AUTOMATED INSPECTION PLANNING WITHIN THE RAPID DESIGN SYSTEM

Francis L. Merat†#, Gerald M. Radack*#, Kavous Roumina*#, and Steve Ruegsegger†#

*Department of Computer Engineering and Science
†Department of Electrical Engineering and Applied Physics
#Center for Automation and Intelligent Systems Research
Case Western Reserve University

ABSTRACT

This paper describes a methodology for automated inspection planning of machined parts within a feature-based CIM in which part geometry and tolerances are represented as "features." This representation of information as features is extended to inspection process planning where "inspection plan fragments" are inspection features containing specific information about how toleranced geometry is to be inspected. Tolerances can be either coordinate tolerances or geometric tolerances. A rule base of methods and detailed procedures for evaluating tolerances based upon industrial practices is used to generate the inspection plan fragments. A single tolerance can often be inspected in multiple ways resulting in the generation of many inspection plan fragments. Inspection planning for computer controlled coordinate measuring machines (CMMs) will be emphasized in this paper. The overall inspection process planning consists of generating all possible inspection plan fragments for each tolerance in the design and combining the inspection plan fragments into an overall time-efficient inspection plan. Special considerations which are important in inspection planning for CMMs such as the generation of collision free inspection probe paths will be briefly described. An algorithm for inspection process planning will be described and applied to a sample part. Once an inspection process plan is generated it can be translated into executable code for a computer controlled CMM.

1. INTRODUCTION

Problem Statement

The goal of this work is the development of an automated inspection planner for a CIM system. The inspection planner takes the part model—consisting of a specification of the part geometry as well as dimensioning and tolerancing information—and produces an inspection plan, which gives detailed instructions as to how to inspect a manufactured part to determine whether it is within tolerance. An inspection plan might consist of printed instructions to be manually performed by an inspector, or it might consist of code that can be executed by automated inspection equipment such as a computer controlled coordinate measuring machine (CMM). Ideally, the process plan will consist of a combination of both manual and automatic steps, so that each aspect of a part is inspected in the optimal way in order to achieve the minimum time inspection plan.

Motivation

It can be quite time-consuming for an inspector to understand a drawing and determine how to inspect a part. Programming CMMs is also time-consuming and tedious. This overhead is particularly critical when it is spread over a small number of parts. An automated inspection system can reduce the time between part design and final inspection, cutting costs and allowing better response to market demands.

By generating the inspection plan when the part is being designed, the system can aid the designer to specify parts that do not require unnecessary inspection procedures as well as eliminating any confusion between the designer and inspector over part inspection requirements.

The Rapid Design System

This work is part of a larger effort to develop a Rapid Design System (RDS) [1], the objective of which is to reduce the time from design to manufacture and inspection. The RDS is being developed with the cooperation of a design and manufacturing organization which specializes in custom modification of aircraft and production of replacement parts which are not available from the manufacturer. For this type of application, turnaround time is more important than minimizing total machine time for manufacture and inspection.

2. BACKGROUND

Geometric Dimensioning and Tolerancing (GD&T)

The inspection process is driven by tolerances specified by the designer. Tolerances are modeled on the ANSI Y14.5 standard [2] and can be of two types: coordinate (also known as plus/minus [±] or two point) tolerances and geometric tolerances.

A coordinate tolerance, such as the diametral tolerance on the hole in Figure 1, requires that two point measurements be taken and the distance between the points be evaluated. These are usually shown on an engineering drawing as a dimension with an associated ± tolerance. If this distance is within the specified variation

This research was supported in part by U.S. Air Force Contract No. F33615-87-5250 through Wright Laboratory, Wright-Patterson Air Force Base, Dayton, Ohio.

that tolerance is satisfied. Coordinate tolerances are used to specify relationships between two surfaces or lines and can be measured in any pose (position and orientation).

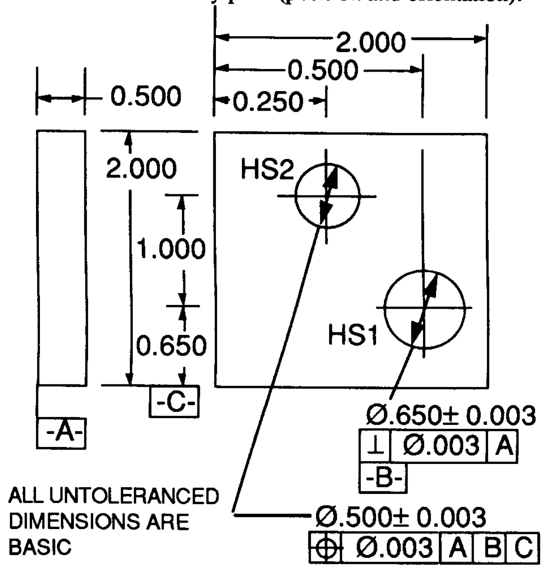

Figure 1 - Partially Dimensioned Part

Geometric tolerances such as the position tolerance on hole HS2 in Figure 1 are more complex and can be used to specify allowable variations in the pose of a geometric feature or can be used to specify allowable variation in the intrinsic properties of a feature, e.g., the flatness of a surface. Geometric tolerances such as those of position and angularity must be defined in datum reference frames. These are coordinate systems whose origin and orientation is uniquely specified by three datums. Datums can be either surfaces or curves (e.g., the axes of holes or shafts). Points can also be used as datums but will not be considered here.

A datum reference frame specifies the part coordinate system to be used for both measurement and evaluation of the part. For example, the datums labeled A, B, and C in Figure 1 define the datum reference frame indicated as |A|B|C| , which must be measured prior to verifying the hole position tolerance.

Coordinate Measuring Machines

A coordinate measuring machine (CMM) is typically a gantry-type robot with three orthogonal degrees of freedom. The "arm" of the CMM is equipped with a touch probe to make point measurements of the surface to be inspected. In general, the motion of the CMM is characterized by two distinct commands: The "MOVE" command is used to move the probe at a high speed between measurements. It is important that this movement be collision-free. The "MEAS" command is used to acquire measurement data and moves the probe very slowly in the specified direction until the probe tip contacts the part. Simple touch probes are fixed in orientation and may consist of nothing more than a precision rod with a spherical tip. More sophisticated probes are similar to a robot wrist with two rotational degrees of freedom (see Figure 3).

Inspection

Inspection consists of performing manual and/or automatic operations to evaluate the specified tolerances. The complete sequence of such operations, when executed, will result in the statement that a part meets or fails to meet all specified tolerances. Additional information such as which tolerances were not met, machining variation for process control, and the like, may also be provided.

Datum measurements and evaluations must be done before any geometric tolerances (except those of form) specified in a datum reference frame can be evaluated.

ANSI Y14.5 inspection techniques were developed for "hard" gages using surface plates, fences, plugs and the like. Inspection procedures using CMMs can only sample the surface at a finite number of points and evaluate the part geometry based upon those measurements. This process has been called "soft gaging" and can result in two basic problems in CMM inspection: one, where to sample the surfaces; and, two, how to interpret the sample measurements. An example of the second problem is incorrect results from commercial CMM inspection software [3]. Work is in progress to reconcile "hard" and "soft" inspection techniques [4]. We have used accepted industrial practices for the inspection and evaluation of tolerances in designs produced with the RDS. Many CMM inspection techniques are actually hybrid procedures requiring CMM measurements from manually placed fences or plugs. See, for example [6].

3. REPRESENTATION

The RDS is intended to support "feature-based design," in which parts are described in terms of "features" [1]. Features can be other than purely geometric, with each discipline such as inspection or manufacturing having its own set of features.

Form Features

Form features determine the part geometry. To the designer "form" features such as slots, ribs, bosses, through holes, blind holes and pockets correspond to specific geometric configurations on the surface of the part. These features can be broadly classified as "negative" and "positive" features which, respectively, represent the removal of material from and addition of material to the design. Feature-based designs such as in the RDS are composed of positive features such as blocks; however, they also contain negative features such as holes and slots that physically exist only when attached to positive objects such as blocks.

GD&T Features

Tolerances and datum specifications according to the ANSI Y14.5 standard form the basis of GD&T features.

Coordinate tolerances can be specifically defined as relationships between surfaces. More details of this representation can be found in [2,7].

Inspection Features

The following inspection features represent primitive operations which must be done during inspection:

- manual operation: Place the part in a jig, insert a plug in a hole, etc.
- measurement: Sample a point with a CMM, measure a distance with a caliper, etc.
- evaluation: Apply some operation to one or more data (typically measurements), resulting in some numeric or geometric data.
- comparison: Compare some data (typically the result of an evaluation), producing a binary result.

The inspection planner must combine these operation to make the process plan.

Planning Features

Each *instance* of a GD&T feature class is linked to a feature called an Inspection Plan Fragment (IPF) [8]. Figure 2 shows the relationships between the IPFs and the GD&T features for the part of Figure 1. The perpendicularity and positional tolerance of HS2 are linked to the GD&T datum reference frame ABC through their specification. This relationship is shown as horizontal links in Figure 2.

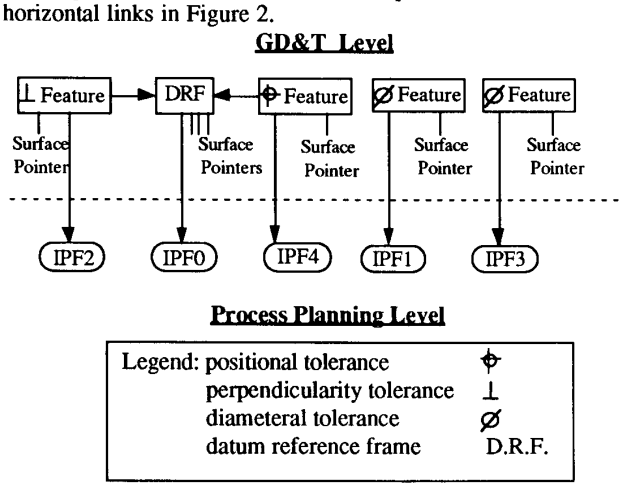

Figure 2 - Relationship between GD&T and Process Planning

Each IPF corresponds to a possible valid inspection procedure for the associated tolerance, and specifies the coordinates of the points to be sampled and the orientation of the CMM probe. For example, the diameter of a hole is inspected by inserting the CMM probe into the hole along the hole axis and measuring at least three sample points.

An IPF is expanded from a macro—called an IPF generator (IPFG)—which is contained in the tolerance *class* definition. The IPFs generated are valid when they intersect geometrically in such a way that not all feature surfaces are present in their entirety. The inspection process planner detects such feature interactions and modifies the IPFs as appropriate.

For each tolerance, separate IPFs are generated for different CMM probes, different probe orientations (we restrict ourselves to two orientations, see Figure 3), and inspection tools other than CMMs such as depth micrometers and plug gages. Only one IPF need be executed for each tolerance. Manual techniques such as inserting plugs into holes for CMM measurements are also included in this IPF expansion. Basic information about how each feature surface can be measured (usually restrictions upon the probe orientation with respect to the surface) is contained in the IPF.

An IPF logically contains the following objects, each of which corresponds to one of the inspection features listed above:

- measurement request: a link to the surface of a feature to be measured, the number of points to be measured, and any constraints on those points, e.g., that they be non-collinear.
- evaluation request: a high-level representation of an evaluation to be performed.
- comparison request: a high-level representation of a comparison to be performed.

The planner translates the "request" objects listed above into more concrete "specification" objects. For example, a measurement specification contains a link to a surface in the B-rep of the part, and (ideal) coordinates of points on the surface where the measurements are to be taken.

4. PROCESS PLANNING ALGORITHM

The steps for generating a part inspection plan are:

Step 1 Generate all IPFs

By calling the IPFGs for each tolerance feature, all IPFs will be generated.

Step 2 Group measurement requests by part orientation

All IPFs which can be inspected with the same part orientation are grouped together for inspection.

Step 3 Select an IPF to measure each tolerance feature

An optimization procedure is used to select one IPF per tolerance feature, in order to minimize total inspection time. The objective function for the optimization takes into account the time required to perform the measurement task (and to enter data into the computer if it is a manual task) as well as the time required to set up for the task, e.g., probe changes or rejigging of the part. Computer processing time to perform the evaluation is assumed to be negligible, and is ignored. An attempt is made to use as few part orientations as possible for measurement, since each orientation requires moving or rejigging of the part, which in turn requires additional measurements to establish the part orientation in the CMM's coordinate system.

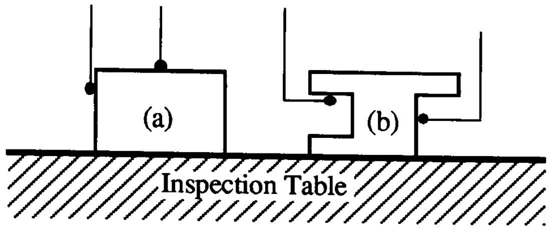

Figure 3 (a) CMM in vertical probe configuration; (b) CMM in horizontal probe configuration

Step 4 Eliminate redundant measurement requests

The complete set of measurement requests is processed to eliminate redundancies. IPFs for different tolerances can generate measurement requests for the same surface. In general, the same set of measurement points may be used to evaluate different tolerances such as position and flatness simultaneously as long as the constraints for all measurement requests are satisfied. In the example of simultaneous measurement requests for position and diameter of a hole, both measurement requests are for three measurement points in a single plane and can be satisfied by the same measurement. Since evaluation and comparison features represent executable code, it would be possible in principle to optimize them as well, eliminating redundant operations. This is not addressed at present.

Step 5 Create "specification" objects from "request" objects

For example, to create a measurement specification from a measurement request, select coordinates on points on the corresponding surface that meet the constraints.

Step 6 Sequence the operations

A data flow graph containing the measurement, evaluation, and comparison specifications is analyzed using the topological sort algorithm, and a valid sequence for execution of the inspection plan elements is generated.

Step 7 CMM probe path planning

CMM commands to move the probe to the sample points specified by the measurement requests without the probe and/or CMM colliding with the part are generated. The path traversed by the probe should be time-efficient. We use a configuration space transform to determine collision-free regions for CMM movement [14] and use a minimum path algorithm for 3-D polyhedral objects to determine a minimum inspection time path [15] in configuration space.

In general, computing configuration-space transformations is inefficient; however, we use a feature-based technique to improve the computational speed of the technique. This work is based upon Branicky [16], who showed that the configuration space transform of an

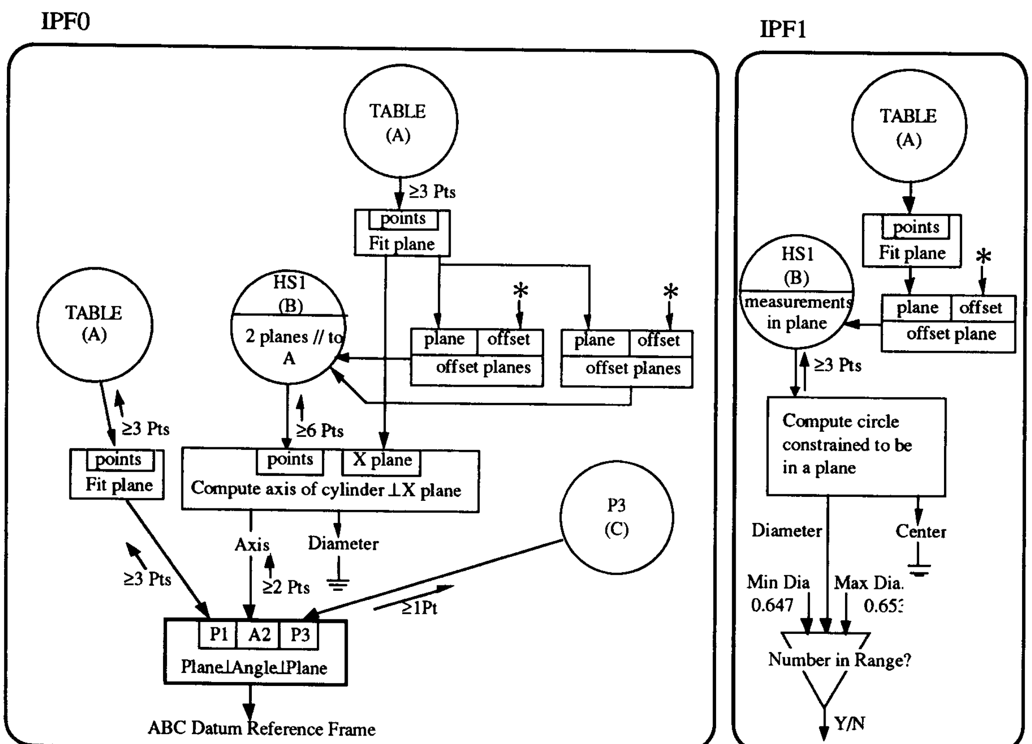

Figure 4 - IPFs for Inspecting Figure 1 Part (Partial set)

object composed of positive features can be computed as the union of the configuration space transforms of the individual "positive" obstacles such as blocks. Jeon [9] has extended the work of Branicky to negative features. The configuration space transforms of our features are precomputed and, for any given design, combined appropriately to quickly compute the configuration space transform of a given design.

Step 8 CMM code generation

Executable code for a computer controlled CMM is generated at this step in a variant of the DMIS [11] CMM programming language.

5. EXAMPLE

The process planning procedure of Section 4 has been applied to the part shown in Figure 1. Due to space considrations, only two of the IPFs generated for the design in Figure 1 are shown in Figure 4. Measurement requests are shown as circles with any constraints shown in the lower half. Rectangles indicate evaluation requests; triangles indicate comparison requests. The flow of data is, in general, from the measurement requests to the evaluation requests to the comparison requests as shown by the arrows. Short arrows in the direction opposite to the data flow show constraints on the input to an evaluation request. These constraints have an obvious effect upon the upstream requests. Data flow from an "*" refers to data from the part geometry and precomputed by the IPFG's when the IPFs are created.

IPF0 is an IPF for the datum reference frame ABC (of type plane-axis-plane). This generates an evaluation request for a primary datum plane determined by at least three non-collinear inspection points as shown by the arrow with the legend ≥3 pts. The evaluation request for A, in turn, generates a measurement request for three points on the CMM inspection table on which the part is assumed to rest. An evaluation request for at least two points on the secondary axis datum B perpendicular to the primary datum surface A is generated. This evaluation, in turn, produces a measurement request for six inspection points, three per plane, in two planes parallel to A. The offset requirement produces an evaluation request for two offset planes parallel to A which, in turn, produces an evaluation request and a measurement request for plane A. An evaluation request for a tertiary datum plane P3 perpendicular to A and B and determined by a minimum of one point is generated. No explicit evaluation request is generated for this requirement since no computations are necessary.

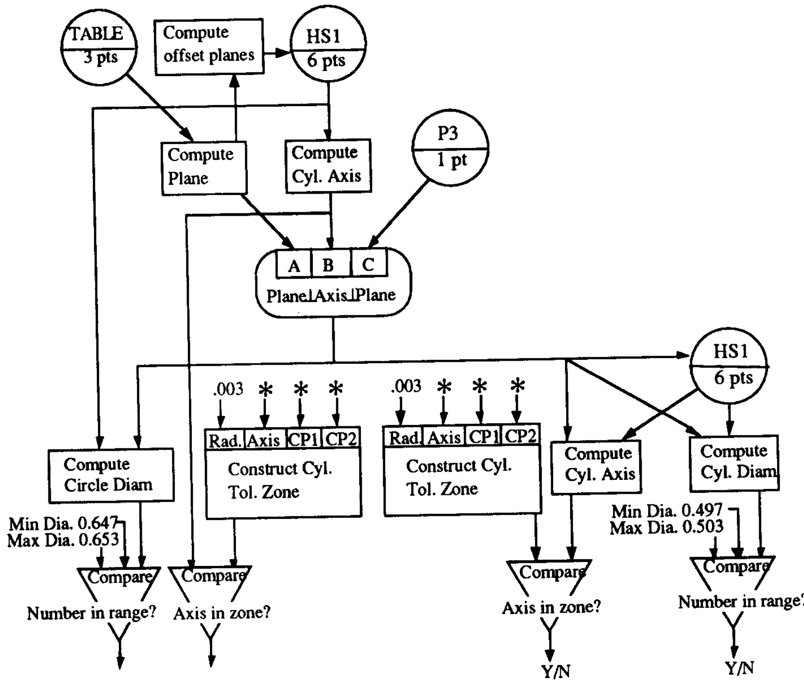

Figure 5 - Process Plan for Inspecting Figure 1 Part.
("*" indicates input from part geometry. Not all process plan details shown.)

IPF1 is an IPF for the diameteral tolerance on HS1. This IPF begins with a comparison request for the diameter of HS1. The maximum and minimum diameters shown come from the GD&T feature associated with the tolerance. This comparison request generates an evaluation request for a diameter computed from a circle in a plane perpendicular to the nominal axis of the cylinder. This evaluation requires at least three sample points as shown and, in turn, requires that a plane offset from A be evaluated and measured.

Only two IPFs are shown because of the complexity of each IPF. Other IPFs are also generated. The diameteral coordinate tolerance requires three coplanar points for evaluation. The perpendicularity and positional tolerances require six non-coplanar points for evaluation. The diameteral tolerances can be evaluated from the positional and perpendicularity measurement requests so the diameteral measurement requests were eliminated as redundant resulting in a single measurement request for each hole. No feature interactions occur.

Elimination of multiple requests results in the process plan shown in Figure 5. Multiple requests for HS1, HS2, and TABLE were combined into single measurement requests. Note that the evaluation and comparison requests for diameteral tolerance remain. TABLE is a measurement request for the table the part rests on (datum A). HS1 is a measurement request for datum B, the axis of hole HS1, and a perpendicularity tolerance on the axis of the hole HS1. P3 is a measurement request for the datum plane C. The inspection point information from HS1 is evaluated to determine it's axis. Plane⊥Axis⊥Plane is a specialized evaluation request to determine the ABC datum reference frame. The diameteral (ø) tolerance on HS1 is evaluated and compared. The perpendicularity tolerance on the axis of HS1 requires that the corresponding tolerance zone (soft gage) be constructed. This type of evaluation request is shown as the large rectangle with explicit inputs. In this case, the inputs are the radius of the tolerance zone (Rad), the nominal axis of the tolerance zone (Axis), and the position of the tolerance zone (CP1 and CP2). The latter three parameters are determined from the part geometry as specified in the design. When evaluated, this tolerance zone and the evaluated axis of HS1 are compared to see if the perpendicularity tolerance is satisfied. The process of measuring, evaluating and comparing the diameteral and positional tolerances on HS2 is almost identical to that for HS1.

The IPF specifies the allowed probe orientations for measurement requests of the geometric and coordinate tolerances shown in Figure 1. The four tolerances shown all correspond to holes with the CMM probe orientation along the hole axis (Figure 3(a)).

The particular CMM we are using in our work uses a DMIS-like programming language called CMES [12]. The command sequence corresponding to the process plan is capable of being automatically generated from the RDS part description although this has not been implemented yet. CMES code corresponding to IPF0 is shown in Figure 6. The coordinates xn,yn,zn define point Pn. The number of points is only unique to each measurement request. For example, points P1, P2 and P3 are defined for the datum A measurement request; these are not the same points P1 and P2 defined in the datum B measurement request, etc. The distance D is the perpendicular distance from the surface at which the probe begins to execute a MEAS command. The CMES command #MC corresponds to a MEAS command; #PT corresponds to a MEAS command. Details of the command language are found in [6].

```
!COMMENTS                      !CMES CODE
!!SETUP DRF                    \RE\Y
!restart                       \EA\NP\
!move above table              \#MC,RS,Z\z+D
!recall PH9 position           \UR,1,PH9\
!!Measure Table (datum A)
!move to point 1               \#MC,RS\x1\y1\z1+D
!measure point 1               \#PT,RS,Z\z1
!save point 1                  \SP,1
!move to point 2               \#MC,RS\x2\y2\z2+D
!measure point 2               \#PT,RS,Z\z2
!save point 2                  \SP,2
!move to point 3               \#MC,RS\x3\y3\z3+D
!measure point 3               \#PT,RS,Z\z3
!save point 3                  \SP,3
!evaluate z plane (A)          \UP,1,2,3\AX-,Z
!!meas. plane C (datum C)
!move to point 1               \#MC,RS\x1+D\y1\z1
!measure point 1               \#PT,RS,X\x1
!save point 1                  \SP,1
!move to point 2               \#MC,RS\x2+D\y2\z2
!measure point 2               \#PT,RS,X\x2
!save point 2                  \SP,2
!evaluate x plane (C)          \UP,1,2\N1-,X,Z
!measure hole (datum B)
!move above hole               \#MC,RS\x1+D\y1\z1+D
!auto inspect hole             \#ID,Z\x1\y1\z1\d1
!save measurements             \SP,3,4,5,6
!evaluate hole axis            \UP,6\N2,Y,X,Z
!create axis system
!get intersection              \PI
!make master datum             \MD
!save axis                     \SA,1
!return to previous level      \ET
```

Figure 6 CMES program fragment to measure Datum Reference Frame |A|B|C|

6. DISCUSSION AND CONCLUSIONS

The algorithm described in Section 4 has been used to manually generate inspection plans and an implementation of the automated process planner within the RDS is in progress. Major limitations of the present work relate to how feature interactions are handled. For example, how is a measurement specification determined when a surface of a geometric feature is transformed into several "separate" surfaces by feature interactions.

The CMM path planner has been implemented for a simple set of features (blocks, slots and holes) on a personal computer in C. It works well for the simple geometries illustrated in Figure 4 where the features do not interact and are parallel to the block surfaces. We are currently investigating whether our described feature-based technique can be used to efficiently generate configuration space transforms when the features are not parallel to the block surfaces. We are also investigating how to extend the work to the other features used to design parts in the RDS.

An optimization algorithm based upon heuristic search techniques [13] that can integrate steps 2 thru 6 of the process planning algorithm is being investigated.

The system described is capable of handling simple parts such as that shown in Figure 1; however, many of the rules described above for inspection process planning are heuristics that have been developed working with inspectors during the implementation process. The inspection planning subsystem of the RDS is being tested in a QC/QA shop and will be reported at a future date.

REFERENCES

1. LeClair, Steve, "Rapid Design System: Feature-Based Design," Proceedings of this conference.
2. Dimensioning and Tolerancing, ANSI Y14.5M-1982, ASME, 345 East 47th Street, New York 10017.
3. Government-Industry Data Exchange Program (GIDEP), Alert No. X1-A-88-01, 22 August 1988.
4. Galm, James, Ph.D. dissertation, "Functional Surface Estimation of Machined Surfaces," Case Western Reserve University, January 1991.
5. ANSI Y14.5.1 Mathematization Committee.
6. "Dimensional Metrology and Geometric Conformance" in The Tool and Manufacturing Engineers Handbook, Edited by Charles Wick and Raymond Veilleux, McGraw-Hill.

7. Jacobsohn, J., Radack, G. and Merat, F., "Integrating Knowledge of Geometric Dimensioning and Tolerancing into a Feature-Based Design System," *Proceedings of Rensselaer's Second International Conference on CIM*, Troy NY, May 1990, 151-159.
8. Radack, Gerald and F.L. Merat, "The Integration of Inspection into the CIM environment," *Proc. of Hawaiian International Conference on System Science*, January 1990.
9. Jeon, O-Keon, "Efficient Inspection Path Planning Algorithm," M.S. Thesis, Case Western Reserve University, Cleveland, Ohio, August 1990.
10. Ferendeci, Derya, "The Inspection of Prismatic Parts," M.S. Thesis, Case Western Reserve University, Cleveland, Ohio, May 1991.
11. DMIS 2.0 Specification, ASME, 345 East 47th Street, New York 10017.
12. CMES Programming Manual, LK Instruments.
13. Kusiak, Andrew, Intelligent Manufacturing Systems,Prentice-Hall, 1990, p.402-409.
14. Lozano-Perez,T., "Spatial Planning: A Configuration Space Approach," IEEE Transactions on Computers, C-32(2), February 1983, p.108-120.
15. Lozano-Perez, T. and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths among Polyhedral Objects," Communications of the ACM, Vol. 22, No.10, 1979, p.165-175.
16. Branicky, M.S., "Efficient Configuration Space Transforms for Real-Time Robotic Reflexes," M.S. Thesis, Case Western Reserve University, January 1990.