optimization problem that will constitute the next phase of this research.

Component Properties Related to Precedence

In order to determine which precedence relationships should be examined more closely, it is useful to draw upon the design for variety (DFV) research [22], which notes that the assembly sequence should be a function of the commonality of a particular component, the lead time associated with each component and the value of the variety that is afforded by that component (in other words, is the component differentiation that adds value to the end user). It is not difficult to image that the more common (or standard) a component is, the earlier in the assembly sequence it should be assembled. Similarly, the longer the lead time of a particular component, the earlier the component should be in the assembly sequence. Lastly, if a component violates these principles, the needed design action

should be dictated by the value the end user places on the differentiation. If a customer values differentiation, it would suggest that the component needs to be redesigned to be sequenced later in the assembly process. Conversely, if it is not valued by the customer, the component should be redesigned to become more standard.

Given these observations, it seems logical that metrics which help identify components that need to be re-sequenced can be developed. With these metrics, it should then be possible to integrate them into an objective function that shows the value of manipulating the precedence constraints associated with those components.

Component Properties Related to Component Division

Another approach that will be explored in order to aid in the more efficient and effective balancing of the manufacturing line is to identify the number of functions that are being implemented (or partially implemented) by a component. The next step in the process would entail looking at the value of variety as was done above, but instead of looking at this value by component, it will need to be assessed by function. The benefit of doing this would be that if there is a mismatch (i.e., a function with high variety value and function with low variety value) within the same component, then this becomes a component candidate for being split-up.

As with the previous approach, the idea is to develop metrics that capture these effects and that can be integrated into an optimization problem. With these two effects, it may be possible to identify the components that yield the most benefit from redesign, seeding the redesign process.

Objective Function Formulation and Solution

If the metrics that were discussed above can be successfully captured, their integration into an objective function should be relative straight-forward. However, the resulting modified line-balancing optimization problem may result in non-linear formulations for which linearization may be necessary. .

Industry Case Study and Design Guidance

In addition to the mathematical and computational work noted above, application of these methods to an industry case study will be necessary to demonstrate their utility. The break assembly case study was effective at illustrating the mechanics, but the redesign opportunities identified were not significant. A more realistic case study will entail more components, that will require new methods for a more efficient and meaningful exploration.

5. SUMMARY

In this paper, the need for integrating DFA and manufacturing line performance was motivated. The literature review demonstrated that there was an opportunity for the development of tools to aid practioners with this problem. The results of the first attempt to address this problem were

presented, as well as what was learned from that work. This provided a foundation to motivate and propose the research path for the next phase of this line of work.

ACKNOWLEDGMENTS

The authors would like to thank Mr. Jeffery Robble for his help in programming the C++ algorithm that systematically executed the COMOSOL algorithm.

REFERENCES

- [1] Redford, A., and Chal, J., 1994, Design for Assembly: Principles and Practice, McGraw-Hill Companies.
- [2] Vincent Chan, F. A. S., 2003, Design for Assembly, <http://deed.ryerson.ca/~fil/t/dfmdfa.html>
- [3] Pahl, G., Beitz, W., and Wallace, K., 1996, Engineering Design: A Systematic Approach, Springer Verlag.
- [4] Bralla, J. G., 1986, Handbook of Product Design for Manufacturing, McGraw-Hill Book Company, New York.
- [5] Robert B. Stone, D. A. M., Varghese J. Kayyalethekkel, 2003, "A Product Architecture-Based Conceptual Dfa Technique," Design studies, 25(pp. 301 - 325).
- [6] Yoosufani, Z., Ruddy, M., and Boothroyd, G., 1983, "Effect of Part Symmetry on Manual Assembly Times," Journal of Manufacturing Systems, 2(2), pp. 189-195.
- [7] Boothroyd, D., Knight, 2002, Product Design for Manufacture and Assembly.
- [8] Boothroyd, G., and Dewhurst, P., 1983, "Design for Assembly: Selecting the Right Method."
- [9] Boothroyd Dewhurst, I., 2007, Dfma Cost Reduction Tools, <http://www.dfma.com/software/index.html>
- [10] Ohashi Toshijiro, I. M., Arimoto Shoji, Miyakawa Seii, 2002, "Extended Assemblability Evaluation Method (Aem) : Extended Quantitative Assembly Producibility Evaluation for Assembled Parts and Products," JSME International Journal, 45(2).
- [11] Hinckley, M., Make No Mistake!: An Outcome-Based Approach to Mistake-Proofing, Productivity Press.
- [12] Coma, O., Mascle, C., and Balazinski, M., 2004, "Application of a Fuzzy Decision Support System in a Design for Assembly Methodology," International Journal of Computer Integrated Manufacturing, 17(1), pp. 83-94.
- [13] Chi-Haur Wu, Y. X., Swee Mean Mok, 2007, "Linking Product Design in Cad with Assembly Operations in Cam for Virtual Product Assembly," Assembly Automation, 27(pp. 309–323).
- [14] Gupta, S., and Okudan, G., 2008, "Computer-Aided Generation of Modularised Conceptual Designs with Assembly and Variety Considerations," Journal of Engineering Design, 19(6), pp. 533-551.
- [15] Caputo, A., and Pelagagge, P., 2008, "Effects of Product Design on Assembly Line Performances: A Concurrent Engineering Approach," Industrial Management and Data Systems, 108(6), pp. 726-749.
- [16] Ken swift, G. J., The Designers' Sandpit, <http://www.hull.ac.uk/MAPP/sandpit/index.html>
- [17] Lambert, A. J. D., 2004, "Generation of Assembly Graphs by Systematic Analysis of Assembly Structures," European Journal of Operational Research, pp. 932-951.
- [18] Arcus, A. L., 1966, "Comsol: A Computer Method of Sequencing Operations for Assembly Lines," International Journal of Production Research, 4(pp. 259 - 278).
- [19] Lg Electronics Inc., P. T. E. A. M. S., 2009, Flexible Line Balancing V. 3 - an Exceptional Product for Balancing Assembly Lines, <http://www.protech-ie.com/flb.htm>
- [20] Esterman Jr, M., and Kamath, K., 2010, "Design for Assembly Line Performance: The Link between Dfa Metrics and Assembly Line Performance Metrics."
- [21] Thangavelu, S., and Shetty, C., 1971, "Assembly Line Balancing by Zero-One Integer Programming," AIIE Transactions, 3(1), pp. 61-68.
- [22] Martin, M. V., and Ishii, K., 2002, "Design for Variety: Developing Standardized and Modularized Product Platform Architectures," Research in Engineering Design-Theory Applications and Concurrent Engineering, 13(4), pp. 213-235.