Third International Symposium on Robotics and Manufacturing, Vancouver, B.C., July 18-20, 1990.

Automatic Inspection Planning Within a Feature-Based CAD System

Francis L. Merat[*#]
Gerald M. Radack[+#]
Jeremy F. Jacobsohn[*#]
[*]Department of Electrical Engineering and Applied Physics
[+]Department of Computer Engineering and Science
[#]Center for Automation and Intelligent Systems Research
Case Western Reserve University
Cleveland, Ohio 44106

ABSTRACT

This paper describes a CIM system which incorporates automatic generation of inspection plans for coordinate measuring machines (CMMs). Design is done in terms of form features. Associated with each feature are inspection code fragments which are instaces of inspection procedures. There are three basic types of code fragments : (1) fragments for checking the internal dimensions of a feature, (2) fragments for checking the relationships between features, and (3) modified fragments for use when the feature to be inspected intersects another feature. Using this approach, a list of suitable inspection points and approach vectors is generated. The system is consistent with the accepted ANSI Y14.5M standard for dimensioning and tolerancing as well as with observed industrial practice.

INTRODUCTION

The goal of this work is to develop an intelligent inspection planning subsystem for Computer Integrated Manufacturing (CIM) system for mechanical parts. An 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 a Quality Assurance person, or it might consist of codes that can be executed by automated inspection equipment such as a coordinate measuring machine (CMM) or a vision system. Ideally, it would 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.

Our system is intended to support “feature-based design.” In a feature-based design system, a part is described in terms of “features.” These features represent higher-level concepts than the geometric primitives used in traditional systems. One class of features includes slots, ribs, bosses, through holes, blind holes and pockets. These are called “form features” and correspond to specific geometric configurations on the surface of the part. Another class of features includes chamfers and fillets. These serve to modify the geometry specified by the form features. (For example, chamfers cause a square edge to be rounded.) The term “feature” is also sometimes used for attributes such as surface roughness; however, we avoid that usage here.

One of the novel aspects of our system is the ability to represent dimensioning and tolerancing information within the part model. Our representation for dimensions and tolerances conforms to system called "Geometric Dimensioning and Tolerancing" or GD&T. This system is a U.S. standard (ANSI Y14.5M) and is essentially the same as that specified in ISO standard 1101. Each GD&T datum or callout is represented by a feature within the part model.

The specific problem that we address here is how to use feature information to guide the inspection planning process. Features represent a higher-level description of the part than surfaces or primitive volumes that are traditionally used in CAD systems. In our system, each feature has a set of prepackaged GD&T specifications, as well as a set of "inspection code fragments" which can be used to check whether those specifications are met. However, a mechanism is also provided to allow the designer to override these prepackaged specifications, and work directly at the level of individual GD&T callouts. The inspection planner must integrate this information to produce a complete inspection plan. In this paper, we will assume that all inspection is to be done on a CMM. In a separate paper, we described an approach to producing plans for combined manual and automatic inspection [1].

This work is part of a larger project, sponsored by the U.S. Air Force, to develop a "rapid design system" (RDS). RDS is intended to support the fast and economical design of mechanical parts. RDS is being developed with the cooperation of the 4950th Test Wing, a design and manufacturing organization which specializes in custom modification of aircraft, e.g., to add new instrumentation and producing 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.

The objective of the RDS project is to speed the design process by providing (1) an intelligent CAD interface which enables the designer to get his or her design into the computer faster than current systems allow; (2) integrated tools to check a design for manufacturability and inspectability.

The component of the RDS which comprises the intelligent CAD interface mentioned above is called the "Feature-Based Design Environment" or FBDE. This will be described in a later section.

GEOMETRIC DIMENSIONING AND TOLERANCING

Geometric dimensioning and tolerancing (GD&T) is a term applied to all dimensioning and tolerancing and inspection which adheres to the ANSI Y14.5M-1982 standard [2]. This standard applies to both engineering drawings (how the dimensions and tolerances are to be shown) and to the actual inspection (how is the part to be gaged and evaluated for acceptability). The fundamental concept of geometric dimensioning and tolerancing is that the specified tolerance and measurement procedure should relate to the part's function. For example, GD&T allows the specification of a positional tolerance of a hole such as might be typical of a requirement for parts to mate. In general, GD&T will specify a tolerance for a form (examples of this are the cylindricity of a round hole or the flatness of a surface) or a geometric relationship between features (an example of this is the specification of the position of a hole or the perpendicularity of a surface with respect to another surface). The dimensioning and tolerancing of a simple block with a hole in it is illustrated in FIG. 1.

GD&T is very well adapted to feature based design and inspection plan generation for automated equipment. First, it is inherently compatible with a feature-based part description. A basic dimension (used to position a tolerance zone) is readily generated from the part geometry. Tolerances as defined by GD&T are readily applied to design features such as holes, e.g. the position of a hole, and can be easily added to a feature-based CAD system as separate features. Since GD&T also provides a language for defining how a tolerance is to be measured (inspected) such information can be readily incorporated into the definition of a tolerance feature. The actual instance of an inspection procedure, i.e. how a particular feature tolerance will be measured and evaluated, is separately represented as an inspection feature in our system.

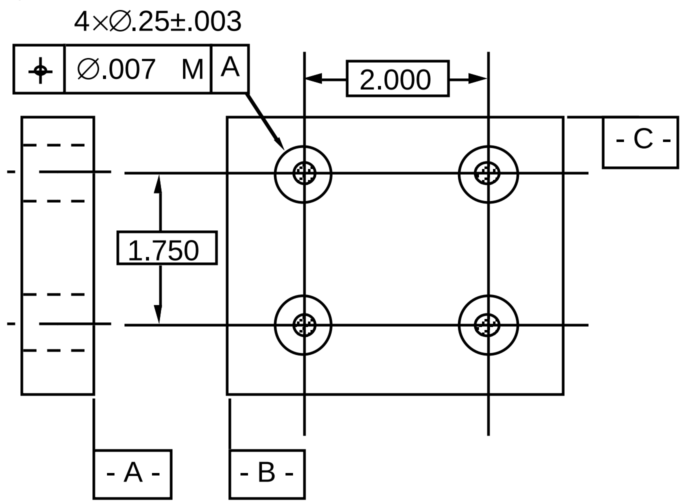

FIG.1 Plate with holes dimensioned and toleranced according to GD&T

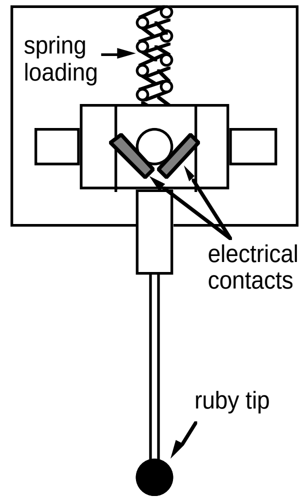

FIG.2. CMM touch probe.

COORDINATE MEASURING MACHINES

A major goal of the RDS is to enable the fast, turnaround of parts in small lot sizes. This implies that the majority of time allocated for inspection would be spent manually inspecting a produced part or writing inspection programs for automated equipment. We have rejected both approaches in favor of automatically generating inspection programs from the geometric and tolerance description of the part. The actual inspection will be performed with a computer-controlled coordinate measuring machine (CMM). Such machines are becoming very popular in industry and essentially are a gantry robot with a (optional) two degree-of-freedom wrist. This mechanical arm can position a touch probe to determine the location of a point on the inspected part's surface. The touch probe closes a pair of electrical contacts whenever the probe tip contacts a surface as shown in FIG.2. Because coordinate measuring machines are precision devices, the probe is constrained to approach a surface to be measured along an "approach direction" corresponding to a calibrated orientation. Typically such calibrations are performed for the probe either parallel or perpendicular to the measurement surface.

The result of the use of a calibrated touch probe produces several constraints upon how such a probe can be used for automated inspection. The first constraint is that the probe can only touch a surface to be inspected at a finite number of inspection points. The ANSI Y14.5M standard was originally defined for manual inspection using hard gages. The reconciliation of point sampling with tolerances specified according to ANSI Y14.5M is not straightforward. As a result we have used accepted industrial practices for the inspection and evaluation of tolerances in designs produced with the RDS.

PREVIOUS WORK

Two aspects of this work have been previously described in the literature. Elmargathy [3] has described a feature-based system for the automated inspection of turned parts. Her work used the ANSI Y14.5 tolerances as applicable to cylindrical parts and represents one of the first published uses of the ANSI Y14.5 tolerance specification as other than a drafting tool. The system was a rule-based system implemented in Prolog and concentrated upon how to access for inspection a free-standing turned part. Because of the nature of turned parts there was no feature interaction as we have found with rectangular prismatic parts.

As described later in this paper the overall inspection plan for a prismatic part can be constructed from the bottom up where each toleranced feature contains information about how it can be inspected. Neglecting the effect of feature interactions, the initial input for part inspection planning can be regarded as a set of goal points for the CMM inspection probe. Additional considerations are that the CMM probe and support mechanism cannot collide with the part anywhere other than a desired inspection point. This is identical to many robot path planning problems. Collision-free paths can be determined by several methods including the classic A* algorithm [4]. It appears that efficient collision-free paths can be readily computed using a feature-based configuration space planner to detect potential collisions and modify the probe movements to avoid such collisions [5]. Neglecting collision detection the planning problem is simply how to move from one inspection point to another with the constraints of traversing the entire set in the minimum time. This is identical to the traveling salesman problem.

THE FEATURE-BASED DESIGN ENVIRONMENT

The Feature-Based Design Environment (FBDE) is used by a designer to create or modify the part model. The part model consists of a collection of programming objects representing the design-with features, as well as others that describe manufacturing and inspection information, and still others that are concerned with constraint checking and constraint satisfaction. Associated with each design feature is a set of alternate schemes for dimensioning and tolerancing the feature, both internally and with respect to other features. The designer picks one of these schemes when he creates an instance of that feature. Each scheme translates to a set of GD&T callouts which can be displayed. If he is unhappy with the choices presented, he may specify different GD&T callouts for the feature by creating instances of “GD&T features.”

PLANNING

The basic strategy of the inspection planner is to generate inspection code fragments (ICFs), which represent the instructions required to inspect individual features, then piece them together to form a complete inspection plan. Inspection planning is done at several levels. With each GD&T scheme within a feature class, a precomputed inspection code fragment generator (ICFG) is stored. The ICFG is a macro which, when called with the parameters of the feature, expands to an ICF which can be used to inspect a particular feature. The ICF is valid when the feature does not interact with other features. We say that a feature interacts with other features if they intersect geometrically in such a way that not all feature surfaces are present in their entirety. When a feature does interact with other features, planning is done on a surface-by-surface basis as follows: Each GD&T scheme has associated with it a set of sample requests and evaluation functions to be applied to the sampled points. A sample request is a request to sample one or more points from a given surface, along with constraints on the points (e.g., that two points must be at least a given distance away from one another). Since there may be several callouts referring to the same surface, it may be possible to use the same physical samples to satisfy several sample requests.Stated formally, the problem is:

> Given a surface S, let $P = \{P_i; 1 \le i \le m\}$, and $Q = \{Q_j; 1 \le j \le n\}$ be sets of variables representing points on S. Let C be a set of constraints on the points in P and let D be a set of constraints on the points in Q. We want to find values for all P_i and Q_j such that (1) all constraints in C and D are satisfied, (2) the set $P \cup Q$ is as small as possible—i.e., as many points of Q are identified with points of P, and vice versa.

This is a very difficult problem computationally. At present it is solved using the following heuristic:

> select values of P_i that satisfy C.
> for j from 1 to n do
>> Find a value for Q_j that satisfies C and D. Use Q_jt= P_i for some i if it satisfies the constraints.

The algorithm backtracks as necessary when it gets "stuck." In the case of individually-specified GD&T callouts, sample requests are also generated and handled in an analogous manner. Finally, these sample requests and evaluation functions are turned into ICFs that can be merged into the overall inspection plan.

<u>Inspection code fragments for checking the internal dimensions of features</u>

Any of several possible inspection code fragments may be generated by the ICFG with the selection being determined by the feature's geometry or manufacturing process. Examples of inspection code fragments for inspection of basic form tolerances are discussed below.

Manufacturing processes also can change how a feature is inspected. Consider the flat surface produced by milling in Figure 4(a). Such a surface has a larger z-axis surface variation in the y-direction than in the x-direction. The surface variation in the x-direction depends upon the exact size of the cutting tool, how many flutes it has, the tool feed rate, the material being cut, etc. The surface variation in the y-direction will also depend upon the feed mechanism and the tool path overlap. The geometrically identical surface shown in (b) is produced by a grinding tool moving in the x-direction. The number of inspection points in the x-direction will be smaller than that for a milled surface due to the greater surface precision of a ground surface; however, the number of inspection points in the y-direction will not change from (a) because of the surface variation produced by the feed mechanism and tool path overlap.

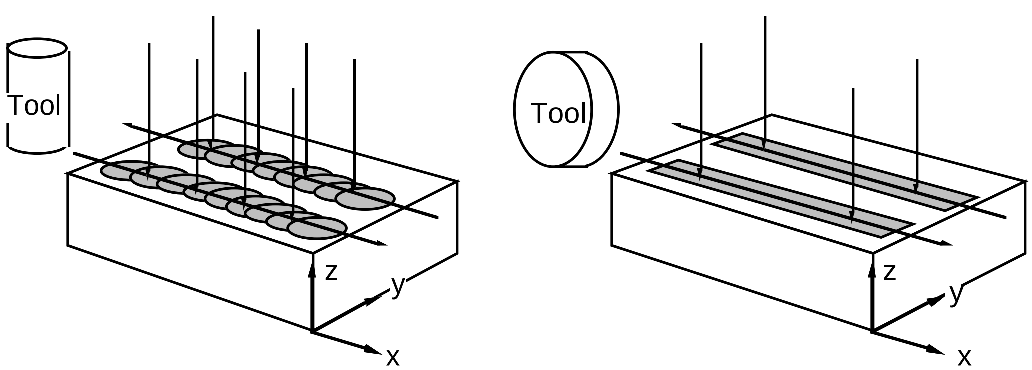

FIG.4 (a) Machining of a flat surface using a vertical mill (b) Machining of a flat surface using a grinder

Inspection code fragments for checking relationships between features

An inspection code fragment for a positional tolerance on a cylinder is shown in FIG.3(a). In practice the number and location of inspection points is a function of the hole "thinness," i.e., a tall and narrow hole might be inspected using two measurement planes with three inspection points each as shown in FIG.3(b).

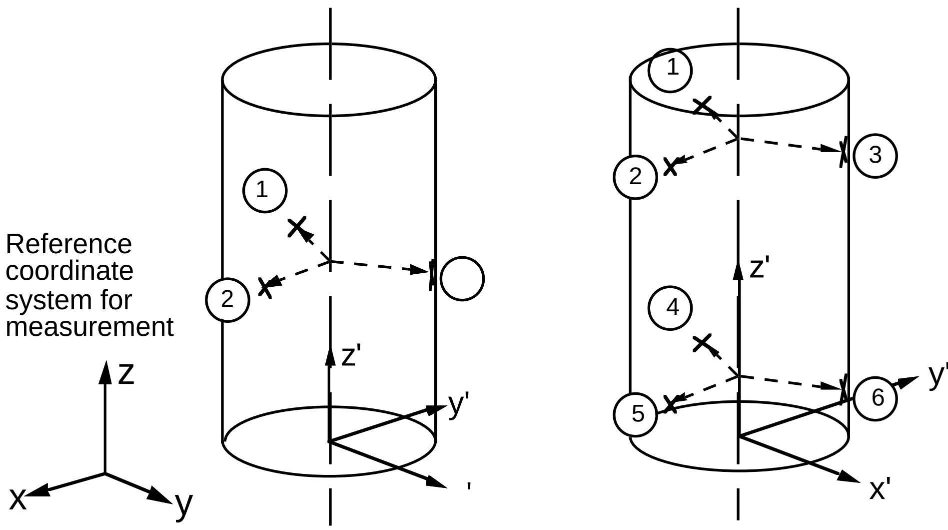

FIG.3.(a) Positional inspection of a hole (b) Positional inspection of a tall, thin hole

Inspection code fragment generation when features intersect

The previous two examples considered only individual features. Consider the pin bracket shown in FIG.5. The details of the part design are not necessary to this example; the hole is specified as a single hole passing through both supports. If the the position of the hole was toleranced, the nominal inspection plan would be similar to that shown in FIG.3(a). However, the hole feature is intersected by a pocket feature (which removed the material between the supports) requiring the inspection code fragment for the hole to be modified from that shown in FIG.3(a). The relevant constraint in this example is that the sample points must be on a physical surface. For the evaluation of a position tolerance it is not necessary that the sample points be on both supports. However, if we had specified a cylindricity tolerance in addition to the position tolerance a second sample request would have been generated. The cylindricity sample request would have required sample points on both supports. The resultant sample set might be like that shown in FIG.5 which can be used to satisfy both sample requests.

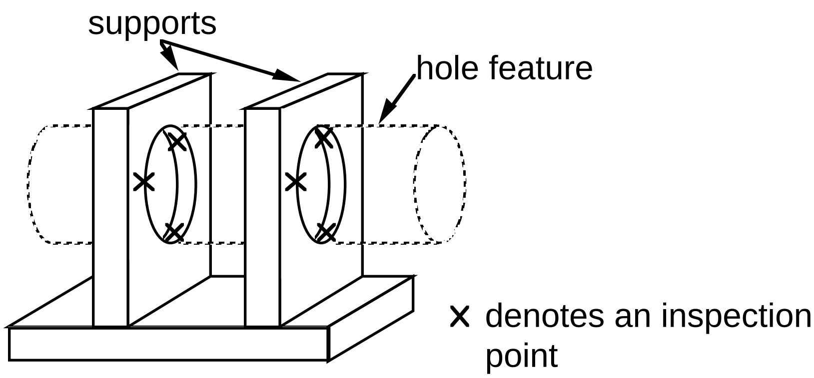

FIG.5 Pin bracket

SUMMARY AND CONCLUSIONS

We have described an approach to intelligent generation of inspection plans for CMM inspection of mechanical parts. Tolerances are specified according to the ANSI Y14.5M format. For features that do not interact with other features, inspection code fragments are generated based on encapsulated knowledge about the features including how they were manufactured. For features that do interact, sample requests are generated for each specified tolerance on the surface to be inspected. The resultant composite sample request will satisfy all requested tolerance evaluations as well as minimize the total number of sample points for that surface. The inspection plan is then generated on a surface-by-surface basis. This approach shows promise for fast generation of reasonably efficient inspection plans suitable for small lot-size manufacturing applications. We are presently pursuing the development of better heuristics for optimization as well as for generation of the overall part inspection plan.

REFERENCES

[1]. Radack, Gerald and F.L. Merat, "The Integration of Inspection into the CIM Environment", Hawaiian International Conference on System Science, January 1990.

[2]. Dimensioning and Tolerancing ANSI Y14.5M - 1982, ASME, 345 East 47th Street, New York, NY 10017.

[3]. H.A. ElMaraghy and H.P.Gu, Expert System for Inspection Planning, Annals of the CIRP 1988, p.85-89.

[4]. Pete E. Hart, Neil J. Nillson and Bertram Rafael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems, Man and Cybernetics, MSSC-4, No.2, July 1968, p.100-107.

[5]. Michael Branicky and Wyatt Newmann, Rapid Computation of Configuration-Space Obstacles, to be presented at the 1990 International Conference on Robotics and Automation