Assembly Sequence Analysis

• Goals of this class
  – Understand one algorithmic approach to finding all feasible assembly sequences
  – Make connection between algorithm and assembly feature models
  – See how assembly sequences can be designed
  – Look at some examples
  – See a video of computer-aided assembly analysis
History

- Assembly sequence analysis applied to line balancing (Prenting and Battaglin, 1964)
- Heuristics such as “the fastener method” (1978)
- Bourjault method (1984)
- De Fazio/Whitney method (1987)
- Gustavson exploded view method SPM (1989)
- Baldwin onion skin method (1989)
- Sukhan Lee method (force paths, subassemblies, 1989+)
- Wilson method (free directions, 1992+)
Role of Sequence Analysis in Concurrent Engineering

• Line balancing applies sequence analysis after the product is designed
• Our goal is to push assembly sequence analysis to the beginning of the development process
• It can be an important lever in concept design
• It interacts with architecture and affects supply chain, build to order processes, JIT, etc.
• To keep up with designers during fluid concept design, the assembly engineers need a tool that gives fast turnaround
Analysis Alternatives

• Find all feasible sequences
• Find all linear feasible sequences
  – add one part at a time
• Find one feasible sequence
• Find one linear feasible sequence
• The first one is of the most interest to assembly line designers
Process Phases

• Eliminate all truly impossible sequences
  – parts physically block other parts
  – sequences dead end before completion

• What remain are called “feasible” = not impossible
  – the good, the bad, and the ugly

• By various criteria, throw out bad and ugly
  – Criteria include technical and business issues

• This is traditional design:
  – generate requirements
  – generate alternatives
  – use requirements to narrow the alternatives
Classes of Approaches

• Most methods assume “one hand”
  – Forbids joining 3 things at one step
• Graph theory analysis of liaison diagram
• Systematic textual analysis of lists of liaisons that contain blockers
• Cut-set methods applied to the liaison diagram
• “Onion-skin” methods that peel off outside parts
• Most of these methods utilize disassembly as the paradigm but it is not necessary
  – “can you remove part X from parts Y,Z,...?” is the same as “can you put part X onto parts Y, Z,...?” under most circumstances
Non-Assembly Steps Can be Included

- Reorientation
- Tests, lubrication
- Temporary disassembly
- These can all be handled one way or another if you are creative
The Liaison Diagram

- A simple graph that denotes parts as nodes and connections as arcs
- Can be augmented with information about the connection
Rules of Liaison Diagrams

• Each part is a node, each arc is a liaison
• Each part has no more than one liaison with any other part
• In a loop of n liaisons, if n-2 arcs are closed, then attempting to close 1 of the 2 remaining will automatically close the whole loop:
Generating Sequences
Selecting Sequences

Image removed for copyright reasons.
Source:

Image removed for copyright reasons.
Source:
Two Alternator Sequences

Source:
Two More Alternator Sequences
Rules of Sequence Analysis

- Parts are rigid
- Liaisons (connections between parts) are also “rigid”
- Once a liaison is made, it stays made
Ask and Address Precedence Questions

- Goal of questions is to find out what moves are forbidden
- This is done various ways by different methods:
  - Computer searches for free paths, using local escape directions and checking for interference
  - Person detects these
- Typical questions:
  - can this part be added to those parts
  - can this set of parts be added to that set of parts
  - must these parts be present/absent in order to add that/those parts
Subset and Superset Rules Cut the Number of Required Questions

• These are true only if parts and liaisons are “rigid”

• Subset rule:
  – if you can add part X to parts \{Y\} then you can add part X to any subset of \{Y\}
  – fewer parts can’t contain blockers that aren’t in the original set

• Superset rule:
  – if you can’t add part X to parts \{Y\} then you can’t add part X to a superset of \{Y\}
  – adding parts can’t remove blockers that are in \{Y\}
  – counter-example in Sony tape deck with motor
Local and Global Freedom

• Local freedom means that the combined escape directions of all liaisons in the query have a common direction (dot product of escape vectors $= 1$)

• Global freedom means that there is a long range escape path that completely separates the parts in the query

• Local freedom can be detected by the computer just by inspecting the escape directions - easy

• Global freedom requires solving the “piano mover’s problem” - difficult or impossible
Finding Local Freedoms

• Use escape direction vectors:
  – Look at escape direction vectors for each feature
  – Look for common vector for them all

• Conventional screw theory will not work
  – It’s too hard to distinguish one-sided motion limits

• “Dr. Whitney, what do you do about the facets?”
Generate Precedence Relations

• Example of cookie jar and cookies
  – Can you put the cookies in the jar if the lid is on?
  – “No.”
  – Therefore: cookies to jar > lid to jar
Diagram Feasible Sequences

- Network of sequences represents
  - States of assembly = feasible subassemblies showing which liaisons have been completed
  - Transitions between states
- A path through the network is a feasible sequence
  - Cookies to jar, then lid to jar
  - Cookies to lid upside down, then jar to lid, then flip
- We decide later which is better
Simple Assembly Sequence Example

Assembly

Liaison Diagram

Local escape directions shown by arrows
Bourjault’s Process as Textual Analysis

- Analysis question:
- \(R(a; b, c, d)=\text{Can you make liaison } a \text{ when } b, c \text{ and } d \text{ are already made?}\)
- Ask this question for every liaison \(a\) combined with every other liaison \(b, c, d\)
- Bourjault’s original process uses graphical analysis based on circuit theory
R(1;2,3,4) Can’t answer because 2,3,4 forces 1
Eliminate 2, 3, or 4
Eliminate 2: R(1;3,4) = No (need to know why)
    Eliminate 3: R(1;4) = Yes (so 4 is not why)
    Eliminate 4: R(1;3) = Yes (so 3 is not why)
    So 1 >= 3,4 (i.e., 3,4 together is why)
Eliminate 3: R(1;2,4) = No
    Eliminate 2: R(1;4) already answered Y
    Eliminate 4: R(1;2) = Y
    So 1 >= 2,4
Eliminate 4: R(1;2,3) = No
    Eliminate 2: R(1;3) already answered Y
    Eliminate 3: R(1;2) already answered Y
    So 1 >= 2,3

R(2;1,3,4) Can’t answer
Eliminate 1: R(2;3,4) = No
    Eliminate 3: R(2;4) = Yes
    Eliminate 4: R(2;3) = Yes
    So 2 >= 3,4
Eliminate 3: R(2;1,4) = Yes
Eliminate 4: R(2;1,3) = No
    Eliminate 1: R(2;3) = aaY
    Eliminate 3: R(2;1) = Yes
    So 2 >= 1,3
Done by symmetry:
4 >= 1,2
4 >= 1,3
4 >= 2,4
4 >= 2,3
Results of Bourjault Method

these two together can’t be first

1,4 do not appear on the RHS, so they are unprecedented, so they can be first, in either order

Then 2 and/or 3 can be next

\[
\begin{align*}
1,2 &\geq 3,4 \\
1,3 &\geq 2,4 \\
1,4 &\geq 2,3 \\
2,4 &\geq 1,3 \\
3,4 &\geq 1,2
\end{align*}
\]
Simple Assembly Sequence Results

Assembly

Liaison Diagram

PRECEDEENCE RULE:  1 & 4 > 2 & 3
Portion of Cutset Method

All questions except the last are answered by inspecting local freedom

R(1,2;3,4)? No: 1,2 >=3,4
R(1,4;2,3)? No: 1,4 >=2,3
R(3,4;1,2)? No: 3,4 >=1,2
R(2,3;1,4)? Yes: 1,4 unprecedented so they can be first.
Other Methods

• Randall Wilson checks global freedom using the “weighted blocking graph”
• This is essentially a search for unidirectional escape paths along any of the local escape directions
• The escape directions are generated by inspecting individual surfaces on adjacent parts that touch each other, essentially rediscovering the mating features
Other Methods, cont’d

- Gustavson and Wolter each generate exploded views by different methods and then generate precedence relations from the order along major explosion directions.
- A reasonable assumption is that unless there is some blockage, all moves along one such direction will be done before starting on another.
- Gustavson finds a heuristic sequence along major explosion directions using part c.g.s and asks the user to fix any errors.
SPAS and Onion Skin Methods

• All local freedoms are checked by the computer
• All global freedoms are queried to the user
• This has two benefits
  – the computer does the easy part without attempting the impossible part or pretending to do it
  – the user must confront the design and become very familiar with it
  – Ref: Russ Whipple SM Thesis, MIT, June 1990
Stability Checking (Simplified)

- Start with the base part or the part in the fixture
- By definition, it is stable
- Check each of its liaisons
  - compare local escape direction to gravity
  - if part can’t slide out then mark it “stable”
- Check the liaisons of each of the newly defined stable parts the same way
- If all parts in the liaison diagram can be marked “stable” then the assembly is stable
- Screw theory can be used to find mobility
Real Assembly Sequence Analysis Example

Images removed for copyright reasons.
Source:

## ASSEMBLY DATA MODEL

### PARTS OF REAR AXLE

<table>
<thead>
<tr>
<th>PARTS</th>
<th>LIAISONS</th>
<th>PRECEDENCE RELATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A = CARRIER ASSY</td>
<td>1 = C TO A</td>
<td>2 &gt; 1</td>
</tr>
<tr>
<td>B = BACKING PLATE</td>
<td>2 = B TO A</td>
<td>5 &gt; 4</td>
</tr>
<tr>
<td>C = SHAFT</td>
<td>3 = J TO B</td>
<td>1 &amp; 2 &amp; 6 &gt; 5</td>
</tr>
<tr>
<td>D = BRAKE DRUM AND T'NUT</td>
<td>4 = D TO C</td>
<td>5 &gt; 7</td>
</tr>
<tr>
<td>E = WITHDRAWN PINION SHAFT &amp; BOLT</td>
<td>5 = G TO C</td>
<td>11 &gt; 8</td>
</tr>
<tr>
<td>F = INSERTED &quot; &quot; &quot;</td>
<td>6 = E TO A</td>
<td>10 &gt; 9</td>
</tr>
<tr>
<td>G = (PUSH IN SHAFT &amp; ) C-WASHER &amp;</td>
<td>7 = F TO A</td>
<td>12 &gt; 10</td>
</tr>
<tr>
<td>PUSH SHAFT OUT</td>
<td>8 = L TO A</td>
<td>12 &gt; 11</td>
</tr>
<tr>
<td>H = OIL</td>
<td>9 = I TO A</td>
<td>3 &gt; 1 &amp; 4 &amp; 5</td>
</tr>
<tr>
<td>I = COVER</td>
<td>10 = H TO A</td>
<td>7 &gt; 10</td>
</tr>
<tr>
<td>J = BRAKE CABLE, COILED</td>
<td>11 = K TO A</td>
<td>9 &gt; 11</td>
</tr>
<tr>
<td>K = FINAL PRESS TEST</td>
<td>12 = M TO A</td>
<td></td>
</tr>
<tr>
<td>L = AIR TEST PLUG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M = FIRST PRESS TEST</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rear Axle

Image removed for copyright reasons.
Source:
Rear Axle Differential Subassembly
After Liaison 1 Shaft to Carrier

Image removed for copyright reasons.
Source:
Pinion Shaft Out (Liaison E) and
Insert C Washer (Liaison G)

Source:
Image removed for copyright reasons.
Source:
Example assembly sequence graph

Each path from top to bottom is a valid sequence. Each box is a valid intermedia assembly state.

Source:
Six Speed Truck Transmission

Drawn by T. L. De Fazio

assy seq anal 10/6/2004 © Daniel E Whitney
Juicer

Image removed for copyright reasons.
Source:
Network Complexity Metric

- The liaison diagram is a network
- How complex is an assembly?
- Network complexity metric $k$: $(\#\text{arcs}) / (\#\text{nodes})$
  - Node = part, arc = connection between 2 parts
- If $n = \# \text{parts}$, then
  - Min $k = (n-1)/n$
  - Max $k = n(n-1)/2n = (n-1)/2$
- Which product will have more assembly sequences?
  - Product with big complexity metric
  - Product with small complexity metric
Chinese Puzzle
Some Data on Liaisons per Part

Liaisons per Part

Image removed for copyright reasons.

Source:
More Data: 34 Products

![Graph showing liaisons per part versus number of parts for different products.]

- Chinese Puzzle: 6 liaisons/part
- Rugged Stapler: 3 liaisons/part
- Paper Shredder: 2 liaisons/part
- V-8 Engine: 1 liaison/part
- Many Consumer Products: lower liaisons/part range
V-8 Engine Liaison Diagram
Constraint Limits Liaisons/Part

\[ M = 3(n - g - 1) + \sum \text{joint freedoms } f_i \]

where

\( n = \) number of parts
\( g = \) number of joints
\( f_i = \) degrees of freedom of joint \( i \)

\[ \alpha = \text{liaisons / part} \]
\[ \downarrow = \text{average dof per joint} \]
\( g = \alpha n \)
\[ \sum f_i = g \downarrow = \alpha \downarrow n \]

and

\[ M = 3(n - \alpha n - 1) + \alpha \downarrow n \]

If \( M = 0 \)

\[ \alpha = \frac{3 - 3n}{n(\downarrow - 3)} \cdot \frac{3}{3 - \downarrow} \text{ as } n \text{ gets large} \]

\[ \begin{array}{|c|c|c|}
\hline
\downarrow & \alpha \text{ planar} & \alpha \text{ spatial} \\
\hline
0 & 1 & 1 \\
1 & 1.5 & 1.2 \\
2 & 3 & 1.5 \\
\hline
\end{array} \]
Number of Liaisons vs Number of Sequences

$y = 276.31x - 1906.3$

$R^2 = 0.9139$
Video

• Made by Randy Wilson and co-workers at Sandia National Labs in 1996
• Obtains one feasible sequence by using feasible escape cones derived from local escape directions
• Permits user to edit this sequence