POSITIONING FEATURES WITHIN THE RAPID DESIGN SYSTEM

Gerald M. Radack*#, Jeremy F. Jacobsohn†#, and Francis L. Merat†#

*Department of Computer Engineering and Science
†Department of Electrical Engineering and Applied Physics
#Center for Automation and Intelligent Systems Research
Case Western Reserve University
Cleveland, OH 44106

ABSTRACT

This paper describes a novel method for representing and manipulating dimensioning and positioning information within a feature-based environment for mechanical parts design. In the feature-based design paradigm, designs are constructed using form features which are high level representations of part geometry, *i.e.*, block, hole, slot, pocket, chamfer, fillet, etc. In the system discussed here, two modes of positioning form features are supported. First is a layout mode, which allows the user to position form features relative to each other, in order to lay out the general geometry of the part. The second is a dimensioning mode, in which the user specifies which dimensions on the part are critically important, and what they should be. The system will complete the dimensioning on request, producing enough dimensions to generate an engineering drawing.

1. INTRODUCTION

Feature-based design is a promising paradigm for improving the level of service and integration in geometric design systems. Features regarding the function, form, dimensioning and tolerancing associated with a part geometry enable design at a higher level of description than afforded by engineering drawings or traditional CAD descriptions. A system called the Rapid Design System [4] is using the feature concepts described here as the basis of a vertically integrated knowledge-based manufacturing application.

The form feature set consists of commonly used conceptual entities such as blocks, holes, ribs, pockets, and the like. These form features are instantiated by the designer and attached together as needed to fulfill functional requirements for the desired part such as mounting points, weight reduction, and guideways for sliding motion.

Form features are parameterized by a set of properties including size, shape and position, specified by the designer through graphical gestures or direct input of numerical values. These values, which we term the intrinsic feature parameterization [3], completely specify the geometry of the design within the feature-based CAD system.

The creation and parameterization of the features correspond to the "back of the envelope" sketching process that occurs early in mechanical design. The dimensions are presented informally, using the designer's spatial intuition to visualize part geometry. The intrinsic parameterization is a relatively local description of a part's geometry. Form features are usually positioned relative to one another, or to the form features with which they are most intimately associated, *e.g.*, a hole positioned relative to the block into which it is "drilled."

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.

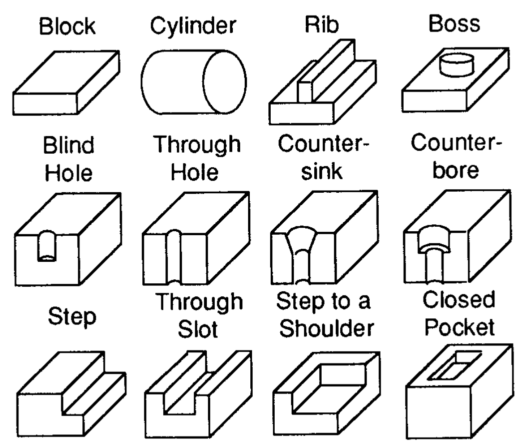

Figure 1. Representative Form Features

2. FEATURES

There are two classes of features used by the designer to specify a part.

Form Features

Form features determine the product geometry. A sampling of the form features currently supported is shown in Figure 1. Features such as ribs, representing the addition of material to the design, are called "positive features"; features such as holes, representing the removal of material from the design, are called "negative features." A form feature representation contains attributes for defining its size and shape (e.g., width of a block, radius of a hole), manufacturing information (e.g., finish requirements), and attributes for positioning and orientating the feature with

CH3051-0/91/0000-0038 $1.00 © 1991 IEEE

respect to other features. The latter will be described further in Section 3.

From the size and shape parameters, a geometric representation of the form feature is constructed using a solid modeler. Using the solid modeler, it is possible to obtain individual surfaces of a form feature, as well as to combine the form features together to form a representation of the designed part geometry.

Dimensioning and Tolerancing Features

The RDS uses features defined in the ANSI Geometric Dimensioning and Tolerancing (GD&T) standard [1] to represent dimensioning and tolerancing information. All GD&T datums and callouts are supported by the system. Callouts such as flatness, cylindricity, profile of a surface, etc., specify the amount by which a manufactured part can differ from the ideal geometry. Since they do not constrain the ideal geometry there will be no further discussion herein of datums and callouts.

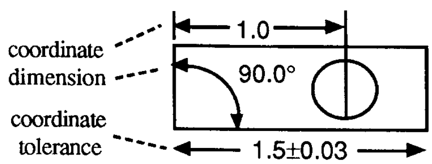

Figure 2. Fragment of Engineering Drawing

On the other hand, there is a well-defined correspondence between coordinate dimensions and tolerances and ideal part geometry (Figure 2). On an engineering drawing, coordinate dimensions (hereafter called "dimensions") are an essential part of the specification of ideal geometry. In the RDS, they are not *necessary* for this purpose, since the intrinsic parameterization can be used. However, it was decided to make dimensions available in the feature set for following reasons:

(1) An engineering drawing must contain dimensions in order to be complete. Thus, it is necessary to generate these dimensions before producing the hardcopy of a design. These dimensions could be generated automatically. However, by specifying them, the designer can customize appearance of the engineering drawing, and can at the same time emphasize functional relationships between features of the part. Figure 3 illustrates two possible ways to dimension a part.

(2) By providing the designer a way of controlling the ideal geometry using dimensions, we can provide an alternate way to specify product shape which may in some cases be more familiar or convenient to the designer.

The representation of a dimension contains the numerical value, and pointers to two surfaces of form features, as well as a flag to tell whether it is "fixed" or "floating." The latter will be discussed in Section 4.

3. THE INTRINSIC PARAMETERIZATION

All features are joined to the surrounding material by constraining one face—which we will call the "top face"—of the feature to lie on a face of some other feature. For negative features this face represents the surface of the material into which the feature is cut, on positive features it represents the surface to which the feature is "glued." The identity of this surface is stored as the "attachment face" attribute of the feature.

Many features have more than one attachment face. For example, a through hole has two: the top of the hole is constrained to lie on the first (or primary) attachment face and the bottom of the hole is constrained to lie on the second attachment face. In all subsequent discussion, we will use "attachment" to mean the primary attachment face unless explicitly stated otherwise.

Our designs begin with a "starting feature." All other features are attached, either directly or indirectly, to the starting feature. For simplicity and ease of discussion, we will assume that the axes of all features are perpendicular and their major axes are also perpendicular to their attachment faces. We call these "orthogonal features."

Thus, once the internal dimensions and the attachment

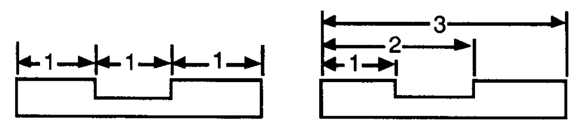

Figure 3. Two Ways to Dimension a Part

faces of a feature are given, the only remaining degree(s) of freedom involve translation in a direction parallel to the first attachment face. These degrees of freedom are fixed by specifying the offset of a designated point—called the "attachment point"—on the top face of the feature with respect to two edges of the attachment face. These edges are called the "offset edges" and may be chosen by the designer. The attachment point is either the center of the top face or one of its vertices.

Attachments are not allowed to be cyclic; that is, if feature A is attached to feature B, either directly or indirectly, then feature B cannot be attached to feature A. Restating this, if we consider a directed graph whose nodes are the features and whose edges represent the attachments, then the graph must be acyclic. To enforce the non-circularity of attachments, the system will ensure that any design changes involving attachments preserve the acyclic property of the feature graph.

4. THE EXTRINSIC PARAMETERIZATION

After defining the geometry using the intrinsic parameterization described in the previous section, the designer may specify dimensions and tolerances.

Every dimension represents a distance or angle between geometric entities in the ideal part geometry. Those entities may be two-dimensional (surfaces), one-dimensional (curves), or zero-dimensional (points). In the RDS, dimension features refer to surfaces, edges or points on the form features that define the part geometry, rather than to the part geometry directly. Since we are assuming here that all features are orthogonal, we can assume without loss of generality that all dimensions refer to faces of rectangular features or axes of circular features, and that they measure distance, not angle.

A dimension is specified by the designer as either "free" or "fixed." A *free* dimension's value is computed from the appropriate intrinsic parameters, and will be updated automatically whenever these parameters are changed, as described in the next section. A *fixed* dimension has a value set by the designer. The system will adjust the intrinsic parameters to make them consistent with the fixed dimension whenever its value is changed.

5. RELATING THE INTRINSIC AND EXTRINSIC PARAMETERIZATIONS

Since the intrinsic and extrinsic parameterizations represent alternate views of the part geometry, there is a set of constraints which must be satisfied in order for the two parameterizations to be consistent. These constraints are determined automatically using a method described below. A constraint manager maintains the consistency of the system as follows:

- If the designer changes the value of an intrinsic parameter, then adjust free dimensions to maintain consistency.
- If the designer changes a dimension, then adjust intrinsic parameters to maintain consistency.

In the case where there may be several ways to change values in order to maintain consistency, rather than making an arbitrary choice, the system reports the inconsistency to the designer, who can then choose one of the following actions:

- Do nothing for the moment.
- Repair the inconsistency manually.
- Choose a way to repair the inconsistency from a set of choices presented by the system.

The designer can turn consistency checking off. This is particularly useful when making a group of changes to the model, since it may be inconvenient to maintain consistency after each step.

The Equivalent Dimensional Path

The relationship of a particular dimension to the intrinsic parameters is determined by stacking up intrinsic parameters into a path that leads from one endpoint of the dimension to the other. This is called the "equivalent dimensional path" (EDP). For example, in Figure 4, the EDP for dimension 1 is:

–offset1-H1 –offset1-P1 + width-starting_block
–offset2-P2 –width-P2 +offset1-H2

In this notation, *a-b* represents attribute *a* of feature *b*.

The equivalent dimensional path for a given dimension is obtained by searching a graph. The nodes of the graph represent *x* or *y* positions of surfaces or axes. There is an edge between nodes *v* and *w* if:

1. *v* and *w* correspond to opposite sides of a rectangular feature. In this case, the edge corresponds to the width, height, or depth of the feature.
2. *v* corresponds to the axis of a circular feature and *w* to the surface. In this case, the edge corresponds to the radius of the feature.
3. *v* corresponds to a surface of a feature attached to *w*. In this case, the edge corresponds to the appropriate offset attribute of the feature containing *w*.

The starting point for the search is one endpoint of the dimension, which is a feature axis or surface. A depth-first search is conducted until the other endpoint of the dimension is reached. The arcs are followed in both directions. Each of these moves connects to another feature axis or face, represented as a node on the search tree.

The constraints are represented by the equation:

$$\mathbf{P}\ \mathbf{E} = \mathbf{B}$$

where **P** is the vector of *m* intrinsic parameter values, **B** is the vector of *n* dimension parameters, **E** is the equivalent path matrix (*n* rows, each representing one dimension feature, and *m* columns, each column representing one intrinsic feature parameter in the feature model) Each e_{ij} is one of {0,1,-1,2,-2}, depending on how feature parameter *j* contributes to the EDP for dimension *i*.

Constraint Satisfaction

We have chosen to investigate one particular iterative constraint satisfaction algorithm, DeltaBlue [2]. This

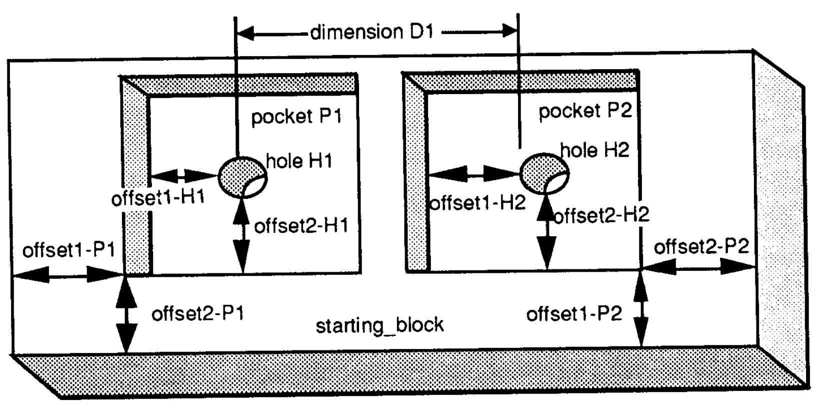

Figure 4. A Part with Intrinsic and One Extrinsic Parameter Shown

algorithm employs user-provided methods to generate new trial values, and allows "optional" constraints, which are ranked hierarchically. New solutions to the constraint network are evaluated on a locally-predicate-better basis. This means that the solver will satisfy as many constraints at each level of ranking (called "strength") before going to the next lower level.

We make use of the constraint hierarchy concept by assigning the indicated strengths to the following constraints:

- *required strength:*
  all constraints relating intrinsic and extrinsic parameters, i.e., the equivalent dimensional path equations represented by the matrix equation **P E = B**;
- *strong strength:* $D_{fixed} = C$
  where D_{fixed} is the value of a fixed dimension (an extrinsic parameter) and C is the value input by the designer;
- *intermediate strength:* $D_{offset} = C$
  where D_{offset} is the value of an offset (an intrinsic parameter) and C is the value input by the designer;
- *low strength:* $D_{float} = C$
  where D_{float} is the value of a floating dimension (an extrinsic parameter) and C is the value input by the designer.

Input to DeltaBlue includes a set of constraints, a set of constrained variables and a current solution. The constraint hierarchy is described to the DeltaBlue solver by an Application Program Interface that allows the client program (the CAD system) to add variables and constraints to DeltaBlue's internal data structure.

Each constraint is provided a method (in the object oriented programming sense of an associated procedure) that produces candidate trial values for the variables that participate in it. The method might provide the exact value that satisfies the equivalent dimensional path equation which is furthest from being satisfied. Since variables may appear in several constraints another constraint will become violated, requiring further iterations. Other methods try fixed incremental or fixed percentage changes to intrinsic parameters. Finally, a more general expert system the above heuristics to choose values, or more specific rules that exploit knowledge of the design features (*e.g.*, choosing hole dimensions from standard increments). Combinations of these methods provide a rich set of possible trial values, increasing the likelihood that a solution can be found.

6. GENERATING DIMENSIONS

Upon request, the system will complete the dimensioning of the part; that is, it will self-generate additional dimensions sufficient to completely describe the ideal geometry of the part. Since we know that the intrinsic parameters completely describe the part geometry, the additional dimensions generated will each correspond to an intrinsic parameter. Any dimensions that are added should be independent of those already existing. We can use the following algorithm to add dimensions along a particular coordinate axis:

```
G ← a graph (V,E) where V = {all surfaces
perpendicular to the axis} and (v,w) ∈ E if there is a
dimension from v to w;
for each intrinsic offset parameter p do
    v,w ← the two surfaces related by p;
    if there is not a path from v to w in G
        create a dimension from v to w;
        add (v,w) to E;
```

This procedure is applied once for each coordinate axis.

7. CONCLUSIONS AND FUTURE WORK

This work adds an additional descriptive tool to feature-based design. It allows part designers more freedom in expressing both the gross geometric layout and the important dimensional relationships of the part. It allows automatic readjustment of the part to satisfy desired dimensional relationships. Finally, dimensional information is presented to the consumers of the design, such as the manufacturing engineer and quality assurance inspector, in accordance with ANSI GD&T.

These extensions to feature based design help improve the performance of the overall computer integrated manufacturing system. Identification of dimensional requirements melds the design system with the "upstream" specification of part function, while consistently and completely dimensioned part drawings aid in the "downstream" manufacturing and inspection phases of part production.

Work is currently underway on extensions to handle non-perpendicular surfaces and edges. In addition, the designer will be able to enter more general constraints on the geometry of the part.

BIBLIOGRAPHY

[1] *Dimensioning and Tolerancing*, American National Standard Y14.5M–1982, American Society of Mechanical Engineers, New York, NY, 1982.

[2] Freeman-Benson, B., Maloney, J. and Borning, A., "An Incremental Constraint Solver," *Communications of the ACM*, **33** no. 1, Jan. 1990, 54-63.

[3] 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.

[4] LeClair, S., "Rapid Design System: Memory-Driven Feature-Based Design," Proceedings of this Conference.