Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

SUPERVISORY CONTROL OF AN AUTOMATED DISASSEMBLY WORKCELL BASED ON BLOCKING TOPOLOGY

Kok-Meng Lee
Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
Atlanta, GA 30332-0405

Michael Martin Bailey-Van Kuren
University of Texas at El Paso
Mechanical and Industrial Engineering
El Paso, TX 79968

Abstract
An important aspect in recycling and reuse for environmentally conscious manufacturing is disassembly. Although disassembly systems contain the same basic elements as assembly system, the disassembly problem differs significantly from the assembly problem due to the fact that the incoming disassembly product is not controlled. In addition, complete dismantling of products is not necessarily required for disassembly whereas assembly builds a product completely. This paper describes a model for automated disassembly that accounts for work cell interaction and used product constraints. The model provides an essential means to determine, in real- time, the next component for disassembly using the knowledge of the product design and sensor feedback to minimize the steps to removing goal components. Sets of components for removal were resolved by minimizing set-up time for disassembling the component. Simulation results based on real product, vision sensor measure, and process input are presented and discussed. It is expected that the concepts demonstrated through this product can provide useful insights to other mechanical assembly.

1. INTRODUCTION

An increasing awareness of the effects of technological advances on the environment has spurred research into "environmentally conscious" or "green" engineering [1]. Under pressure by the European governments, European car manufacturers have invested large efforts in studying the recyclability of their products [2]. A trend can be seen towards increased product take back and subsequent automation, with emphasis on automated disassembly and separation for product recycling and reuse [3]. An important aspect of recycling, reuse and disposal of consumer products is the disassembly of the product and its components [4], which is essential for acquisition of desirable or undesirable material from a product, or segregation of dissimilar materials.

Although disassembly systems contain the same basic elements as assembly systems, the disassembly problem differs significantly from the assembly problem in three aspects: First the condition of incoming disassembly product is not controlled. Secondly, the extent of the disassembly for a particular product may vary with material needs creating different goals for the cell for the same product. Third, the complete dismantling of products is not necessarily required for disassembly whereas assembly builds a product completely.

Weigi [5] performed some experimental disassembly of video camera recorders. Their strategy focuses on the removal of fasteners and jammed components and does not adjust for missing components. Dario et. al [6] present a theoretical framework for disassembly that emphasizes sensor information processing and fusion. Spath [7] outlines an information system for disassembly. Schmult [8] created a complete system for disassembly of block structures which simplifies the domain of possible motions to disconnect parts. Little work, however, has been done on modeling and control of an automated disassembly system which takes into account of the product configuration variations or uncertainties, particularly in the context of real-time supervisory control. For this reason, we develop a method to model and control an automated disassembly workcell, which provides an essential basis for performance evaluation of different strategies of disassembly.

Specifically, this article provides the following: (1) an automated disassembly system model for describing workcell and used product states collectively; (2) a supervisory control algorithm for reduces processing time while accounting for variations in the product configurations; (3) an automated disassembly simulation that incorporates real product and workcell parameters; and (4) an analysis of the results which provide insights into the system behaviors and limitations.

The remainder of this article is organized as follows: Sections 2 and 3 describe the modeling of the used disassembly product states and automated workcell model based on the concept of blocking topology. Section 4 outlines the sensor measure for detecting jammed and missing components. Section 5 discusses the results. Conclusions are given in Section 6.

2. USED DISASSEMBLY PRODUCT MODEL

The *used disassembly product* (UDP) is defined as any product containing components connected by reversible mechanical connections. A mechanical connection is reversible if the component can be removed by reversing the motion that assembled it. Thus, reversible connections do not include welds, rivets, or certain snap-fits. However, the UDP may contain missing, replacement, additional or jammed components. As with most assemblies, the order of component removal is essential since some components block the removal of others.

The UDP model provides a mathematical representation which determines the next component to be disassembled as a function of the current used product state $u(\ell)$. In other words,

$$c_i(\ell+1) = h\big(u(\ell)\big) \tag{1}$$

where $u(\ell)$ defines the components or sub-assemblies that are attached or detached from the base assembly at the time instant ℓ^{th}. This is a discrete point set represented by u = (A,D) where A and D the attached and detached component sets respectively and are in the form $\{c_1, c_2, .., c_i, \ldots, c_n\}$.

2.1 Component Blocking Topological Space

The disassembly process is a progression through the component states. Only some states are reachable from the present component state. For each component, a neighborhood of possible components for removal can be defined using a blocking topology and a metric can then be applied to the component state space to measure the distance to goal components.

In specifying a topology, the ability to handle unknown states as well as deal with local information instead of the global configuration of the product were considered. By denoting the relationship where c_1 precedes c_2 by $c_1 \Rightarrow c_2$, we define the k^{th} neighborhood in the component state space, N_k, as follows:

$$c_i \in N_k \text{ if } c_k \Rightarrow c_i \text{ or } i = k$$

where the subscript (•) of c refers the i^{th} or k^{th} component. The null component in the component space, when no components are detached from the used product, is c_0.

<u>*Lemma:*</u> *The component state space with neighborhoods or sets of components defined by a component and all components blocked from disassembly by the component is a Hausdorf topological space.*

I. For each c_k in C, there exists a neighborhood N_k that contains c_k. Proof: k=k. Therefore, by the definition of a component neighborhood, $c_k \in N_k$ for all c_i.

II. The intersection of two neighborhoods of c_k contains a neighborhood of c_k. Proof: Let $c_2 \in N_1$ and $c_2 \in N_3$ or two neighborhoods that contain c_2. Furthermore, $c_2 \in N_2$ by definition. Since $c_2 \in N_1$ and $2 \neq 1$, then $c_1 \Rightarrow c_2$. Furthermore, for all $c_i \in N_2$ not including c_2, $c_2 \Rightarrow c_i$. If $c_1 \Rightarrow c_2$ and $c_2 \Rightarrow c_i$, then $c_1 \Rightarrow c_i$. Thus, $c_1 \Rightarrow c_i \ \forall \ c_i \in N_2$, therefore $N_2 \subseteq N_1$. Since, c_2 does not precede c_1, then $c_1 \in N_1$ and $c_1 \notin N_2$. Thus, $N_2 \subset N_1$. In a similar manner, $N_2 \subset N_3$. $N_2 \subseteq (N_1 \cap N_3)$. Thus, $(N_1 \cap N_3)$ contain N_2, a neighborhood of c_2.

III. If c_i is a point in N_k there exists a N_i such that $N_i \subseteq N_k$. Proof: As shown in part II, if $i \neq k$, $N_i \subset N_k$. Otherwise, i=k and $N_i = N_k$.

IV. If c_i does not equal c_k, there exist N_i and N_k such that $N_i \bullet N_k = 0$. Proof: First, $N_i \bullet N_k$ is defined as the dot product operator for two neighborhoods. For point sets, $N_i \bullet N_k = 1$ if $N_i = N_k$ and 0 otherwise. If $c_i \Rightarrow c_k$ then $c_i \notin N_k$ and $c_i \in N_i$; $N_i \bullet N_k = 0$. Likewise, if $c_k \Rightarrow c_i$ then $c_k \notin N_i$ and $c_k \in N_k$; $N_i \bullet N_k = 0$. If neither component precedes the other, then $N_i \cap N_k = \varnothing$ and $N_i \bullet N_k = 0$.

Therefore, the component space under the blocking topology is Hausdorf and a metric can be applied. Once a component, c_k, is identified on the product assembly, the product state is in the neighborhood N_k. The neighborhood identifies components that can next be removed due to the direct precedence relationship. Thus, the product can transition to any neighborhood N_i where $c_k \Rightarrow c_i$. By maintaining a history of removed components, the set of possible components for removal is equal to the union of the neighborhoods of removed components adjoin the set of removed components.

2.2 Component Search Function

The component search function utilizes a component blocking topology to designate the next component to be disassembled by minimizing the number of disconnections to reach the goal component in the neighborhood, $N_{c(\ell)}$. Consider a vector defined as

$$v^q = \begin{pmatrix} v_1^q & \cdots & v_n^q \end{pmatrix}^T \tag{2}$$

where v_i^q is the number of goal components blocked by the i^{th} component after q disconnections and the vector becomes irrelevant when

$$\left| v^q \right|_1 = 0. \tag{3}$$

Equation (3) corresponds to a vector where no goal components are blocked and the minimum value of q satisfying Equation (3) is the minimum number of disconnections to remove all goal components. The vector v^q is referred here as the goal blocking vector which can be calculated as follows:

$$v^{q+1} = [B] v^q \quad \text{where} \quad v^0 = \begin{pmatrix} c_{1,goal} & \cdots & c_{n,goal} \end{pmatrix} \tag{4}$$

$c_{i,goal}$ is zero if c_i is a goal component and 1 if not. $[B]$ is the component blocking matrix defined as

$$[B]=\begin{bmatrix} b_{11} & b_{12} & . \\ . & \dots & . \\ . & \dots & b_{nn} \end{bmatrix}, \qquad b_{ki}=\begin{cases}1 & \text{if k blocks i,}\\ 0 & \text{otherwise.}\end{cases} \tag{5}$$

With the definition of the goal blocked vector, the component search function can be defined as

$$c(\ell+1)=c_a\left| v_i^q = \underset{\forall c_i \in N_{c(\ell)}}{Min}[q]\right. \tag{6}$$

The component c_a is returned that has a lowest number of disconnections to leave the goal components unblocked. If there is a unique component with the minimum value of q, then no other searching is required and the returned component is the next component for removal.

3. AUTOMATED WORKCELL MODEL

The *automated workcell* consists of all manipulators, sensors, tooling and fixtures required for the disassembly process and a controller for these resources. The workcell model determines the next UDP state for a given component removal task subjected to constraints imposed by the resource blocking, the prevention of activating a particular resource due to other resources being active. By defining the inter-relationship among the resources, it prevents blocked resources from activating and processes unblocked resource tasks. The function intrinsically accounts for various resource types and tasks. Mathematically, the workcell model is stated

$$u(\ell+1)=g\left(u(\ell),w_{ci},c_i\right) \tag{7}$$

where w_{ci} is the unique work cell state required to remove component c_i.

3.1 Description of Work Cell State

In disassembly, the key dynamic resources are the tools and sensors and their motions result from their rigid attachment to manipulators or robots. These motions defined with respect to a pre-specified workcell reference frame provides the basis for a resource's description.

The automated work cell state is determined by two vectors in product space, $w=(\vec{x},\vec{\varphi})$ where $\vec{x}$ is the resource position vector; and $\vec{\varphi}$ the resource motion vector. All resources are considered as rigid bodies in three dimensional space represented by a dual vector notation. For the i^{th} resource position, $x_i=\left(\vec{\theta}_i,\vec{p}_i\right)$ where $\vec{\theta}_i=\left(\theta_1\ \theta_2\ \theta_3\right)^T$ and $\vec{p}_i=\left(p_1\ p_2\ p_3\right)^T$ are the orientation and position vectors of the i^{th} resource respectively. Similarly, the resource motion is $\varphi_i=\left(\vec{\omega}_i,\vec{\upsilon}_i\right)$ where $\vec{\omega}_i=\left(\omega_1\ \omega_2\ \omega_3\right)^T$ and $\vec{\upsilon}_i=\left(\upsilon_1\ \upsilon_2\ \upsilon_3\right)^T$ are vectors describing the rotational (in Euler's angles) and translational motions respectively.

To determine the resource position r_i in real-time, the position dual vector is transformed by mapping the motion dual vector to a 3x3 matrix and then multiplying the position dual vector by the matrix. The mapping for the motion dual vector is given by

$$\varphi(r_i)\rightarrow\Gamma(r_i)=\left(e^{[\alpha]},v\times e^{[\alpha]}\right) \tag{8}$$

where $$[\alpha]=\begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$$

The resulting equation to update a resource position by a given motion is

$$\Gamma(r_i)x(r_i)=\left(e^{[\alpha]}\vec{\theta},\ e^{[\alpha]}\vec{d}+\vec{v}\times e^{[\alpha]}\vec{\theta}\right) \tag{9}$$

For a static resource, the position of which remains constant, the motion dual vector is defined as 1 (active) or 0 (inactive). The data structure so-formulated provides a compact description of the automated work cell that is computationally efficient. The automated workcell state vectors can be referenced to locate resources or determine active resources. Furthermore, the structure accounts for static and dynamic resources, inherently.

3.2 Resource Blocking

The resource blocking for a workcell with m resources is represented by a mxm matrix $[R]$ as follows:

$$[R]=\begin{bmatrix} r_{11} & r_{12} & . \\ . & \dots & . \\ . & \dots & r_{mm} \end{bmatrix} \text{where } r_{kj}=\begin{cases}1 & \text{if k blocks j,}\\ 0 & \text{otherwise.}\end{cases} \tag{10}$$

Note that the j^{th} column of $[R]$ provide the resource blocking information for the j^{th} resource. Thus, resource blocking can be obtained from the dot product of the work cell state motion vector and the column vector from the resource blocking matrix. A non-zero dot product indicates that the j^{th} resource should not be activated since one or more resources may interface or block the j^{th} resource. Thus we have,

$$w(\ell+1)=\begin{cases}\begin{pmatrix} x_1,\dots,x_{j-1},\Gamma(r_j)x(r_j),x_j,..x_m \\ \varphi_1,\dots,\varphi_{j-1},\varphi_j=\varphi,\varphi_{j+1},\dots,\varphi_m\end{pmatrix} & \\ & if\left|\vec{\varphi}\bullet col_j[R]\right|=0 \\ w(\ell) & if\left|\vec{\varphi}\bullet col_j[R]\right|\neq \end{cases} \tag{11}$$

The disconnection motion triggers a change in the used product state only if the workcell state matches the required workcell state for the component that is specified for disconnection:

$$u(\ell+1)=\begin{cases}\left(\{c_1,\dots,c_{i-1},c_{i+1},..,c_n\},\{c_i,\dots,c_m\}\right) & \\ & \text{if } w(\ell+1)=w_{ci} \\ u(\ell) & \text{if } w(\ell+1)\neq w_{ci}\end{cases} \tag{12}$$

3.3 Minimization Of Non-Value Added Operations

Often it is possible to find multiple components satisfying Equation (6) which minimizes the number of disconnections. In these cases, a secondary search function that attempts to avoid non-value added operations such as tool changes and product reorientations, and deadlocks due to component jamming and replacement components can be developed.

Consider a set of possible components for removal, as determined from Equation (6), as a set F. The cost function J can be defined for each of the components in the subset F of the local neighborhood of the current component for removal, $c(\ell)$:

$$J(c(\ell),c_i)=\sum_{m=1}^{4}a_m(c(\ell),c_i) \tag{13}$$

where $a_1(c(\ell),c_j)=\begin{cases}0 & if\, c_{j,tool}=c(\ell)_{tool}, \\ 1 & otherwise.\end{cases}$

(14a)

$$a_2(c(\ell),c_j)=\frac{\left|c(\ell)_{orient}\bullet c_{j,orient}\right|}{\pi} \tag{14b}$$

$$a_3(c(\ell),c_j)=c_{j,jam}=P(jammed|c_j) \tag{14c}$$

$$a_4(c(\ell),c_j)=c_{j,replace}=P(replacement|c_j) \tag{14d}$$

and where

$c_{i,tool}$ = j, if resource r_j is the tool utilized for the component removal;

$c_{i,orient}$ = a unit vector specifying the fixture orientation required to remove the component.

$c_{i,jam}$ = a value from 0 to 1, the probability that the component is jammed; and

$c_{i,replace}$ = a value from 0 to 1, the probability that the component is a replacement.

Note that the cost function J for the i^{th} components is defined relative to $c(\ell)$. By iteratively comparing the cost function of the i^{th} components with that of the current component for removal $c(\ell)$, the next component is selected from all components in the neighborhood of the current removal component, $N_{c(k)}$:

$$c(\ell+1)=c_i\left|J(c(\ell),c_i)=\underset{\forall c_j\in F}{Min}\left[J(c(\ell),c_j)\right]\right. \tag{15}$$

4. SENSOR MEASURE

The UDP state is observed by sensing the component features. Variations in sensor measures are assumed to be normally distributed. The feature data are represented by $N_{ri}(s_{ri},\mu_{ri},\sigma_{ri})$ and $N_{mi}(s_{mi},\mu_{mi},\sigma_{mi})$ which characterize the features when the component can be recognized or is missing respectively. The elements $s_{(\bullet)}$, $\mu_{(\bullet)}$ and $\sigma_{(\bullet)}$ are the numerical value and its corresponding mean value and standard deviation characterizing the component features. Using the features stored in the design data base, the sensed features are tested to determine if the component is attached, missing, or indeterminable by means of a component feature comparator. The component feature comparator output comprises of one of three results: the component is present, the component is missing, or a replacement or damaged component is present. This information is interpreted in context of the current system states and are defined by the following equations:

$$c_i=\begin{cases}c_a & \text{if component is present} \\ c_d & \text{if component is missing} \\ \varnothing & otherwise\end{cases} \tag{16a}$$

and
$$u(\ell+1)=u(\ell)+(c_a,c_d). \tag{16b}$$

If a damaged or replacement condition is encountered, the used product state remains unchanged. The ability of the disassembly system to account for these error conditions is handled within the controller.

5. Disassembly Simulation

In order to study the effects of the disassembly parameters quantitatively, simulation models for a number of integrated disassembly process scenarios are developed for a single-use camera as shown in Fig. 1. The camera provides a good example from a material and connectivity perspective.

The following analysis on the single-use camera represents three strategies: The first strategy, <u>targeted disassembly</u> (TD), takes maximum advantage of the reusable components (base, viewbox, subassembly, and cantilever). Thus, only non-reusable components are specified as goal components and the removal operations are non-destructive. The second and third strategies are two different alternatives for recycling where metals segregated from plastics. In the second strategy, <u>completely disassembly</u> (CD), the used products are completely dismantled while segregating the materials. Thus, all the components are treated as goal

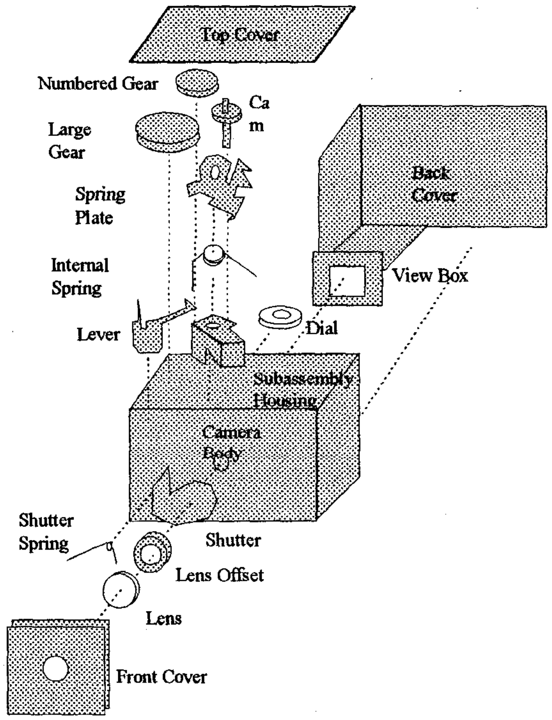

Fig. 1 Single-Use Camera

components. The third strategy, material segregation (MS), attempts to minimize the number of components to be removed. Hence, only the non-plastic components are treated as goal components.

5.1 Experimental Setup

The disassembly controlled system was modeled after an automated robotic kitting cell [9] at Georgia Tech. Specific resources used include a tool manipulator, a fixture which manipulates the product orientation to allow component removal by tool manipulator, and a vision sensor mounted on the end-effector of the tool manipulator. The resources and its blocking matrix [R] are summarized in Table 1.

In Table 2, the value of PO is the rotation in degrees between a defined home orientation and the orientation required to disconnect the component. The jamming probabilities for the snap-fit connections are based on the difficulties encountered in manually disassembly. The components in **bold** are components that user tampering is less likely and are potentially reusable. The DT is the total time taken for the series of elemental operations required to disassemble the component and was determined experimentally using the lowest operating velocity (75 mm/sec) of a Cincinnati Milacron T3 robot.

The single-use camera has 17 components constructed of plastic and metal materials. The components are summarized in Table 2 where * denotes non-plastic components; TN refers the tool number in Table 1; DT is the disconnection time in seconds; PO denotes the product orientation; and JP characterizes the jamming probability of the components. The one values of the component blocking matrix [B] is given by Equation (17).

$$b_{1,3} = b_{3,6} = b_{4,13} = b_{5,17} = b_{5,18} = b_{6,7} = b_{7,4} = b_{7,8} = b_{7,12} = b_{8,18} = b_{8,19} = b_{9,10} = b_{10,11} = b_{12,2} = b_{13,14} = b_{14,15} = b_{14,16} = b_{15,16} = b_{16,2} = b_{17,2} = b_{18,2} = b_{19,2} = b_{19,9} = b_{19,10} = b_{19,11} = 1 \quad (17)$$

The sensor measure for the component is the area of an image object. Component feature data were recorded for increasing sample sizes from 10 to 100. It was found that the mean and standard deviation did not vary significantly for samples of 40 or larger.

Each time that the simulation was run represents a batch of product processing through the disassembly work cell. As preliminary simulation runs showed little variation in results as the number of runs was increased above 50.a batch size of 50 used products was chosen to provide a significant sample of output data for the test case. The real mean and standard deviation of the output statistics are not known but expected to be normally distributed.

5.2 Results and Discussions

Table 1 Resources

j	Resources	Resource Blocking Matrix [R]					
1	Tool manipulator	0	1	1	0	0	0
2	Product Fixture	1	0	1	0	0	0
3	Camera	1	1	0	0	0	0
4	Vacuum tool	0	0	0	0	1	1
5	Pryer tool	0	0	0	1	0	1
6	On/off Gripper	0	0	0	1	1	0

Table 2 Single-use Camera Components

I	Component	DT, s	TN	PO	JP
0	Null	0	0	0	0
1	**Base**	47.23	6	0	0
2	Top cover	65.95	5	0	0.8
3	Front cover	53.6	5	90°	0.4
4	Back cover	62.3	5	-90°	0.4
5	Labeled gear	40.48	4	0	0
6	Large gear	40.59	4	0	0
7	Cam	40.48	4	0	0
8	**Spring plate***	43.14	4	0	0
9	**H. E. Spring***	48.85	6	0	0
10	**Housing**	48.49	6	0	0
11	**Lever***	42.58	5	0	0
12	Lens	41.0	4	90°	0
13	Lens offset	41.0	4	90°	0
14	Shuttle spring*	53.4	4	90°	0
15	Shutter*	40.5	4	90°	0
16	**View box***	38.3	5	-90°	0
17	Dial	41.4	4	-90°	0
18	**Subassembly (8, 9, 10)**	25.39	6	0	0.4

Some of the results obtained are shown in Fig. 2 and Table 3. A more detailed discussions can be found in [10]. Fig. 2 shows typical tool changes of the disassembly controlled system as a function of time for the three different strategies. Each discrete point represents the completion of a component removal. The results indicate that the TD strategy took approximately 12.2 minutes to reach the goal state, 30% and 50% shorter than the MS and the CD strategies respectively. The smaller goal set and the fact that the set was located toward the product exterior and required mostly the same tools accounts for this result.

When the disassembly is implemented in a production

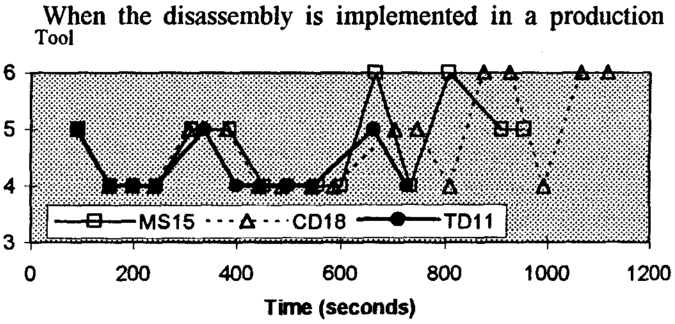

Fig. 2 Number Of Tool Changes As A Function Of Time

Table 3 Performance Comparison among Strategies

Strategy	CD	MS		TD
Goal Components	18	6	15	11
Tool changes	9	10	8	6
Orientation changes	4	3	4	3
Time-to-goal (sec.)	1116	975	950	730
Components Removed	18	15	15	11

line approach where modular stations are arranged in sequence, the number of tool changes to achieve the goal product state has a direct effect on the system layout. Fig. 2 shows the disassembly sequence { 2; 5, 6, 7; 3; 12, 13, 14 15; 4; 17} of the TD strategy implemented using the production line approach. The 10 modular stations are required to achieve a throughput of 40 units per hour operated at an average machine utilization of 84%. For the same throughput, the MS and the CD strategies would require 13 and 16 stations at an average machine utilization of 81% and 77% respectively.

Although the MS strategy has only 6 goal components, the number of tool changes was higher than that of the CD strategy. Recall that the system optimization is essentially a two-step process. The first step is to satisfy Equation (7) which aims at minimizing the number of disconnections. The second step minimizes the cost function given by Equation (16). If the number of goal components is relatively small, Equation (7) tends to yield a unique or smaller set of components to satisfy the first criteria. Since all components are goal components for the CD strategy, the secondary criteria that minimizes tool changes is called more often than that of the MS strategy with 6 goal components. The primary criteria of removing goal components becomes more prominent for material segregation. The minimization of the cost function may not be considered. Consequently, the number of tool changes for the MS strategy with 6 components was greater than that of the CD strategy.

Due to the component connectivity, 15 components must be removed to acquire the goal in the MS strategy. The simulation was repeated with the 15 components removed specified as goal components. As compared to the CD strategy, the MS strategy (with 15 goal components) shows a significant improvement in reducing tool changes and the number of components removed to achieve the objective.

6. CONCLUSIONS

A model for automated disassembly was developed that describes the used product and workcell states. Based on the concept of blocking topology, the model accounts for workcell interaction and used product constraints. It was shown that the next component for disassembly can be selected from the knowledge of the product design and the current component for disassembly that minimizes the steps to removing goal components. Sets of components for removal were resolved by minimizing set-up time for disassembling the component.

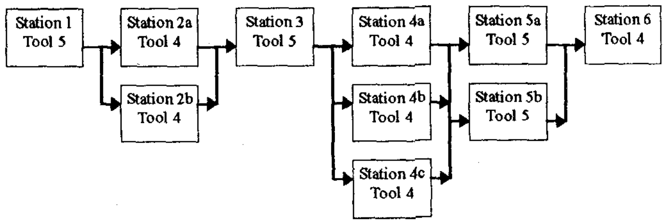

Fig. 3 Implementation of TD Strategy: Product Line Approach

A method for utilizing sensor feedback to detect differences from the expected design configuration was presented. Simulation results based on real product and process input showed that the model can be used under different disassembly strategies to provide performance measures of the disassembly process. Furthermore, simulation results can be extended to provide disassembly line proposals based on a disassembly strategy.

REFERENCES

1. Navin-Chandra. D. "Design for Environmentability," *Third International Conference on Design Theory and Methodology*, Miami, Florida, American Society of Mechanical Engineers, (1991), pp. 119-125.
2. Burke, D. S., K. Beiter, and K. Ishii. "Life-Cycle Design for Recyclability," *Fourth International Conference on Design Theory and Methodology*, Miami, Florida, ASME, (1992), pp. 325-332.
3. Ashley, S. "Designing for the Environment," *Mechanical Engineering*, V. 115, 3 (1993), pp. 53-55.
4. Jovane, F. *et. al.*, "Key Issue in Product Life Cycle: Disassembly", *CIRP Annals*, v42 n2 1993, pp. 651-658.
5. Weigi, Alexandra, "Requirements for Robots Assisted Disassembly of Not Appropriately Designed Electronic Products: Lessons from First Studies", *Proceedings of the 1994 IEEE Int. Symposium on Electronics & the Environment*, 1994, IEEE Service Center, Piscataway, NJ, USA, pp. 337-342.
6. Dario, Paolo and Rucci, Michele, "Approach to Disassembly Problems in Robotics", *1993 International Conference on Intelligent Robots and Systems*, 1993, IEEE Service Center, Piscataway, NJ, USA, (IEEE cat n 93CH3213-6), pp. 460-467.
7. Spath, D., "Utilization of Hypermedia-Based Information Systems for Developing Recyclable Products and for Disassembly Planning", *CIRP Annals*, v. 43 n 1, 1994, pp. 153-156.
8. Schmult, Brian, "Autonomous Robotic Disassembly in the Blocks World," *International Journal of Robotics Research*, Vol. 1, N. 5, October 1992.
9. Lee, K.-M. and Qian, Y., "Intelligent Vision-based Part-feeding on Dynamic Pursuit of Moving Objects," Symposium at SPIE's Photonics East, Philadephia PA, 23-26 October 1995.
10. Bailey-Van Kuren, M. M. "Automated Cell Supervisory Control for Product Disassembly", Ph.D. Thesis, Georgia Institute of Technology, 1996.