

AC 2007-1539: CONCEPTUAL DESIGN ENVIRONMENT FOR AUTOMATED ASSEMBLY LINE – FRAMEWORK

Sheng-Jen Hsieh, Texas A&M University

Dr. Sheng-Jen (“Tony”) Hsieh is an Associate Professor in the College of Engineering at Texas A&M University. He holds a joint appointment with the Department of Engineering Technology and the Department of Mechanical Engineering. His research interests include engineering education, cognitive task analysis, automation, robotics and control, intelligent manufacturing system design, and micro/nano manufacturing. He is also the Director of the Rockwell Automation laboratory at Texas A&M University, a state-of-the-art facility for education and research in the areas of automation, control, and automated system integration. Dr. Hsieh received his Ph.D. in Industrial Engineering from Texas Tech University, Lubbock, TX.

Conceptual Design Environment for Automated Assembly Line – Framework

Abstract

Automated systems play an important role in our daily life and our national economy. Educating students about how to design automated assembly systems is very important. However, education in this area most often takes place in senior design courses. This may be because of the multi-disciplinary nature of the problems and the high cost of automated equipment. In this paper, we present a prototype web-based system called the Automated Assembly Line Design (AALD) environment that allows users to rapidly create and compare conceptual designs of automated assembly line systems in a systematic way. In addition, the prototype allows instructors to post new problems and to monitor how students design assembly line systems using a mouse-tracking feature.

1. Introduction

Automated systems play a significant role in our daily life and national economy. They are used to manufacture everyday products such as golf balls, cookware, and cell phones. They are also used for applications such as luggage sorting conveyor systems at major airports, control of roller coasters at amusement parks, and automated robotic welding lines in the automobile industry.

Traditionally, engineering courses have covered only selected automated assembly line design concepts, usually in the context of teaching some other topic. For example, in industrial engineering, workstation design, assembly line design and line balancing are covered in courses on production systems. In mechanical engineering, design of devices such as material handling trays are covered in mechanical design courses. There is typically no single course that addresses automated assembly design issues.

Education on the big picture of system integration is typically accomplished via capstone courses or senior design projects¹. In this type of instruction, students are expected to learn by doing, and learning outcomes may vary depending on the type and difficulty of the selected projects. Another approach is to create an interdisciplinary course, such as Industrial Automation², which allows students from different disciplines—such as mechanical and electrical engineering—to enroll. This approach can bridge the gap and create a common language across disciplines. Hsieh³ described a need to better understand how engineers develop expertise in automated system integration and to design high quality educational curricula that will equip students with the necessary skill sets.

In the area of software design, reported efforts include the design of Delmia *IGRIP* software^{4,5}, Adept Digital Workcell Design Software (formerly *SILMA* software)⁶, and FESTO *Cosimir*^{7,8}. The primary focus of these software programs are off-line simulation, cycle time calculation, programming, interference checking; and they are not web based. This paper describes a design framework for an automated assembly line design toolkit. This toolkit, which was developed by the author as part of an NSF CCLI grant, is designed to allow users to systematically design and

evaluate different design alternatives, with an emphasis on design of automated assembly lines. This tool is intended to help bridge the gap between practical needs and current methods of automated system design education.

2. Conceptual Design Environment for Automated Assembly Line

The Automated Assembly Line Design (AALD) environment presents design problems in the following sequence.

a) **Problem stage:** The problem stage presents the design problem. The system can present up to eight different kinds of problems. Each problem has a picture and problem statement. Whenever the user clicks on a problem button, the corresponding problem statement and picture appear on the screen. A scroll bar is available when viewing longer problem statements.

b) **Process stage:** In the process stage, the user identifies symbols corresponding to desired assembly line functions. The user selects and sequentially allocates symbols for an assembly line design and enters processing time for each function. The cost for each function will be loaded on the system.

c) **CPM stage:** In the CPM stage, the user constructs an assembly line network. A matrix represents the possible network columns and rows that describe the relationships among assembly functions. The user makes decisions to establish relationships for better assignment of work content to workstations for completing the assembly line design.

d) **Parts Feeder stage:** In this stage, the user views a precedence diagram reflecting his/her selections. The user also selects parts feeders for appropriate assembly functions.

e) **Layout and Simulation stage:** The user can choose one of two layouts: in-line or u-shaped. The simulation shows analyses of the design performance to allow evaluation of the properties of the conceptual design. The user may review calculated values, such as cycle time, critical path, and total cost.

f) **Show Designs stage:** The user reviews each design in terms of cycle time, total cost, and processing times and costs for each assembly function. The designs can be efficiently analyzed.

The AALD is designed primarily to help undergraduate students to visualize designs, to identify potential problems (such as bottleneck stations), and to document designs systematically. It can also be used by practicing engineers as a tool for rapid creation of conceptual models for demonstration to potential customers. It is not intended for use as a theoretic or analytic tool for addressing issues such as uncertainty in workstation processing times. In the future, after the feasibility of the initial concept has been established, a companion tool that addresses more complex issues may be developed for use in graduate education.

3. Primary Components of Conceptual Design Environment

In this section, we cover primary components of the AALD prototype. Each of these components corresponds to one of the design stages listed in the previous section.

3.1 Process Component

This component allows the user to document the sequence of an assembly process. There are two windows—an *operations window* (on the left) and a *design window* (on the right). In the operations window, the user can use the arrow key to navigate through different types of operations. Each operation has symbols representing various types of equipment that can be used to perform that operation. For example, for the *handling* operation, there are four different types of equipment, ranging from single finger grippers to triple finger grippers. Each type can handle a different number of components simultaneously and therefore has a different cost.

Primary categories of operations include *Feeding, Handling, Composing, Checking, and Adjusting*. Each operation has associated *time* and *cost* elements depending on the complexity of the operation. In general, as equipment cost increases, the corresponding cost element value also increases. The time element allows the user to enter an estimate of the time required for each operation. To provide increased flexibility for instructors, the cost elements for the process are stored in a text file. Cost elements can be easily modified by changing the text file. Figure 1 shows a sample Process screen.

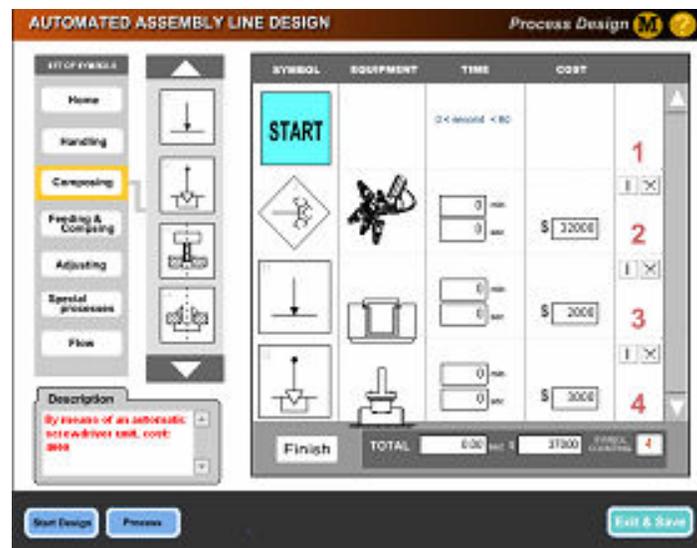


Figure 1. Sample screen from Process component.

3.2 Critical Path Method (CPM) Component

CPM is an algorithm that processes the precedence relationship and time duration among operations with a purpose of identifying the critical path for the entire operation of a project. The critical path is the longest path of the project; therefore, if you can reduce the time required to complete the critical path, you can reduce the total time to complete the entire project. Figure 2 shows an example of a network structure where operations are represented by a node symbol and the duration of each operation is shown on the link. CPM is widely used in industry for project management.

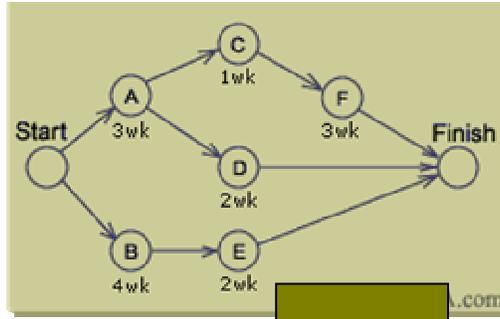


Figure 2. Example of a CPM network structure.

In the CPM component, the user will enter the precedence relationships among operations by clicking on the rectangular box associated with each cell (Figure 3). The component will automatically identify the critical path and calculate the cycle time for the entire assembly operation. This information will be displayed in the bottom left corner of the interface as shown in Figure 3. The times for each operation were entered in the *Process* component.

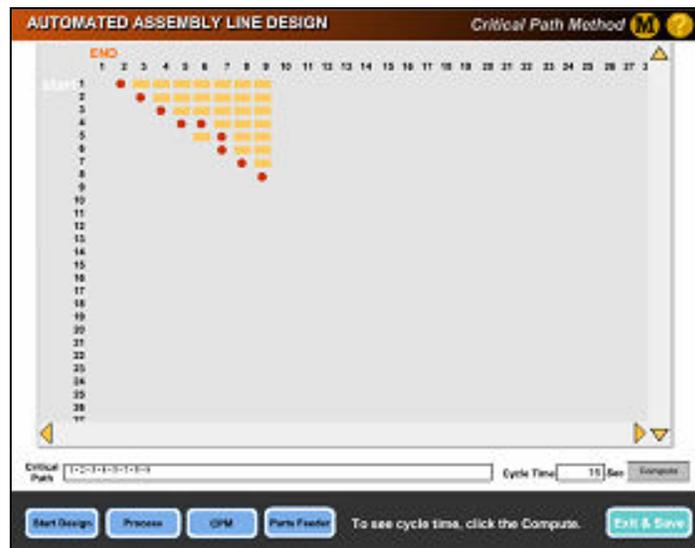


Figure 3. Screenshot of CPM component.

3.3 Parts Feeder Component

The Parts Feeder component allows the user to select a part feeder for each symbol. When the user clicks on the layout button during the CPM stage, a precedence diagram comes up in the middle of the screen. If the user chooses over 20 symbols, the diagram will be longer than the width of the window, so the user can use left-right scroll buttons. Figure 4 is a screenshot of the Parts Feeder component.

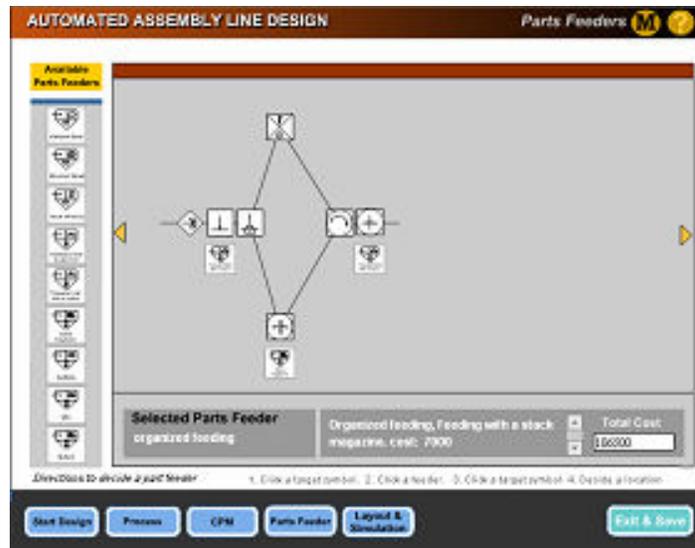


Figure 4. Screenshot of Parts Feeder component.

To read the description for a particular type of part feeder, users can move the mouse cursor over the part feeder. Each part feeder includes a description, which is displayed in a box at the bottom of the screen. The total cost, including part feeders, is displayed in a text box on the bottom right of the screen. When the user clicks on the Next Stage button (which leads to the Simulation process), the design is fixed and cannot be changed. Users can see the designs later during the Show Design stage. Figure 5 shows a series of screens in which a user chooses a part feeder for each operation.

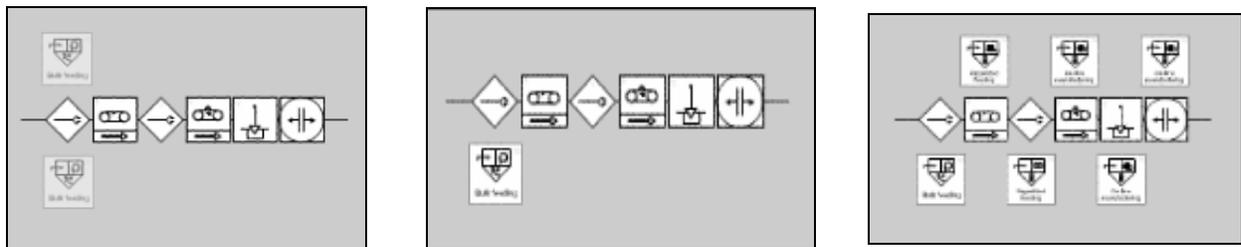


Figure 5. Screenshots showing selection of a part feeder for a handling operation.

3.4 Layout and Simulation Component

This component allows users to (1) select two different layout alternatives for the operations described in previous stages, (2) simulate material flow in each proposed layout, and (3) identify the critical path, cycle time, and overall cost of the operations. Layout alternatives include *In-line* and *U-shape*. Figures 6 and 7 show the two layout alternatives.

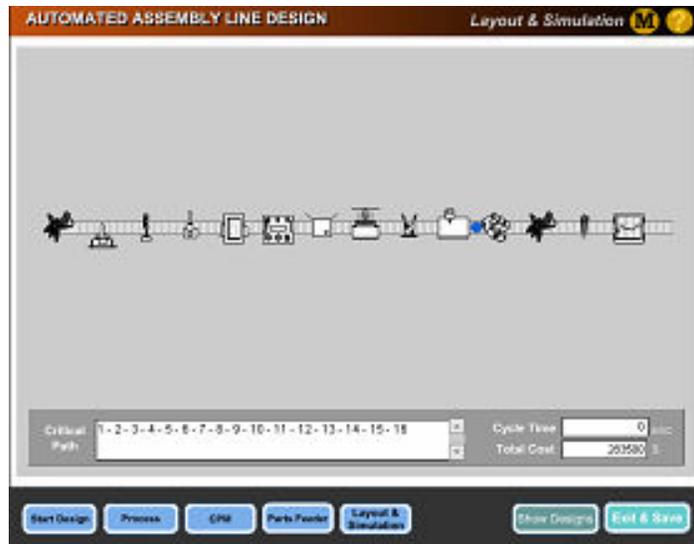


Figure 6. In-line layout.

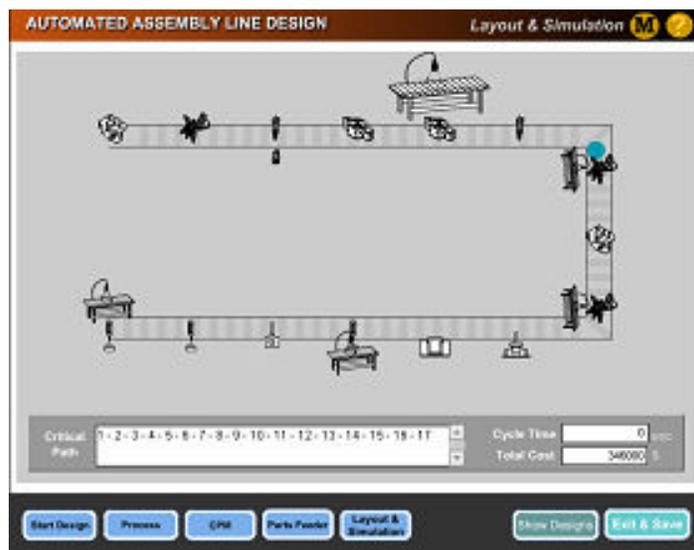


Figure 7. U-shaped layout

As shown in the above figures, the Layout and Simulation component allows users to visualize and evaluate a design to see if the proposed design is closed to the desired one. If a design is not appropriate, changes can be made by going back to main menu to start a new design.

3.5 Show Designs Component

The Show Designs component allows users to review their designs and the corresponding analytic results of each design in a separate window. The analytic results include cycle time and total cost of each design. Users may have up to six different design alternatives; a simplified screen of each design is shown in a separate window.. To view a design, the user simply clicks on the corresponding window. Due to the window size limitation, users can only view two designs at one time. To compare multiple design alternatives, user can close one of the designs (by clicking on the circle with X in it on the bottom left corner of the window) and open another

design by clicking on one of the six numbered windows. Figure 8 shows screenshots from the *Show Design* component.

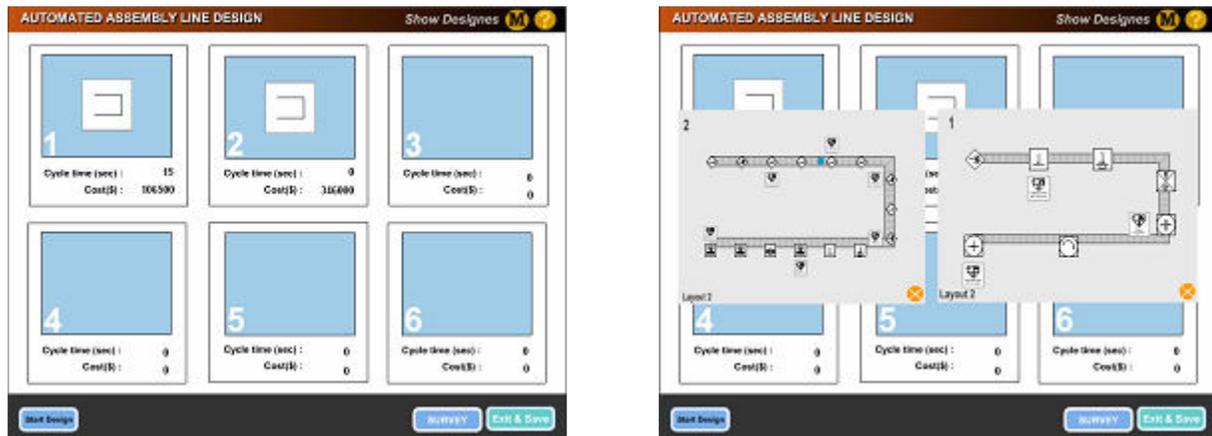


Figure 8. Screenshots from Show Designs component.

4. Conclusions and Future Directions

This paper described the need for a system to teach automated assembly line design and the contents of a prototype Automated Assembly Line Design (AALD) environment. The AALD environment allows users to experiment with the design of automated assembly line systems in a systematic way. In addition, the mouse tracking log files can be used to investigate how much time users spend on each event, page, and design step, as well as the sequence of user actions. Thus, if a user spends a lot of time on one page, this may suggest that he or she is thinking or does not know what to do next. If a page is not visited by most users, it may be because the users did not notice the controls to access the page. If a user moves back and forth between pages quickly, it may be because he or she wants to know the effect of changing a parameter. This type of analysis can help to better understand how learners use the prototype and assist in improving the design of the user interface.

Future directions include (1) addition of an interactive learning tutorial that includes case studies about automated system design; (2) extensive user testing of the prototype, (3) soliciting industry experts' input about the prototype; (4) comparing and contrasting how experts and novices solve similar problems; and (5) making the prototype available on the web.

Acknowledgments

This material was supported by National Science Foundation grant number 0238269. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Bibliography

1. Schuyler, P.R., Implementing a complete control curriculum in the classroom, in *Proceedings of the Frontiers in Education Conference, 1997. 27th Annual Conference. Teaching and Learning in an Era of Change (Vol. 2)*, pp. 604 -609.

2. Cambron, M.E. and Lenoir, H.J., "Introduction to Industrial Automation, a multi-disciplinary course at Western Kentucky University," *ASEE Annual Conference Proceedings, 2004*, pp. 8363-8370.
3. Hsieh, S. "Automated Manufacturing System Integration Education: Current Status and Future Directions," *Proceedings of 2005 ASEE Annual Conference, June 12-15, 2005, Portland, OR*.
4. <http://www.delmia.com>.
5. Cheng, F.S. "A Methodology For Developing Robotic Workcell Simulation Models," *Proceedings of the 2000 Winter Simulation Conference*, J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, eds.
6. <http://www.adept.com>.
7. COSIMIR (2005). *COSIMIR® Industrial – Integrated Simulation and Programming Software*. Accessed online (January, 2005). <http://www.axicont.com/Cosimir.htm>
8. Harris, S. (2005). COSIMIR Software. Rixan Associates, Inc. [Phone Conversation]. January, 2005.