

The Structure Representation for the Concurrent Analysis of Product Assembly and Disassembly

Jyh-Cheng Yu* and Yi-Ming Li

*Department of Mechanical and Automation Engineering
National Kaohsiung First University of Science and Technology, TAIWAN, R.O.C.
jcyu@ccms.nkfust.edu.tw

Abstract

This paper presents a simple and novel structure representation supporting the assembly and disassembly planning of electromechanical products. The proposed Relationship Matrix derived from a directed graph represents both the information of the component connectivity and the layout precedence of functional elements. The feasible assembly and disassembly sequences and the minimum service steps for the malfunction component can be easily derived with the corresponding inference rules. The structure representation and the inference kernel can be readily applied to future concurrent design review for assemblability, serviceability, and recyclability.

Keyword: Layout Design, Concurrent Design, Sequence planning, Relationship Matrix, Product design

1. Introduction

Product layout designs will affect assembly and disassembly sequences that are crucial factors in design for assembly[2], serviceability[3], and recyclability[4]. Design for assembly aims to improve the assembly process by reducing the part count, making individual parts easier to assemble, and reducing the possibility of assembly errors. Design for disassembly, on the other hand, focuses on maintainability and recyclability, such that flawed components can be easily accessed and replaced during the service and valuable resources can be efficiently retrieved at the end of the product life cycle. The product objects must be well arranged in the product space to obtain the optimum product architecture.

Several modeling strategies have been proposed for the design representation of product architecture, such as directed graph [7], the liaison diagram [8], and the Component-Fastener Graph [6]. The connectivity and interference information is used to derive the assembly and disassembly sequences. Some suggested the directional Interference Matrices from a CAD model to describe the spatial interference in disassembly procedure and to automatically generate all the possible assembly[10] and disassembly[11] sequences. However,

detailed CAD models are often not available at the early stage of design and the slanted insertion of assembly can't be described using the interference matrices of principal directions. Homem and Sanderson [9] proposed the AND/OR graph, a compact representation of all possible assembly plans of a given product.

Recent works focus on the disassembly sequencing [5] due to the rising attention on the product maintenance and the end-of-life processing where incomplete disassembly is often preferred. However, the analysis of assembly and disassembly should be a continuous trade-off of life cycle factors, and should be considered in a simultaneous fashion.

This study presents a simple and novel structure representation for the conceptual design to support the assembly and disassembly planning of electromechanical products. With the proposed inference rules, the feasible assembly sequences and the required disassembly sequence for the service part can be derived to perform concurrent analysis for conceptual designs.

2. Product Assembly and Disassembly

The ease of product assembly and disassembly is the common requirement in the life-cycle design such as design for assembly, serviceability, and recyclability. The derivation of feasible assembly sequences is important in the assembly optimization. Product assembly can be deemed as a process to establish the structure relationship of the composing objects. The process often starts from a base part, such as the rear housing in Figure 1, secured in a fixture, and adds the following components to the main assembly. Auxiliary tools, such as screwdrivers, wrenches, etc., might be required to complete the assembly. The sequencing should consider the feasibility and ease of assembly. For instance, the components assembled earlier should not obstruct the assembly of subsequent components. Also, we should group the assembly steps together for the components using the same tools.

Sometimes, to ease the assembly effort, several components might be grouped together to form a subassembly before attaching to the main assembly. The subassemblies could be optional or mandatory. The constituents forming the optional assemblies can also be

put to the main assembly sequentially instead of a group. On the contrary, the constituents of the mandatory subassembly have to be assembled together first, otherwise they cannot be inserted to the main assembly. For the example in Figure 1, the front bearing must be inserted to the front housing, and fastened by the retainer and three short screws to compose a front module before assembled to the rear housing. This front module is a mandatory subassembly because the constituents cannot be sequentially put into the main assembly based on the rear housing.

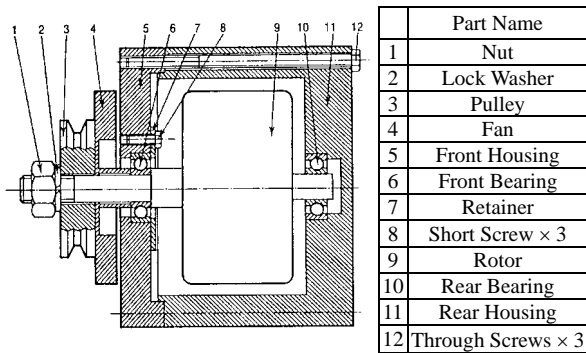


Figure 1. The product layout of an automotive generator

It is unavoidable that products might require service during their life cycle. If a certain part of a product fails, technicians need to retrieve the part for maintenance or replacement. The objects block the way to access the flawed part and those connected to it have to be removed before the service. To reduce labor costs, those requiring frequent service should be placed in the outer region of the layout space. Therefore, all the service procedures have to be considered at the early stage of design to obtain the best structure layout.

Due to the lack of design representation of product layout, the analysis of product assembly and disassembly usually occur late at the prototyping stage, which often incurs higher design cost if any modification is required. This study will propose a simple scheme to describe the product structure of fixed-form constituents and the inference rules for assembly and disassembly sequences.

3. Representation of Product Structure

3.1. The Constituent Objects of the Product

Product Function Analysis [1] is often used in the conceptual design to transform customer requirements to product functions. Design engineers then conceive specific objects and mechanism to realize the terminal functions. The assembly of the conceived objects composes a conceptual design. To obtain the product stability and rigidity, connecting features and fasteners, such as screw, bolts, and retaining rings, will be introduced.

To reduce the searching space in the analysis assembly and disassembly sequences, this study classifies the product constituents into the functional elements and the fasteners. The functional elements are the constituents that realize certain customer requirements. The fasteners, such as screws, rivets, retaining rings, etc, on the other hand, are auxiliary objects mainly to maintain structure stability and rigidity (Figure 2). Because the fasteners always accompany the corresponding parts to fasten during the product assembly, we just need to find out the correct order of the functional elements, and the complete assembly steps can be readily derived.

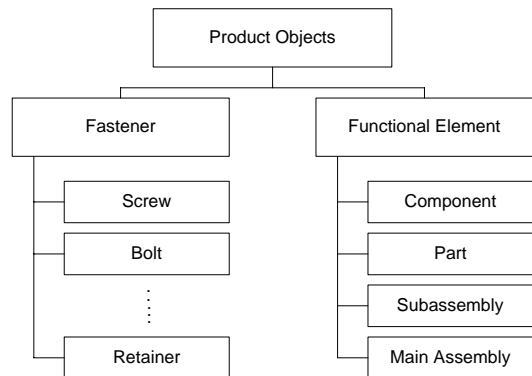


Figure 2. Classifications of Product Objects

Table 1. The classification of the composing object of the generator

Functional elements	Fasteners
Pulley	Nut
Fan	Lock Washer
Front Housing	Short Screws (3)
Front Bearing	Through Screws (3)
Rotor	Retainer
Rear Bearing	
Rear Housing	

For the example of the automotive generator shown in Figure 1, the classification of the composing objects is presented in Table 1. Another reason to identify the fasteners in a product structure is that the fasteners are usually the targets to redesign for assembly. DFA encourages incorporating the fastening feature to the functional element to reduce the part count. Designers should only have legitimate reasons to use auxiliary fasteners. Also, the fastening method should consider the ease of disassembly and reassembly if the functional element to secure will need service during the life cycle.

3.2. The Object Relationships

The relationship among functional elements can be divided into two categories: the Physical Links and the Layout Interference. Physical links represent direct connection of two objects by either geometrical

interlock or auxiliary fasteners. The other relationship that is termed “Layout Interference” exists when an object affects the assembly sequence of other objects due to spatial arrangement, even though there is no direct contact among these objects. It is impossible to finish the assembly if certain object is assembled before the other when there is a Layout Interference between these two objects. Here the interference relationship is a general concept and doesn’t differentiate the interfering direction because the detailed object geometry is often undecided at the conceptual design stage.

For instance in Figure 3, the relationship between objects *C* and *B* is a Physical Link since object *C* is connected to object *B* by a snap feature. Another example of Physical Link exists between objects *D* and *B*; however, the connectivity is established by a screw fastener. On the other hand, there is no direct contact between objects *F* and *E*, but object *E* must precede object *F* in the assembly sequence due to the covering effect. This remote relationship between objects *F* and *E* is termed the “Layout Interference”.

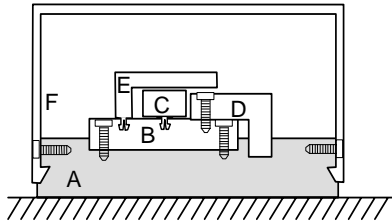


Figure 3. Example of product layout

This study combines the concept of interference and connectivity, and proposes the Object-Relationship Graph (OR-Graph) to represent the product structure. The attributes of the object contain information of the object category, the part name, the material type, and the part weight; while the attributes of the relationship contain the relationship type, the assembly method, and the fastener type if individual fasteners are applied. During the assembly, one object usually has a precedence preference over the other because of size, weight, and ease of assembly. Although it is not necessary for all the Physical Links, the pre-assigned information can greatly reduce the searching space of assembly and disassembly sequences.

The OR-Graph is a directed graph where the arrow direction indicates the next assembly occurrence. For each relationship O_i-O_j , O_i is called the *related object* and O_j is called the *relating object* if O_j precedes O_i in the assembly sequence because of either the assembly conventions or the Layout Interference. The OR-Graph is established from the *relating object* to the *related object*. The corresponding OR-Graph for the previous layout is shown in Figure 4. For examples, the direction of the dashed link connecting objects *E* and *C* indicates a Layout Interference that *C* should precede *E* in the assembly sequence otherwise object *E* cannot be installed. The direction of the solid link connecting

objects *B* and *A* indicates a Physical Link that *A* precedes *B* due to ease of assembly.

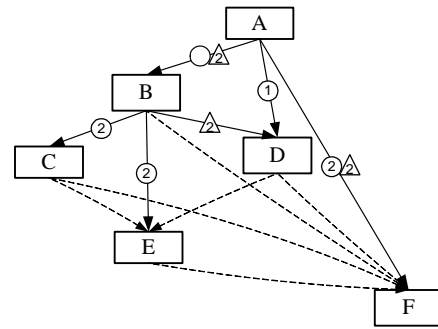


Figure 4. The OR-Graph of Figure 3

3.3. Relationship Matrix

The OR-Graph is modeled as a matrix form, the Relationship Matrix (*RM*), to facilitate the derivation of the assembly and the disassembly sequences. The Relationship Matrix is an $n \times n$ matrix recording the object relationships where n is the number of the functional elements. The functional elements are listed in the columns and the rows of *RM* in the same order. The rows indicate the *related objects*, and the columns indicate the *relating objects*. The relationship between O_i and O_j is denoted in $RM(i, j)$. The corresponding Relationship Matrix of the layout example in Figure 3 is shown in Table 2. The notations used in this paper are explained as follows:

Table 2. The Relationship Matrix of the example layout in Figure 3

$$RM = \begin{matrix} & \begin{matrix} \text{F} & \text{D} & \text{C} & \text{B} & \text{E} & \text{A} \end{matrix} \\ \begin{matrix} \text{F} \\ \text{D} \\ \text{C} \\ \text{B} \\ \text{E} \\ \text{A} \end{matrix} & \begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 1 \\ 0 & 0 & 0 & 3 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 2 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

- (a) “0” if there is no relationship between O_i and O_j .
- (b) “1” for a Physical Link between O_i and O_j . O_i is the *related object* and O_j is the *relating counterpart* of the relationship pair O_i and O_j . For instance in Figure 3, object *E* is connected to object *B* with a snap feature, and the entry at row *E* and column *B* is denoted by “1”.
- (c) “2” for a Layout Interference between O_i and O_j . For instance in Figure 3, object *E* will prevent the assembly of *C* and *D* if *E* is installed before them. Here *C* is the *related object* for this interference relationship, and the entries at row *E* and columns *C* and *D* are denoted by “2”.

- (d) “3” if there are both a Physical Link and a Layout Interference between O_i and O_j , such as object D and object B in Figure 3.

The Relationship matrix of a more realistic example such as the automotive generator of Figure 1 is shown in Table 3. The matrix considers only the functional elements to reduce the dimension. The object order in the initial RM is randomly selected. The *relating objects* are identified for each relationship of *related object*, and their corresponding relationship codes are filled into the entries. For example, the Pulley is inserted to the Rotor and in contact with the Fan. Also, the Pulley will interfere the assembly of the Fan if installed first. Therefore, the entries of the row “Pulley” at the columns “Fan” and “Rotor” are 3 and 1 respectively. Two fasteners, the Nut and the Lock Washer, secure the relationship between the Pulley and the Rotor, which are not shown in the matrix and will be introduced later when the complete assembly steps of the Pulley is considered.

Table 3 The Relationship Matrix of the automotive generator

	Pulley	Fan	Front housing	Front bearing	Rotor	Rear bearing	Rear housing
Pulley		3	0	0	1	0	0
Fan	0		0	0	1	0	0
Front housing	0	0		2	2	2	1
Front bearing	0	0	1		1	0	0
Rotor	0	0	0	0		3	0
Rear bearing	0	0	0	0	0		1
Rear housing	0	0	0	0	0	0	

4. The Inference of Assembly Sequences

4.1. The Characteristics of Rational Relational Matrix

When the row order of the objects from the top of the matrix represents a feasible assembly sequence, the Relationship Matrix is termed rational and will present a particular pattern. The matrix is often a lower triangular matrix with all zero entries above the diagonal. For instance, Table 4 is a rationalized RM of the example layout in Figure 3.

- (1) No Physical Link exists in the entries located at the upper triangular of the rational Relationship Matrix, RM_r . Because the rows represent the *relating objects* of the relationship pairs, if an object will only assemble to the precedent objects, there will be no Physical Link between the object and the following objects, and the row order will be a feasible assembly sequence.
- (2) The number of Layout Interference in the upper triangular of the RM_r is minimized. For a feasible assembly sequence, the earlier objects should not obstruct the assembly of later objects. Therefore, there should be no Layout Interference in the

upper triangular of the RM_r , unless the object is the base of a mandatory subassembly that will be addressed in the later section.

Table 4 The rationalized Relationship Matrix of the example in Figure 3

	A	B	C	D	E	F
A	0	0	0	0	0	0
B	1	0	0	0	0	0
C	0	1	0	0	0	0
D	1	3	0	0	0	0
E	0	1	2	2	0	0
F	1	2	2	2	2	0

4.2. The Inference of a Feasible Assembly Sequence

The initial construction of the Relationship Matrix doesn't guarantee rational. Through the following row and column operations, we could rationalize the RM and obtain a feasible assembly sequence:

- (1) Move the entries with the Physical Link on the upper triangular to the lower triangular by exchanging the object order. If $RM(i, j)$ is “1” or “3” and $j > i$, exchange the column i with the column j and the row i with the row j . The purpose is to rearrange the object order in the RM such that for all the Physical Links the *relating objects* will always precede their *related* counterparts.
- (2) Move the entries with the Layout Interference on the upper triangular to the lower triangular by exchanging the object order. If $RM(i, j)$ is “2”, $j > i$, and $RM(j, i) \neq 1$, exchange the column i with the column j and the row i with the row j .

Repeat the above two transformations until the following two conditions are met, and the Relation Matrix is called rational, RM_r .

Rational condition #1:

For $j > i$, $RM_r(i, j) \neq 1$ and $RM_r(i, j) \neq 3$.

Rational condition #2:

For $j > i$, $RM_r(i, j) \neq 2$, unless $RM_r(j, i) = 1$.

The rational RM_r is not unique. We can continue to search for other rational RM_r 's. However, the proposed methodology is not intended for the exhaustive search of assembly sequences. The object order listed in the column or the row of every rationalized RM_r would infer a feasible assembly sequence. For instance, Table 4 is a rationalized RM of the example layout in Figure 3. Therefore the object order in the RM_r , i.e., A, B, C, D, E , and F , represents a feasible assembly sequence. Because the rational matrix contains only the functional elements, the complete assembly sequence can be readily obtained

by inserting the auxiliary fasteners after the corresponding objects to be constrained.

4.3. Recognition of Mandatory Subassemblies

If the product structure contains certain mandatory subassemblies, we cannot assemble all the objects to the main assembly in a sequential order. The existence and the members of the mandatory can be recognized from the rationalized Relationship Matrix.

Theorem #1: If the Relationship Matrix RM_r is rational and $\exists RM_r(i, j) = 2, j > i$, the structure contains at least one mandatory subassembly. The number of the mandatory subassemblies is equal to the number of rows contains entry “2” in the upper triangular of RM_r .

Theorem #2: The sub-base of the mandatory subassembly is the object in row i where $RM_r(i, j) = 2$ and $j > i$. The members of the mandatory subassembly are the objects appeared in the upper triangular and interfered by the sub-base.

Table 5 is the rationalized Relationship matrix of the automotive generator in Figure 1. The matrix satisfies both the rational conditions. The entries at the upper triangular are all zero except the one of the row “Front Housing” and column “Front Bearing”. The Layout Interference could not be moved to the lower triangular because the entry at row “Front Bearing” and column “Front Housing” is “1”. According to the theorem #1, there is a mandatory subassembly and the Front Housing is the sub-base. The members of the subassembly include the Front Bearing that is interfered by and appeared after the “Front Housing”. During the assembly, the “Front Bearing” is fastened to the “Front Housing” by a “Retainer” and three “Short Screws” to form the “Front Cover Module”. The “Front Cover Module” is then assembled to the main assembly by the Physical Links between the members of the subassembly and the rest of the functional elements. In this case, the subassembly is inserted to the “Rotor” and attached to the “Rear Housing” by three “Through Screws”.

Table 5. The rationalized Relationship Matrix, RM_r , of the automotive generator

	Rear housing	Rear bearing	Rotor	Front housing	Front bearing	Fan	Pulley
Rear housing		0	0	0	0	0	0
Rear bearing	1		0	0	0	0	0
Rotor	0	3		0	0	0	0
Front housing	1	2	2		2	0	0
Front bearing	0	0	1	1		0	0
Fan	0	0	1	0	0		0
Pulley	0	0	1	0	0	3	

After the recognition of mandatory subassemblies, the Relationship Matrix can be divided into certain sub

Relationship Matrices, Sub- RM , and the main Relationship Matrix, Main- RM_r . The Sub- RM describes the relationship among the members of the subassembly such as the Front Cover Module in the example of the automotive generator. The relationship between the subassembly and the rest of the functional elements are combined from the members of the subassembly as shown in Table 6.

Table 6. The Sub-Relationship Matrix of Front Cover Module of the automotive generator

	Front housing	Front bearing
Front housing		2
Front bearing	1	

Table 7. The Main-Relationship Matrix of the automotive generator

	Rear housing	Rear bearing	Rotor	Front Cover module	Fan	Pulley
Rear housing		0	0	0	0	0
Rear bearing	1		0	0	0	0
Rotor	0	3		0	0	0
Front cover module	1	2	3		0	0
Fan	0	0	1	0		0
Pulley	0	0	1	0	3	

4.4. Generation of Assembly Steps

The row order in the rationalized Relationship Matrix provides a feasible sequence of the structure. The fastening methods are the attributes of the entries of the RM . The insertion of the fasteners to secure the corresponding objects will generate the complete assembly steps. The assembly of the mandatory subassembly should be done first.

5. The Inference of Disassembly Sequences

To remove an object for service, technicians have to remove the objects blocking the access to the serviced part and disconnect all the objects attached to it. We can identify these objects from the rational Relationship Matrix using the depth-first-search method. The inference rules will provide the minimum disassembly and reassembly procedure, and some objects will be removed as a subassembly if further disassembly is not required. The inference scheme is as follows:

- (1). The serviced object is first added to the Disassembly Queue (DQ) that is a First-In-Last-Out processing queue.
- (2). Add the *related objects* that have the relationship of either the Physical Link or the Layout Interference to the serviced part to the DQ . These objects can be identified from the column of the

serviced part, and are put to the *DQ* according to their assembly order.

- (3). Take one object from the *DQ*, and check the disassemblability. If the target object is not interfered by any other object, remove the target object. Otherwise, add those interfering the target object to the *DQ*.
- (4). Repeat step (3) until all the objects in the *DQ* are removed.

Consider the object *A* of the example layout in Figure 3 has to be replaced. From the column of “*A*” in Table 4, we can identify that the objects *B*, *D*, and *F* have to be added to the *DQ* along the object *A*. Object *F* is the first retrieved from the *DQ* for disassembly process. Because no object interferes object *F* as seen from the column “*F*” of Table 4, dismantle the fasteners of object *F* if any, and remove object *F* from object *A*.

The next object in the *DQ*, object *D*, cannot be removed directly because *D* is interfered by *E* that is then added to the *DQ*. Continue the disassemblability check of each object in the *DQ* until the target object *A* is finally retrieved. According to the proposed inference scheme, the minimum disassembly steps to retrieve object *A* can be obtained as follows:

- (1). Unscrew the Fastening Screws of object *F*, and remove object *F* from object *A*.
- (2). Remove object *E* from object *B*.
- (3). Unscrew the Fastening Screw of object *D*, and remove object *D* from object *B* and object *A*.
- (4). Unscrew the Fastening Screws of object *B*, and remove object *B* from object *A*.
- (5). Remove the target object *A*.

Noted that object *C* is not in the *DQ* because *C* does not interfere the removal of *B* that can be observed from the Relationship Matrix. Therefore, object *C* is removed along with object *B* in disassembly step 4 without disconnecting the relationship between *C* and *B*. The reassembly sequence after the service is simply the reverse order of the derived disassembly sequence.

6. Conclusions

This paper presents the Relationship Matrix to describe the assembly architecture of an electromechanical structure. The single matrix derived from directed graph represent both the information of the component connectivity and the layout precedence of functional elements, which greatly reduce the searching space. The proposed methodology provides a simple solution for the concurrent analysis of assembly and disassembly at the early stage of design when the detailed geometry of product constituents is not yet available. Using the proposed inference rules, the mandatory subassemblies can be identified, and feasible assembly sequences and the minimum service steps for

the malfunction component can be easily derived. The structure representation and the inference kernel can be readily applied to concurrent design review for assemblability, serviceability, and recyclability. However, the memory requirement and the searching efficiency using the matrix representation for a complex system will become a concern. The applicability of the proposed scheme to complex products is pending future investigation. The introduction of Entity Relationship models, objected oriented data structure, and the design data standardization scheme such as STEP will be a solution to improve data integrality and exchangeability.

Acknowledgments

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 88-2212-E-011-017 and NSC 93-2516-S-327-005.

References

- [1]. D. Clausing, (1994), *Total Quality Development: A Step-by-Step Guide to World-Class Concurrent Engineering*, New York ASME press.
- [2]. G. Boothroyd and P. Dewhurst, (1983), *Design for Assembly: a designer's handbook*, Boothroyd Dewhurst Inc., Waverfield, Rhode Island.
- [3]. K. Ishii, C.F. Eubanks, and M. Marks (1993) “Evaluation Methodology for Post-manufacturing Issues in Life-cycle Design,” *Concurrent Engineering: Research and Application*, Vol.1, pp.61-68.
- [4]. R. W. Chen, D. Navin-Chandra, and F. B. Prinz (1993), “Product Design for Recyclability: A Cost Benefit Analysis Model and its Application”, *Electronics and the Environment*, IEEE, pp.178 – 183.
- [5]. Y. Tang, M. C. Zhou, E.Zussman, and R. Caudill (2000) “Disassembly modeling, planning and application: a review”, *Proc. - IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2197 - 2202
- [6]. T.C. Kuo, H.C. Zhang, and S.H. Huang (2000), “Disassembly analysis for electromechanical products: A graph-based heuristic approach” *Int. Journal of Production Research*, v 38, n 5, pp. 993-1007.
- [7]. N.S. Ong and Y.C. Wong (1999), “Automatic Subassembly Detection from a Product Model for Disassembly Sequence Generation”, *Int. J. of Advanced Manufacturing Technology*, 15, pp. 425-431.
- [8]. T. L. De Fazio and D.E. Whitney (1987) “Simplified Generation of All Mechanical Assembly Sequences”, *IEEE J. of Robotics and Automation*, Vol. RA-3, No. 6, pp. 640-658.
- [9]. L.S. Homem de Mello and A.C. Sanderson, (1990) “AND/OR graph representation of assembly plans”, *IEEE Transactions on Robotics and Automation*, Vol. 6, Issue: 2, pp. 188 – 199.
- [10]. G. Dini and M. Santochi (1992), “Automated Sequencing and Subassembly Detecting in Assembly Planning”, *Annals of The CIRP*, Vol.41, pp.1-4
- [11]. Y. M. Huang and C.-T. Huang (2002), “Disassembly matrix for disassembly processes of products” *Int. Journal of Production Research*, v 40, n 2, pp. 255-273.