An artificial intelligence approach for generating assembly sequences in CAD/CAM

Monjy Rabemanantsoa & Samuel Pierre*

LICEF Laboratory, Télé-université, Université du Québec, 1001, rue Sherbrooke Est, 4e étage, Case Postale 5250, Succ. C, Montréal, Québec, Canada, H2L 4L5

(Received 10 October 1994; accepted 21 June 1995)

This paper presents an innovative artificial intelligence approach for generating assembly sequences on a consortium of database emulating expert systems. Procedures include shape and feature recognition using a model-based CAD (computer-aided design) analyser, data structure and data modeling, knowledge-based representation, and inference processing through a set of heuristics and rules. The main tool here is an object-oriented concept as a means for managing geometrical data, topological data and abstraction. Abstractions added to data semantics help to build a knowledge base to meet assembly requirements. As a result, an AI paradigm supports knowledge processing for the search of good solutions and good decisions in the assembly process. An efficient implementation of the integrated system has been achieved and successfully demonstrated by results based on parts with varying complexities.

Key words: artificial intelligence, knowledge-based system, object-oriented database, assembly system, CAD/CAM.

1 INTRODUCTION

Conceptual assembly is usually a complex task involving geometric and physical constraints between components and it therefore requires a large amount of information and computation. In general, a feature-based modeling database may contain the following types of information: 1 a list of all instances of all form features; a generic definition of each feature in terms of generic parameters, geometry and topology; a relationship between form features (dependencies and adjacencies); a geometric model of the entire part; material properties and other nongeometric attributes; tolerances and surface finish where appropriate. The next step is to build up a method of interpreting feature information for manufacturing purposes. A complete scheme integrating these issues is addressed: database modeling from CAD design, a knowledge-based representation, and an inference engine for knowledge processing.

The parts or products are designed with a parametric and feature-based solid modeling CAD system, Pro/Engineer. Data are then stored in two related formats: IGES and NEUTRAL files. We designed an object-oriented database written in C language called SISDES (System of Integrated Software for Data Exchange Specification) to support a model analyser for surface and feature recognition. Given the concept of abstraction and data semantics, the system carries out a knowledge-based representation in which functions of 'contact' and 'relative mobility' for mating conditions are embedded. An inference engine is in charge of processing knowledge functions for the purpose of generating assembly sequences. To do so, the forward chaining method has been adopted into a problemsolving software called XGEN (Assembly Sequences Generation) developed in Mprolog. The rest of this paper is organized as follows: section 2 summarizes the basic concepts and discusses related work; section 3 deals with feature-based design translation, while section 4 presents database modeling and describes the methodology for managing information needed to achieve this process; section 5 is dedicated to knowledge-based representation using AI techniques called 'decision tree' and 'production rule'; knowledge processing is described in section 6, and section 7 presents the system outputs in terms of tests and results to show proof of methodology's correctness. This paper is a comprehensive survey of some recent conference publications2-5 in which AI applications in CAD/CAM are discussed.
2 BASIC CONCEPTS AND RELATED WORK

A mechanical assembly is a set of interconnected parts representing a stable unit in which each part is a solid object. Surface contacts between parts reduce the degrees of freedom for relative motion. Attachments by means of screw, glue or pressure act on surface contacts and eliminate all degrees of freedom for relative motion.6

A subassembly is a non-empty subset of parts having one element or more in which every part has at least one surface contact. The assembly process consists of a succession of tasks, in each of which subassemblies are joined to form a larger subassembly. Thus any subassembly can be characterized by its set of parts, and any state of the assembly process can be characterized by a partition of the set of parts of the whole assembly. This process starts with all parts separated; at the end, all parts are put together.6

In this paper, the idea is to structure contents in order to obtain pattern-accessible data that allows us to answer questions such as: 'Which two-holed surface, in the first part, fits the subassembly with a shaft?' In this case, automatic recognition is required of interacting form features such as holes, slots, pockets. An important use of CAD database facilities is made in the construction of a knowledge-based system handling the details needed to capture and illustrate assembly information. The problem is not only to make all information available but also to use all relevant information automatically for good decisions in design and planning.

Jackson and McMaster7 presented the basis for establishing a set of design rules for automated assembly, and some generalized design guidelines. For a similar purpose of effective and efficient use of assembly techniques, Conradson and colleagues8 tested an automated assembly in a high-mix product environment. The process of locating components in an available space while satisfying spatial relationships among the components was introduced by Kim and Gossard;9 the process called 'packaging' is formulated as a constraint optimization problem in a solid environment. Because of the geometric and physical constraints of mating conditions in an assembly process, Huang and Lee10 developed a geometric mating graph including all the necessary geometrical and topological information; the issue titled Precedence Knowledge in Feature Mating Operation Assembly Planning focuses on the important role of knowledge acquisition in the generation of assembly sequences.

It is appropriate to mention research work carried out by De Fazio and Whitney,11 who developed the assembly process by a sequence of states leading to a direct graph of assembly states. An approach based on geometric reasoning was proposed by Nnaji.12 The architecture includes the concept of features and shapes identification of the objects to be assembled. Finally, Hernani and Scarr15 presented a comprehensive range of design rules for automated assembly. The proposed methodology uses a CAD package and an expert system to provide the designer with an interactive means of accessing product design information.

Based on this reasoning, considerable efforts have been made to build an object-oriented database in the quest for solutions to the geometrical and physical constraints of assembly feasibility. The result is a knowledge-based system that integrates a set of heuristics and rules. We have developed a new approach for integrating AI with DB in CAD field applications where DB supports data processing and AI supports knowledge processing.

3 FEATURE-BASED DESIGN TRANSLATION

The approach used for automated assembly consists of integrating constructive solid geometry (CSG) with a database; the methodology for managing information and relations among the entities emulates the object-oriented technique for a CAD database management...
Assembly sequences in CAD/CAM 99

system. Figure 1 presents the integrated system used for generating assembly sequences.

3.1 Module interaction

Once the products or parts have been designed by the parametric feature-based solid modeling system, Pro/Engineer, related data are stored in two data exchange formats, IGES and NEUTRAL. The next module, SISDES, is a DB dedicated to the effective management of design information by translating data from the two files mentioned above.

The knowledge base is developed in accordance with the artificial intelligence paradigm. The method known as 'production system' was adopted from among other methods such as logical predicate, semantic network, frame. Then XGEN, the AI problem-solving software, handles the rule base and facts of the knowledge formalism for generating mechanical assembly sequences. The main tool here is IGES. However, IGES has a few weaknesses. For instance, it is difficult to use IGES to distinguish adjacent surfaces and edges common to two parts. Consequently the adoption of a database management system requires the presence of another file, such as the NEUTRAL file available in the Pro/Engineer system.

To remedy the lack of structure in the IGES system, other international standards such as PDES and STEP are being studied and hopefully they will be implemented in the near future. Mufti and colleagues14 define PDES as 'Product Definition Exchange Specifications' and STFP as 'Standard for the Exchange of Product Model Data'. Both are expected to support wire-frame model and data structures for solid modeling. Since 1983, Aerospatiale (France) have also developed SET, which is functionally equivalent to IGES, while VDA-FS was developed under the leadership of Germany's Verband der Automobilindustrie, focusing only on the area of free form surfaces.15

3.2 Structure of the IGES file

The structure of the IGES file is based on the concept of 'entity'. In all, there are about 50 entities, depending on the various forms of the drawing, and this leads to about 150 subformats because each entity has 3 main characteristics:

- geometry: point, line, circle, surface
- annotation: dimension, titles, notes
- structure: combination of geometrical elements

IGES has two basic formats: binary and compressed ASCII. In most cases, postprocessors use the second format and the files are divided into five sections: start section, global section, directory entry section, parameter data section, and terminate section. Figure 2 is a typical IGES file printout showing section D (directory entry section) and section P (parameter data section). In extracting data for assembly process planning, these sections too have to be used simultaneously. For

---

Section D

| 110 | 0 | 0 | 1 | 0 | LINE 430 | 132 |
| 110 | 0 | 1 | 1 | 0 | LINE 440 | 134 |
| 102 | 63 | 1 | 1 | 0 | 0 | 0 | 0010100000 | 133 |
| 102 | 0 | 0 | 1 | 0 | CURVE 100 | 136 |
| 108 | 69 | 1 | 1 | 0 | 0 | 0 | 0000000000 | 137 |
| 108 | 0 | 0 | 1 | 1 | PLANE 100 | 138 |
| 110 | 70 | 1 | 1 | 0 | 0 | 0 | 0001010000 | 139 |
| 110 | 0 | 0 | 1 | 0 | LINE 450 | 140 |
| 110 | 0 | 1 | 1 | 0 | 0 | 0 | 0010100000 | 141 |

Section P

| 124,-1,0,0,400,0,0,1,150,0,1,0,0,0,0, ; | 111P 56 |
| 100,0,170,50,195,50,195,50, ; | 113P 57 |
| 124,-1,0,0,400,0,1,150,0,1,0,0,0,0, ; | 115P 58 |
| 100,0,170,50,195,50,195,50, ; | 117P 59 |
| 102,2,113,117, ; | 119P 60 |
| 108,0,1,0,1,150,109,0,0,0,0,0,1,125, ; | 121P 61 |
| 108,0,1,0,1,150,119,0,0,0,0,1,125, ; | 123P 62 |
| 402,1,1,21,123, ; | 125P 63 |
| 110,100,300,0,100,150,0,1,150,100,100,300,100, ; | 127P 64 |
| 110,100,300,0,100,150,0,1,150,100,100,300,100, ; | 129P 65 |
| 110,100,300,0,100,150,0,1,150,100,100,300,100, ; | 131P 66 |
| 110,100,300,0,100,150,0,1,150,100,100,300,100, ; | 133P 67 |
| 102,4,127,129,131,123, ; | 135P 68 |
| 108,1,0,0,1,100,135, ; | 137P 69 |
| 110,0,300,0,100,100,0,1,150,100,100,300,100, ; | 139P 70 |
| 110,0,300,0,100,100,0,1,150,100,100,300,100, ; | 141P 71 |

Fig. 2. Example of an IGES file.
instance, the pointer 71 in the tenth line and second column of section D refers to section P, where the coordinates of the extremities of a line are given as (100, 300, 100) and (100, 300, 0). Hence this is a line parallel to the z-axis. However, if the geometric entity is more complex (such as a surface or a compound curve), one needs to start from section P. For instance, pointer 133 in section P refers to pointer 67 in section D which brings one back to section P; this gives the dimensions of one of the lines that form the compound surface.

3.3 Specification of the NEUTRAL file

The NEUTRAL file is a basic CAD data file in Pro/Engineer created with its Pro/Neutral interface. The file may be created for a single part or for an assembly system with emphasis given to information about parts and subassembly relationships not supported by IGES; the information includes relations, attributes and an application interface. A typical format is in ASCII, in which lines beginning with the character # are comments. The other lines are in the following format: 'level', 'field' and 'value'. In the example illustrated in Fig. 3, the value of 2*0.0, 0.1 in the fifth line indicates that the X, y and z coordinates of the extremity of a line are 0.0, 0.0 and 0.1. The value represented by the character -+ indicates a pointer.

4 DATABASE MODELING

The processing provides a framework for integrating a CAD environment by providing a mechanism for managing and manipulating data with programs.

4.1 Object modeling

Object-oriented concepts are based on fundamental principles of complexity management which are very helpful for complex-object modeling and data manipulation related to complex entities such as CAD or CAE (computer aided engineering) system environments. Each object has its own identifier OID (object identifier). However, two similar objects with the same dimensions but a different OID are said to be different. Object identity allows ‘direct graph’ modeling, and objects are characterized by ‘properties’. A property may be an object characteristic such as an attribute, a function or a subobject component. For example, the object circle can have the following properties:

- simple attribute: radius (R)
- composite attribute: centre (x, y, z)
- function: surface (πR²)

In the assembly domain, the objects are typically composites or aggregates of components. The term ‘composite object’ is commonly used to denote a layered abstraction model. Through the development phases of composite objects, the OID provides a natural paradigm for maintaining the uniqueness of objects independent of structure or content. In addition, the use of an abstract data type enhances program modularity because modifying the object structure does not affect the manipulation of external objects. In our model, an abstract data type is implemented as an object collection with the same structure and representing a ‘class’.

4.2 Object dynamic binding

In the object-oriented concept, a class defines an abstract data type and allows us to describe the behaviour of a collection of objects in a modular way, e.g. circles with the same radius, polygons with a common shape. The concept of a class hierarchy eliminates the possibility of specifying and storing redundant information. The higher level mode is called a ‘superclass’ and a lower level mode a ‘subclass’. ‘Methods’ are operations that can either retrieve or update the state of an object. Data integrity is ensured by a procedure called ‘demon’. A demon is a triggered operation associated with an object when a particular condition happens (‘true’ predicate or operation execution). For example, in a cuboid class a demon may be associated with an object to maintain its state in a subassembly:

IF height > 2" THEN state = base_part

In our application, a superclass SOLID_3D represents an object-oriented model. The notion of data encapsulation within a programming language and the paradigm for creating any object as a nested object make up precisely the concept of complex objects illustrated in Fig. 4.

4.3 Database implementation

Geometrical and topological information is not sufficient for establishing design rules for assembly planning.
Geometric interference, physical constraints, and kinematic relationships are all important issues to be addressed. Thus the use of an object-oriented approach provides the tool, without any interactive means, for modeling, entities and properties, as well as for managing design information effectively.

SISDES manages the data needed for a complete integration into three areas: geometry, topology and tolerances. In our application, geometry refers to a solid model. In 3D, a solid is uniquely determined by its boundary, which consists of planar, cylindrical, spherical and sculptured surfaces. Individual parts are identified from IGES while common surfaces and common edges related to composite parts are extracted from the NEUTRAL file. Geometry helps to make a representation scheme for parts and products. Topology is the connectivity between geometric elements. Hurt gives the following definitions: 'Geometry is where the objects are located; topology is why they are located there.' Topology is therefore in charge of processing the rules for connecting elements of geometry to produce parts. Tolerances are allowable variations between design and manufacturing processes such that the various components can be assembled into a product.

Therefore data are stored as instances of abstract data types and all conceptual entities are objects. A class inherits all instances variables from its superclass. A CSG diagram represents a set of objects as a generic object and is equivalent to the point of view of generalization abstraction. Generalization abstraction captures the 'is-a' AI relationship:

- superclass = part: object
- class = primitives: cuboid/conoid/cylinder, etc.

Another concept of abstraction called 'aggregation' is added to model part-component structures. Aggregation abstraction involves hierarchies where object instances inherit the characteristics of higher level objects. It captures the 'is-part-of' AI relationship:

- surface: aggregation (curves + vertices)

The two formal approaches are used to automate recognition. An example of SLOT recognition in Fig. 5 could be as follows:

IF

Primitive = Cuboid_class
and surface_code of faces (F1,F2,F3) = F (flat)
AND
face F1 is adjacent to face F2
face F2 is adjacent to face F3
Edge U is_part_of F1 and F2
Edge KL is_part_of F2 and F3
and angle between F1 & F2 is $< 180$ (concave)
and angle between F2 & F3 is $< 180$ (concave)

THEN
Faces F1,F2,F3 form a feature SLOT

This procedure can also be used for the feature HOLE recognition in Fig. 5 by adding a Cylinder_class to the Primitive statement and by changing Edge statements to Curve_arc statements. Figure 6 shows the basic recognition scheme associated with abstractions: the quest for surface description is to use aggregation — referred to as a 'sweep generation and B-spline interpolation'.

The sweeping path is referred to as the 'spine', the
function swept is referred to as the ‘cross-section’, and the geometrical relationships between them are referred to as the ‘sweeping rule’. Surface recognition via sweeping is analogous to the translation of a ‘solid of revolution’ along a path. For instance, cone recognition is achieved by translating a circle, the radius of which is a linear function, along a straight line. In addition, the ‘B-spline method’ is used to complete the recognition of curved surfaces; the location of the curve depends on \( n \) neighboring control points and the mathematical representation is defined by

\[
P(t) = \sum_{i=0}^{n} P_i N_{i,k}(t) \quad 0 \leq t \leq n - k + 2 \tag{1}
\]

\( k \) represents the parameter that controls the order of continuity; \( N_{i,k}(t) \), the blending function, is defined as follows:

\[
P'(t, S) = \sum_{i=0}^{n} \sum_{j=0}^{m} P_{i,j} N_{i,j}(t) N_{j,1}(S) \]

The B-spline has local control, which enables a user to define a shape without connecting many pieces of curve together.

Aggregation and generalization can fully represent entities. Implementation based on data abstraction supports conceptual data modeling and knowledge-based representation.

5 KNOWLEDGE REPRESENTATION FORMALISM

Formalism is involved as a major driving factors of data modeling but also includes nongeometric attributes such as process specification in terms of definition of assemblies and subassemblies. To do this, the knowledge base integrates the automated-feature-recognition and position plus orientation needed for part mating. The possibility of contact and relative mobility for each pair of components is defined from rule-based specifications.

The objective of this approach is to achieve complete integration between CAD and CAM. Following Shah and Bhatnagar’s methodology, we developed our own specific data requirements according to explicit characteristics (type 1), implicit characteristics (type 2), and extrinsic characteristics (type 3). For example, a hole should be described in terms of diameter, length and an orientation vector; these parameters are of type 1 because they may be available in the database from IGES. The radius of a cylinder is of type 2 because it may be derived from the center coordinates and the diameter. However, if the angle between the axes of the two holes of equal diameter is needed, this type 3 information is neither in IGES nor in NEUTRAL; it will depend on the concept used for the model. In order to investigate planning for assembly, this approach formalizes the process of building assembly applications and allows the assembly planner to select and concentrate only on mating surfaces. The AI techniques of decision tree and production rule provide a better understanding of the surfaces: if horizontal surfaces \( i \) and \( j \) have the same ordinate, check whether these surfaces are adjacent or in contact. The position of the surfaces is located from a spatial relationship to some principal coordinate frame attached to the basic component. The knowledge base is expressed as mobility (M) and contact (C) functions for each pair of parts.
A syntax for the representation of C and M functions has been developed based on six degrees of freedom. We assume there is a body coordinate system towards six directions fixed to each of \( n \) components. Contact and mobility of one component with respect to another one are then evaluated for each of \( n \) components. This leads to a combination of 2 into \( n \), i.e. \( n(n - 1)/2 \).

The contact between component \( b \) and component \( a \) is a function \( C(a, b) = (V_1, V_2, V_3, V_4, V_5, V_6) \) and the relative mobility of component \( b \) vis-à-vis component \( a \) is a function \( M(a, b) = (V_1, V_2, V_3, V_4, V_5, V_6) \) where \( V_1 = X, V_2 = Y, V_3 = Z \) and \( V_4, V_5, V_6 \) are their respective complement direction. We take into account the part union angle \( \theta \) and the relative mobility in terms of translation and rotation \( \phi \). For example, in Fig. 7 component \( b \) has a union angle of 20° with respect to \( V_2 \) (or \( Y \)) in terms of the contact function leading to a rotation of 20° with respect to \( V_3 \) (or \( Z \)) in terms of mobility function. The final expression is as follows:

\[
C(a, b) = (V_1, V_2 : 20, V_3, V_4, V_5, V_6) \quad \text{and} \quad M(a, b) = (V_1, V_2, V_3 : -20, V_4, V_5, V_6)
\]

The next step is to encode \( V_1, V_2, \ldots, V_6 \); we assigned a code to each \( V \) to match each degree of freedom (Table 1) in order to simplify the computer-based processing. A part mating along an assembly axis having \( \theta_i = 0 \) and \( \phi_i = 0 \) is defined as a linear assembly trajectory; an attachment by means of screw is defined as a 'circular assembly trajectory'; and an assembly system having \( \theta_i \neq 0; \phi_i \neq 0 \) is called a 'nonorthogonal assembly trajectory'. This latter is shown in Fig. 7: Table 1 presents the related knowledge base.

### Table 1. Knowledge representation of assembly prototype in Fig. 7

<table>
<thead>
<tr>
<th>C (Contact)</th>
<th>M (Mobility)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C(a, b) = (2.1:20,2.2,0.2) )</td>
<td>( M(a, b) = (0.40:-20.0,0.0) )</td>
</tr>
<tr>
<td>( C(a, c) = (2.1,0.2,0.0) )</td>
<td>( M(a, c) = (0,1,1,0,0,1) )</td>
</tr>
<tr>
<td>( C(b, c) = (0.0,0.0,0,0) )</td>
<td>( M(b, c) = (1,1,1,0,1,1) )</td>
</tr>
<tr>
<td>( \theta_1 = 0; \theta_2 = 20°; \theta_3 = 0 )</td>
<td>( \phi_1 = 0; \phi_2 = 0; \phi_3 = -20° )</td>
</tr>
</tbody>
</table>

**Rule 1**

The assembling function \( A \) is derived from the mobility function \( M \):

\[ A(b, a) = M(A, b) \]

**Rule 2**

IF \( A(b, a) \) is possible,

THEN \( A((b, a), c) = A(a, c) * A(b, a) \)

**Rule 3**

\[ A(a, b), (d, c) = A(b, a) * A(d, c) \]

Moreover, we add a few properties introducing the inverse functions:

\[ M(a, b) = M^{-1}(b, a) \]
\[ C(a, b) = C^{-1}(b, a) \]

**Rule 4**

IF \( M(a, b) = (V_1, V_2, V_3, V_4, V_5, V_6) \),

THEN \( M^{-1}(a, b) = (V_4, V_5, V_6, V_1, V_2, V_3) \)

IF \( C(a, b) = (Y_1, Y_2, Y_3, Y_4, Y_5, Y_6) \),

THEN \( C^{-1}(a, b) = (Y_4, Y_5, Y_6, Y_1, Y_2, Y_3) \)

The method of data-driven or forward chaining taking account of these properties is very effective in solving the sequence generation. The missing functions 'M' and 'C' are thus generated by the inference mechanism itself, so the strategy consists of first generating all the permutations and, second, testing
that a system planning should not have to analyze an assembly but rather should use the tree sequence structure to issue level command actions. Usually assembly generating involves a high degree of implied information from the shapes of and the relationship between the surfaces and features to be manipulated and assembled.

7 TEST CASES AND RESULTS

The implementation of this integrated system is structured as a number of layered modules because none of the existing geometric modelers or CAD systems at present provide specific data such as geometric tolerances, feature recognition, form-feature dependencies or adjacencies, relationships between component characteristics. Consequently, many researchers \(^{17,18}\) have defined their own specific data representation to solve the lack of formal protocols between CAD and manufacturing applications. Surface recognition and feature recognition are the key issues of our first layer; the fundamental approach here is object-oriented modeling in conjunction with a set of schemes such as feature-based representation and abstract data type. The
Assembly sequences in CAD/CAM

latter successfully demonstrated the efficiency of the two major concepts, namely the abstraction of generalization and the abstraction of aggregation. The first step of computation processing allows us to display a model-based surface recognition (Fig. 9) for identifying flatness, protrusions and depressions. The second step displays a feature-based representation (Fig. 10) with associated edges and vertex coordinates from spatial relationships to a principal coordinate frame. Surface recognition and feature recognition are of type 3 extrinsic characteristics and cannot be extracted automatically from the CAD database. A recognition procedure was considered in section 4.3. Such needs dictate the implementation of PDES/STEP but, in spite of their existence, the development of a CAD interface related to ‘recognition’ remains essential for automated planning activities. In assembly planning, automated feature recognition provides a better understanding of objects in a given coordinate frame to determine which features of a component mate on another. A series of components with various complexities have been tested to check the correctness and completeness of our methodology. The next layer looks at the functions called C (contact) and M (relative mobility) which represent the knowledge formulation that defines mating conditions and spatial relationships between components. In the last layer, the embedded inference mechanism was shown to be an effective way to accomplish the objective. Figures 11 and 12 illustrate the feasibility of an assembly sequence generation. This
8 CONCLUSION

This paper discusses the fundamental problem arising from the lack of a conceptual framework for integrating product design in product manufacturing. Parts are designed using the Pro/Engineer CAD system. In this context, we have developed a prototype object-oriented database system to translate the design information from the IGES and NEUTRAL files and to support the management of data. Like a few other databases, SISDES also contains some knowledge of the semantics of the data. This information essentially consists of the existence or absence of contact and mobility of one component with respect to another. Using this information, all the feasible assembly sequences can be generated as a tree sequence structure and then evaluated, the most appropriate one being chosen to carry out the equipment layout. This integrated environment has been achieved by associating the solid modeling system with an object-oriented DB (SISDES) and an expert system (XGEN) to prove the validity of the generation process in mechanical assembly sequences. The implementation of the knowledge base as a central component of the integrated system bridges the gap between CAD and CAM. This new approach utilizes some aspects of AI for a manufacturing application of cognitive science.

ACKNOWLEDGMENTS

Part of this research was supported by Northern Telecom, Outside Plant Division, Manufacturing Department, Montreal, Canada. We would like to thank our colleague Dr. Otto Ikome for revising the manuscript.

REFERENCES


