Int. J. Engng Ed. Vol. 22, No. 2, pp. 359–372, 2006
Printed in Great Britain.
0949-149X/91 $3.00+0.00

An Oriented Constraint Solving-Based Methodology Approach to Learn Dimensioning*

M. LUISA MARTÍNEZ and JESÚS FÉLEZ
Grupo de Ingeniería Gráfica y Simulación. Dpto. Ingeniería Mecánica y Fabricación, E.T.S. Ingenieros Industriales, Universidad Politécnica Madrid, C/ José Gutiérrez Abascal, 2, 28006, Madrid. E-mail: mlmmuneta@etsii.upm.es; jfelez@etsii.upm.es

Dimensioning is one of the most important stages in design development. It is traditionally learned through the sketches or drawings of parts to which the student adds their dimensional values. However, there are no computer teaching applications which are specifically oriented towards teaching the student how to dimension and showing what alternatives exist for such dimensioning. A methodology has been developed based on a computer application which in turn is based on variational geometry, which will allow students to design a sketch and obtain the different alternative dimensions, in line with ISO 129. Creating sketches through conceptual design implies the use of constraints. When developing conceptual design-based CAD programs, two independent modules must be created: on the one hand, the sketcher module, which must define the model's geometrical constraints and interpret the user's intention through a system of rules. On the other hand, the calculation module which must solve the final geometry and eventually dimension the mechanical part. The proposed approach establishes the complete geometry and constraints of a sketch and relates it with the complete dimensioning of the sketch. The developed methodology gives as a result a complete and consistent dimensioning of the sketch following the rules established by a standard like ISO. The methodology establishes the most suitable dimensioning but, if the student wants to substitute any dimension for another, the algorithm automatically reconfigures the complete dimensioning and proposes another different complete one.

Keywords: Dimensioning; technical drawing teaching; ISO 129; automatic sketching

INTRODUCTION

A KNOWLEDGE of drawing is essential to engineering. Drawing is present at every design stage, either as a preparatory sketch, an outline or an end product drawing. It is essential for the engineer to be able to obtain sketches which will serve as the starting point for drawings drawn by hand by the student himself, aided only by drawing instruments such as a compass and a ruler.

Drawing teaching in engineering courses has seen some slight changes in recent years. Teaching programs in this subject usually include the learning of a CAD program of a more or less complex nature. Sometimes other computer programs are used as a back-up, for example geometry programs to explain different subjects. Concepts such as dimensioning are usually learned in the classroom and then applied directly to the plans made by the student following the teacher's advice.

CAD programs enable dimensioning to be applied to a drawing since they include tools for linear, angular, and serial dimensioning, etc., and are capable of evaluating the magnitude of elements like straight lines, the radius of a circle or angles. Help tutorials which explain how to use these tools are usually provided by the programs *but they do not show what the best dimensioning is*. However, none of them is able to offer the different dimensioning alternatives for the same view or even detect redundancies or incoherencies in dimensioning. The effectiveness of a good dimensioning was studied by Turner [1] who analyzed technical drawings and concluded that there is 'noise' between what the drawing tries to express and what it shows. He pointed out that 70% of rejected parts are due to drawing errors. Around 40% of drawing errors were dimensioning errors. According to Folini [2] in the automobile sector, around 70% of the time invested in design is devoted to modifying previously produced models.

We do NOT know if studies exist that evaluate the improvements offered by CAD systems with respect to dimensioning errors in the drawings. Nevertheless, a good dimensioning is essential when preparing a technical drawing, since without this, the product will not be dimensionally appropriate. Various experiments have been carried out to improve spatial visualization and the visual skills needed by an engineering student using different commercial CAD programs such as AutoCAD [3], as well as studies into the results of applying different methodologies [4, 5] for teaching spatial visualization.

* Accepted 17 September 2005.

With the aim of helping our students to improve and better understand the dimensioning of a technical drawing, a new methodology has been developed. This is based on variational geometry which will allow the dimensioning of a sketch made by a student to be obtained, along with its different alternatives, and detect if the view is correctly dimensioned, over-dimensioned or if any dimensions are missing. This methodology has become reality in the development of a computer application which also detects the user's intention when making a sketch, and serves to introduce the student to the use of parametric CAD programs.

This system allows the student to work on various basic aspects of their training like: recognizing redundant and inconsistent topological and metric restrictions, detecting the user's design intentions, providing a dimensioning of the drawing based on international standards (ISO 129 in this case) in such a way that the program recognizes redundant and inconsistent dimensions introduced by the user, and is capable of determining which dimensions are alternative to others should the dimensioning not have a single solution.

THEORETICAL BASIS

CAD systems are a good answer for the design and representation of final products. Indeed, there exists a need to make conceptual design easier and to build-in better tools to do it. Parametric CAD systems are used in the design of families of objects which, in spite of having individual geometries, share the same topological constraints. Parametric models store the object's geometry through variable dimension parameters. Parametric design means more flexibility in the design process since geometry and constraints can be defined without specifying the actual dimensions of the object. Additionally, constraints allow for the description of dependencies between the elements which make up the object, as well as dependencies between objects.

Constraint-based modeling is one of the more modern approaches to product design. In it, engineering knowledge combines with the geometrical and topological knowledge that the designer has of the parts themselves. Constraint-based modeling, and design with features, is behind the development of the new CAD systems. The influence of the new techniques can be seen in the architecture of so-called parametric CAD systems. Besides the new geometry and topology modelers, sketchers and constraint solvers have cropped-up, becoming a key element in the new CAD systems. The sketcher serves as an interface with the designer when modeling with constraints. Also, rule-based methods are applied for automatic constraint detection. Constraint solvers evaluate the full set of constraints entered by the designer or inferred through the system of rules.

Variational geometry is a powerful method for defining and modifying geometric models, based on geometrical constraints rather than on defining geometry through Cartesian points. The theory of variational geometry was developed by Hillyard and Braid, [6, 7] and by Light, Lin and Gossard, [8, 9].

The architecture of constraint-based CAD systems is a hybrid where geometry and topology go hand in hand with modeling components and constraint solving.

Most constraints are set in 2D and only a few of them in 3D projections of a solid, so the sketcher module turns into one of the keys of this type of system. To obtain a consistent model of a mechanical part, every one of the constraints must be evaluated or solved. For this reason, the other key component is the constraint solver, which normally uses either numeric algorithms or a system of rules.

For the modeling to be efficient, the CAD system must aid the designer not only by solving a system of constraints, but must also help to formulate the constraints. This is achieved through automatic constraint detection.

Thus, the design process can be divided into three phases: sketching, constraint definition and solving. While doing the sketch or draft, the designer need not worry about exact dimensions. In the second phase, constraints can be liberally applied and the system can help by automatically detecting them. The third stage is done entirely by the system.

Constraint-based CAD systems can be grouped into two main categories, depending on their built-in solver [13]:

- **Equation solvers:** These generally employ numeric methods. Most of them use methods derived from Newton Raphson and need an initial value close to real; otherwise they show convergence problems. Other solvers use symbolic methods. This searches for a polynomial system having the same roots as the initial problem. Grobner's base [14] and Wu Ritt's Method [15, 16] are normally employed. Computation time is high. Yet other solvers employ propagation methods based on graphs where vertices are variables and constants and the edges represent their constraints.
- **Constructive solvers:** These systems follow the assumption that constructions of a sketch can be achieved simply by using a ruler, a compass and a protractor [17]. Normally they use either a symbolic solution of the systems through predicates or graph construction analysis [18].

This paper presents a new approach of the constraint-based solvers. The proposed approach establishes the complete geometry and constraints of a sketch and relates it with the complete dimensioning of the sketch. The developed methodology gives as a result a complete and consistent dimensioning of the sketch following the rules established by a standard like ISO. The methodology establishes the most suitable dimensioning but, if the user wants to substitute any dimension for another, the algorithm automatically reconfigures the complete dimensioning and proposes another different one.

The sketcher module

A variational parametric system consists of a set of geometric primitives whose dimensions and relative situation to each other are defined by a series of mathematical equations.

Complex geometries can be drawn in the sketcher by using three simple entities: lines, circles, and arcs. The combination of these elements and the application of geometric constraints will produce complex geometries.

To define the geometric primitives, a set of characteristic points has been established for each one. The above mentioned primitives have the following characteristic points:

- *Primitive:* characteristic points
- *Straight line:* both ends
- *Arc:* both ends and the center
- *Circle:* the center and one point on the circumference

The set of primitives underlying the geometry make up a vector of variables generically known as *generalized co-ordinates*. The set of variables has the following form:

$$\begin{aligned} q &= (x_{11} \quad x_{12} \quad \ldots \quad x_{i1} \quad x_{i2} \quad \ldots \quad x_{n1} \quad x_{n2}) \\ &= (q_1 \quad \ldots \quad q_j \quad \ldots \quad q_p) \end{aligned} \tag{1}$$

where, in a two-dimensional case, $p = 2 \times n$, the geometry we are looking for is determined by the values of q.

The geometric entities will be interrelated by a set of geometric conditions generically known as *geometric constraints*. In the most general case, each geometric constraint will be a nonlinear equation depending on the generalized co-ordinates, of the form:

$$R(q) = 0 \tag{2}$$

where $R(q) = 0$ is a set of m nonlinear equations with p unknowns.

Two types of geometric constraints are defined: topological and dimensional. Topological constraints are those which specify the position of the elements relative to one another. Examples of topological constraints are: perpendicularity, angles between straight lines, horizontality, verticality, orientation of straight lines, parallelism, tangency, symmetry, coaxiality, element alignment, etc. Topological constraints are used to create equations which relate the components of vector q to one another.

Dimensional constraints specify the actual dimensions of the mechanical part. The following are examples of dimensional constraints: X distance between two points, Y distance between two points, X–Y distance between two points, distance between two parallel lines, radius of a circle, and radius of an arc. Each dimensional constraint has associated with it a new variable d_I, generically known as *driven co-ordinate*. Driven co-ordinates will give the final geometric configuration of the mechanical part. Thus, the set of equations defining the geometry is of the form:

$$R(q, d) = 0 \tag{3}$$

Table 1 lists the equations used in some of the constraints considered.

Let's see an example. Given a rectangle such as the one in Fig. 2. Table 2 gives the variables and elements to consider.

Figure 3 shows the mechanical part constrained and dimensioned. To determine the final configuration of the geometry, the Equation System (2) must be solved. The solution to the sketch will be found when the system solves the equations given in Table 3. The equations in this system are nonlinear. All numerical methods of solving systems of nonlinear equations use iterative solving of linear equations approximating the set of equations by a first-order development series of the following form:

$$R(q) = R\left(q^{(k)}\right) + R_q\left(q^{(k)}\right)\left(q - q^{(k)}\right) \tag{4}$$

so the idea is to solve the following equations:

$$R\left(q^{(k)}\right) + R_q\left(q^{(k)}\right)\left(q - q^{(k)}\right) = 0 \tag{5}$$

The $R_q\left(q^{(k)}\right)$ matrix, generically known as J, is the Jacobian matrix of the system of equations. It is formed by the partial derivatives of each constraint equation in relation to the generalized coordinates of the system.

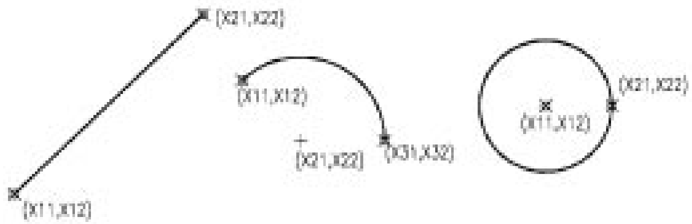


Fig. 1. Characteristic points of each primitive.

Table 1. Equations which define the geometric constraints.

Constraint	Equation
Perpendicularity	$\lvert \vec{u}_r \cdot \vec{u}_s \rvert = x \cdot x' + y \cdot y' + z \cdot z' = 0$ $(x_{12} - x_{11}) \cdot (x_{22} - x_{21}) + (y_{12} - y_{11}) \cdot (y_{22} - y_{21}) = 0$
Angle between two straight lines	$\cos(\vec{r},\vec{s}) \cdot \sqrt{(x_{12} - x_{11})^2 + (y_{12} - y_{11})^2} \cdot \sqrt{(x_{22} - x_{21})^2 + (y_{22} - y_{21})^2} -$ $-(x_{12} - x_{11}) \cdot (x_{22} - x_{21}) + (y_{12} - y_{11}) \cdot (y_{22} - y_{21}) = 0$
Horizontality	$y_{11} - y_{12} = 0$
Verticality	$x_{11} - x_{12} = 0$
Orientation of a straight line	$\cos(\vec{r},\vec{i}) \cdot \sqrt{(x_{12} - x_{11})^2 + (y_{12} - y_{11})^2} - (x_{12} - x_{11}) = 0$
Parallelism	$\lvert \vec{r} x \vec{s} \rvert = (x_{12} - x_{11}) \cdot (y_{22} - y_{21}) - (x_{22} - x_{21}) \cdot (y_{12} - y_{11}) = 0$ $d(P_m,\vec{r}) \cdot \sqrt{(x_{22} - x_{21})^2 + (y_{22} - y_{21})^2} - \left(\frac{(x_{12} + x_{11})}{2} - x_{21}\right) \cdot (y_{22} - y_{21}) -$ $\left(\frac{(y_{12} + y_{11})}{2} - y_{21}\right) \cdot (x_{22} - x_{21}) = 0$
Tangency between a straight line and an arc	$(x_{10} - x_{11}) \cdot (y_{12} - y_{11}) - (x_{12} - x_{11}) \cdot (y_{10} - y_{11}) - \text{radius} \cdot \sqrt{(x_{12} - x_{11})^2 + (y_{12} - y_{11})^2} = 0$ $(x_{12} - x_{11}) \cdot (x_{12} - x_{10}) + (y_{12} - y_{11}) \cdot (y_{12} - y_{10}) = 0$
Tangency between two arcs	$\sqrt{(x_{11} - x_{12})^2 + (y_{11} - y_{12})^2} - R_1 = 0$ $\sqrt{(x_{11} - x_{12})^2 + (y_{11} - y_{12})^2} - R_1 = 0$ $(x_{11} - x_{12})(y_{13} - y_{12}) - (y_{11} - y_{12})(x_{13} - x_{12}) = 0$

The Jacobian matrix plays a key role, not only in numerically solving the equations, but in the actual analysis of the constraints. It allows redundant conditions or insufficient definition of the system to be spotted. Adequate analysis of the Jacobian matrix lets us determine if redundant or inconsistent constraints exist, if more constraints are needed and which constraints are just alternative possibilities. It's general form is:

$$J = \begin{bmatrix} \frac{\partial R_1}{\partial q_1} & \frac{\partial R_1}{\partial q_2} & \cdots & \frac{\partial R_1}{\partial q_p} \\ \frac{\partial R_2}{\partial q_1} & & & \frac{\partial R_2}{\partial q_p} \\ \cdots & & & \cdots \\ \frac{\partial R_m}{\partial q_1} & \cdots & \cdots & \frac{\partial R_m}{\partial q_p} \end{bmatrix}$$

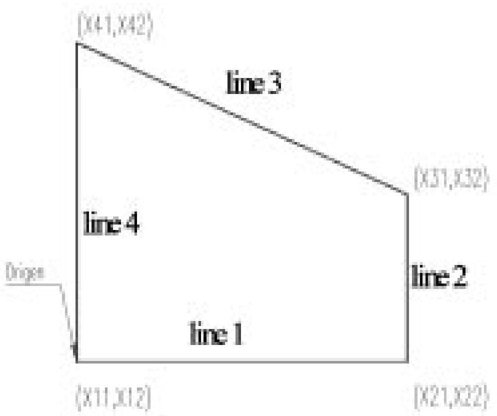


Fig. 2. Geometric constraints which define a profile.

To numerically solve the system of Equation (2) several numeric methods may be used. The most popular is Newton-Raphson's (in any of its variants). This is an attractive method due to it being quadratically convergent. It has, however, the important disadvantage of lacking convergence when the initial estimation is not close to the final solution.

Alternatively, minimization methods may be employed. These methods establish an error func-

Table 2. Constraint equations for the part shown in Fig. 2

Concept	Elements	Equations
Characteristic	Point 1	$(x_{11} \quad x_{12})$
points	Point 2	$(x_{21} \quad x_{22})$
	Point 3	$(x_{31} \quad x_{32})$
	Point 4	$(x_{41} \quad x_{42})$
Primitives	Line 1: from point 1 to point 2	$(x_{11} \quad x_{12}) - (x_{21} \quad x_{22})$
	Line 2: from point 2 to point 3	$(x_{21} \quad x_{22}) - (x_{31} \quad x_{32})$
	Line 3: from point 3 to point 4	$(x_{31} \quad x_{32}) - (x_{41} \quad x_{42})$
	Line 4: from point 4 to point 1	$(x_{41} \quad x_{42}) - (x_{11} \quad x_{12})$
Constraints	Position of point 1	$x_{11} = 0x_{12} = 0$
	Line 1 Horizontal	$x_{22} - x_{12} = 0$
	Line 4 Vertical	$x_{41} - x_{11} = 0$
	Line 2 parallel to line 4 at a distance of 100	$100 \cdot \sqrt{(x_{11} - x_{41})^2 + (x_{12} - x_{42})^2} + + \left(\frac{(x_{12} + x_{11})}{2} - x_{11}\right) \cdot (x_{12} - x_{42}) - \left(\frac{(x_{32} + x_{22})}{2} - x_{12}\right) \cdot (x_{11} - x_{41}) = 0$ $(x_{11} - x_{41}) \cdot (x_{22} - x_{32}) - (x_{12} - x_{42}) \cdot (x_{21} - x_{31}) = 0$
	Distance between points 2 & 3 is 25	$25 - \sqrt{(x_{31} - x_{21})^2 + (x_{32} - x_{22})^2} = 0$
	Distance between points 4 & 1 is 70	$70 - \sqrt{(x_{41} - x_{11})^2 + (x_{42} - x_{12})^2} = 0$

tion between the initial approximation to the solution, the constraint equations, and the final solution, and then try to minimize its value. From the standpoint of computational time, these methods are slower, but they ensure convergence. The developed approach uses one of these, known as the Levenberg-Marquard method.

This method tries to avoid the difficulties encountered by the Gauss Newton method when, at some point of the iterative process, the Jacobian matrix has an incomplete range. To avoid this, the direction $p_k = q - q_k$ is calculated by solving the following sub-problem:

$$\text{minimize} \left\{ \left\| R(q_k) + J(q_k)p_k \right\|_2^2 = \mu_k \left\| p_k \right\|_2^2 \right\}$$

where the parameter μ_k controls and limits the size of p_k. Note that p_k is defined even if $J(q_k)$ has an incomplete range. As $\mu_k \to \infty$, $\|p_k\|_2 \to 0$ and p_k becomes parallel to the direction of maximum slope.

Thus defined, this problem can be seen as equivalent to the problem of optimizing with conditions:

$$\text{minimize } \left\| R(q_k) + J(q_k)p_k \right\|_2 \text{ such that } \|p\|_2 \leq \delta_k$$

where $\mu_k = 0$ if conditions are not fully met (they aren't active) in q_k and $\mu_k > 0$ if they are. The set of feasible p vectors, $\|p_k\|_2 \leq \delta_k$, can be understood as the trust region of the linear model $R(q) \cong R(q_k) + J(q_k)p, p = q - q_k,$ to within which the search for the optimum of the problem is restricted.

Detecting the intent of design

Quite frequently in complicated sketches, when the designer has the responsibility for defining their constraints, there will be an insufficient amount of them for a complete definition, while some of the existing ones will be redundant. To avoid this, parametric design programs try to make the designer's task easier by internally using a set of rules generically known as *designer's intention rules*. These rules have formulations such as 'if the line drawn by the designer is almost perpendicular, the line in the sketch is perpendicular', or 'if two arcs drawn by the designer are

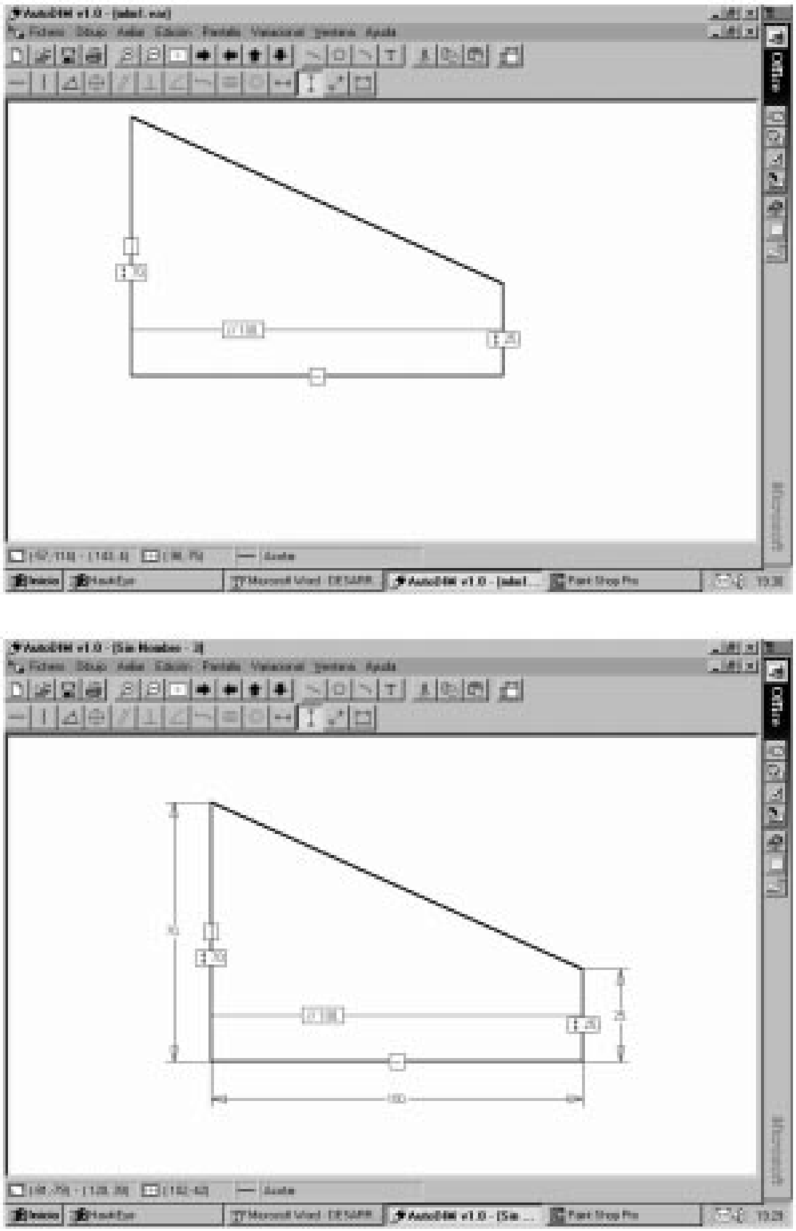


Fig. 3. Constrained sketch and its dimensioned result.

almost coaxial, the arcs in the sketch are coaxial'. To evaluate these rules, an *accuracy factor* is defined. This consists of a series of rule validation functions such that if the value given by a function is within the accuracy factor of the associated rule, the corresponding constraint is enforced. Otherwise, the next rule is checked (Fig. 4).

There are as many accuracy factors as there are constraints, so the user can modify and adjust them as he prefers. The program is able, after a few set exercises, to recognize a user's accuracy factor tendencies, which amounts to a customization of the program to the user (Fig. 5).

The detection of the designer's intent starts by verifying rules related to relationships, such as tangency, perpendicularity, etc. Once these are checked, verticality, horizontality and orientation of straight lines is verified. As a last step, the dimensional constraints are imposed. Here, alignment among elements is considered, and the distance between parallel elements, and vertical and horizontal distances is evaluated. Constraint

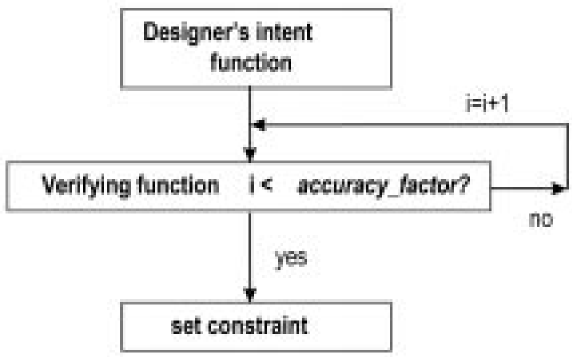


Fig. 4. Accuracy factor.

Fig. 5. List of accuracy factors.

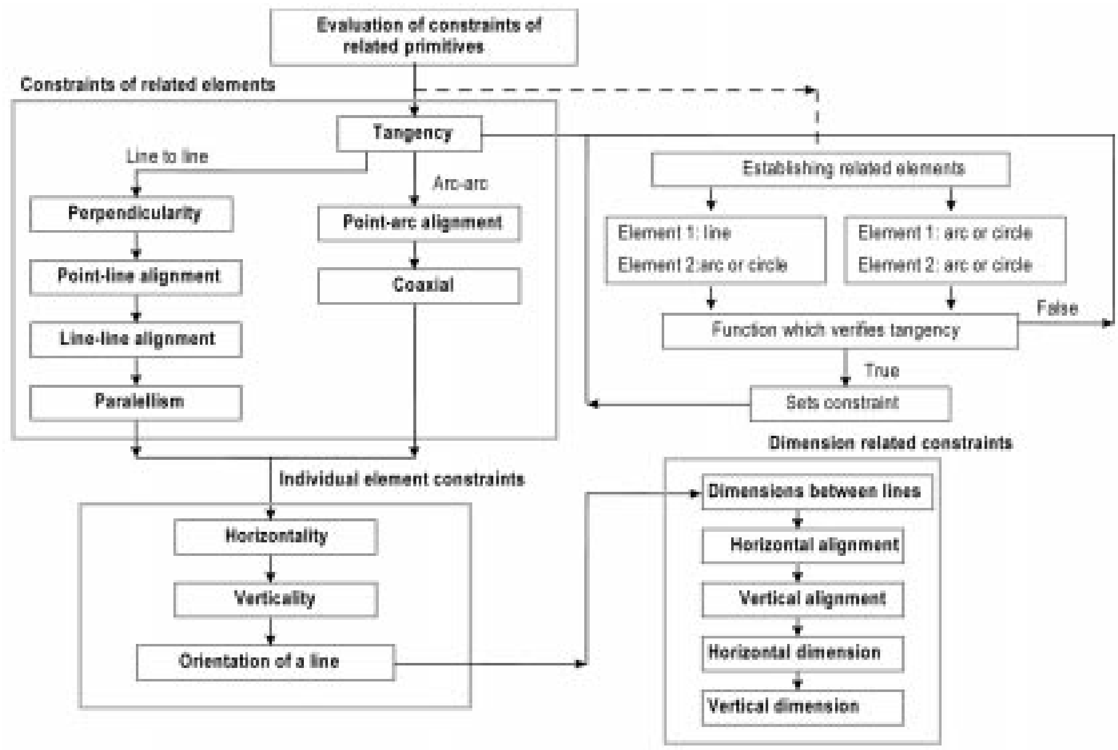


Fig. 6. Constraint application sequence.

verification within the program follows the diagram shown in Fig. 6.

At the same time as the constraints are being imposed, the range of the system is studied (Fig. 7). Constraints imposed due to the function giving a positive are checked for redundancy. If there is no redundancy, the range of the system is calculated. Once it coincides with the number of unknowns, the system can be solved. If the constraint is redundant, the system prompts for confirmation. If the user decides to go ahead with the constraint, the lesser pivot equation is eliminated, so as not to increase the range of the system.

These operations can be automatically carried out by the program or they can be used as an additional help when implementing the sketch constraints.

Once all the topological and dimensional constraints have been included in the sketch, the system will have all the information necessary for design; all that remains is to properly place them around the mechanical part following ISO 129.

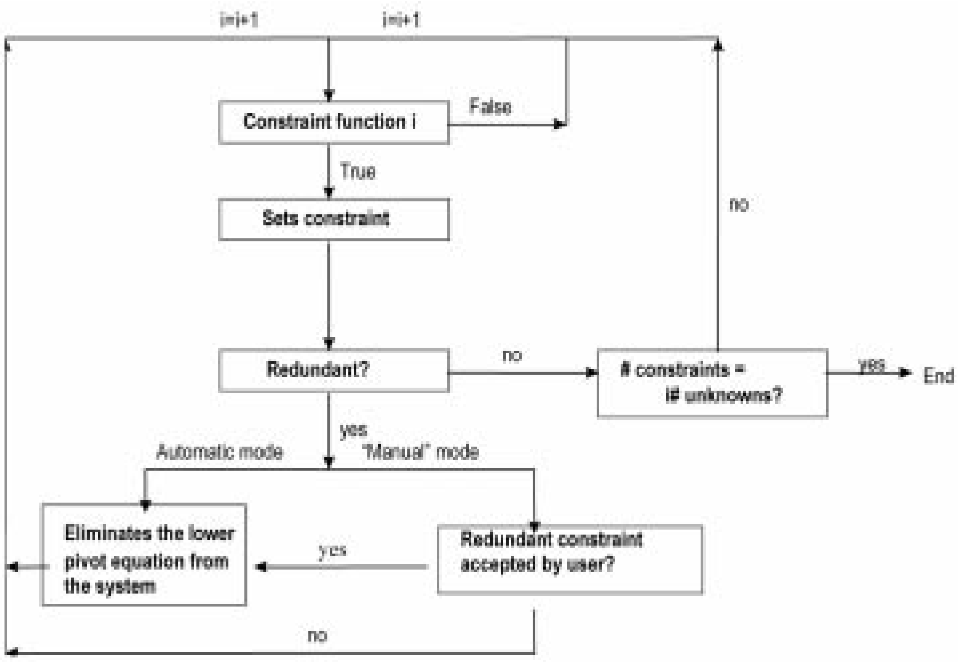


Fig. 7. Study of the system's range.

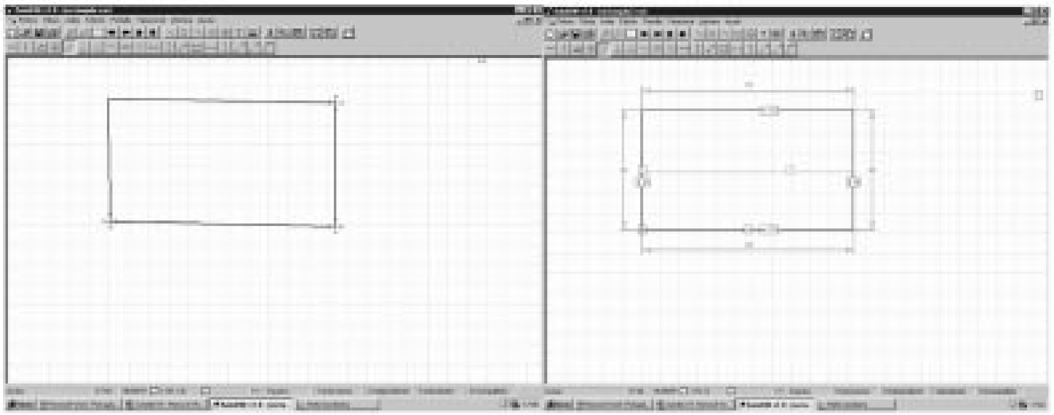


Fig. 8. Rectangle constrained and it's dimensioning. The parallelism constraint is redundant.

DIMENSIONING PROCEDURE

Current variational systems, once the geometry is defined, are not capable of establishing the mechanical part's dimensioning, although some authors have considered this matter [22]. At most, they allow the addition of dimensions imposed by the user while defining the constraints themselves, but they cannot deduce a complete dimensioning from topological and dimensional constraints. Neither can they reorganize these dimensions, so their position must be defined by the user. The proposed method allows the complete lay-out and standardized dimensioning of the sketch. To this end, dimensions are associated with the constraints. Not all constraints produce dimensions; such is the case, for instance, of perpendicularity, horizontality, verticality or alignment conditions.

The program allows either the automatic dimensioning of the mechanical part or the adding of topological and dimensional constraints step by step. Additionally, it has algorithms which establish the orientation of the dimensions as linear, angular, or radial. These layout criteria follow the ISO 129 standard.

Analysis of the Jacobian matrix lets us determine if constraints are sufficient and which, if any, are inconsistent or redundant. The program displays this either as textual information or by arrows indicating the degrees of freedom of the geometry, which disappear as the coordinates of the associated points become constrained. Then, the designer can correct any errors, where constraints are inconsistent or redundant, or add new ones in order to complete the sketch. The chosen pivoting method selects as independent constraint the one with the greater absolute

Table 3. Priority factors

Constraint	Priority factor	Constraint	Priority factor
Perpendicularity	1	Symmetry	1
Horizontality	1	Position of points	1
Verticality	1	Distance between two points	0.1
Orientation of a line	1	Vertical distance between two points	0.1
Parallelism	1	Horizontal distance between two points	0.1
Alignment of points	1	Angle constraint between lines	0.1
Alignment of lines	1	Distance between parallel lines	0.1
Alignment between a point and a line	1	Radius of arcs	0.02
Coaxiality	1	Angles between lines	0.001
Belonging	1		

value. This can lead to the selection as independent or redundant of dimensionally inconsistent constraints, as exemplified in Fig. 8.

The Jacobian matrix of this rectangle is:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -0.6 & 0 & 0.6 & 0 & -0.6 & 0 & 0.6 & 0 \end{bmatrix}$$

The lesser pivot equation, the last row, corresponds to the parallelism constraint and the program, since it is the ninth equation and has the lesser pivot, classifies it as redundant. It would not seem appropriate that a dimensional constraint, in this case a distance of 113 between the lower vertices, should override a topological one.

Since the program uses automatic dimensioning algorithms and detects dimensions to complete the sketch, the set of independent constraints must be prioritized. The pivoting method used reorganizes the matrix following the order of magnitude of the pivot. This feature can be taken advantage of by assigning certain constraint equations a priority relative to others. This can be achieved by a value known as *priority factor* which multiplies the constraint equation. When the terms of the Jacobian matrix are obtained, they are multiplied by this factor. The pivoting method will follow the new order when selecting the pivots that are selecting as independent the equations with the highest priority index. This way, dimensional constraints tend to become redundant instead of the higher order topological constraints.

Table 3 gives the priority factors assigned to the constraints.

Once independent, redundant and inconsistent dimensions are identified, they must be represented according to rules. To that end, a series of algorithms have been developed which allow dimen-

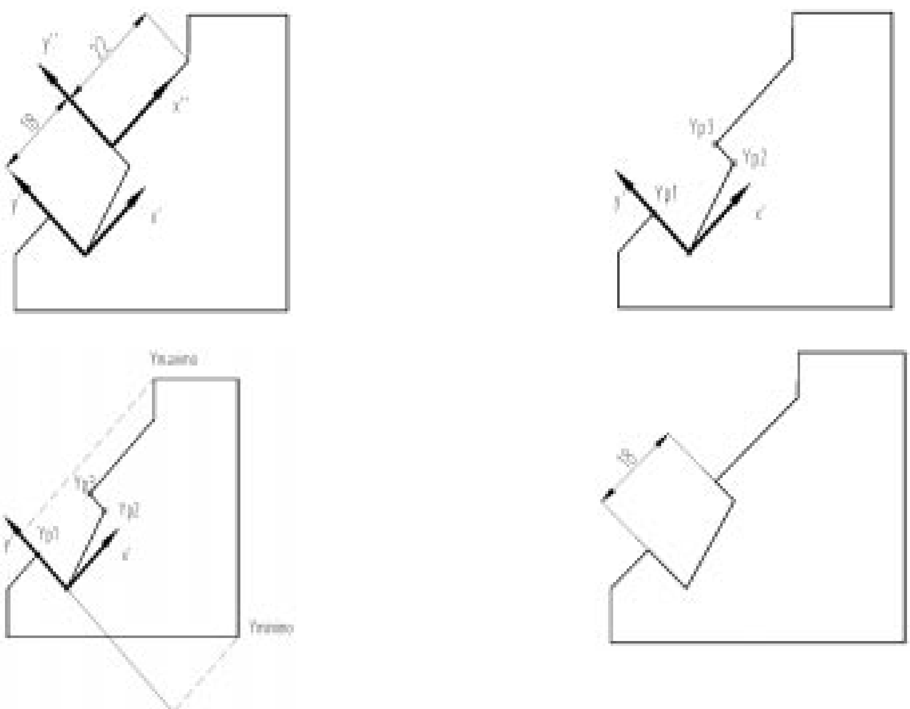


Fig. 9. Dimension layout algorithm.

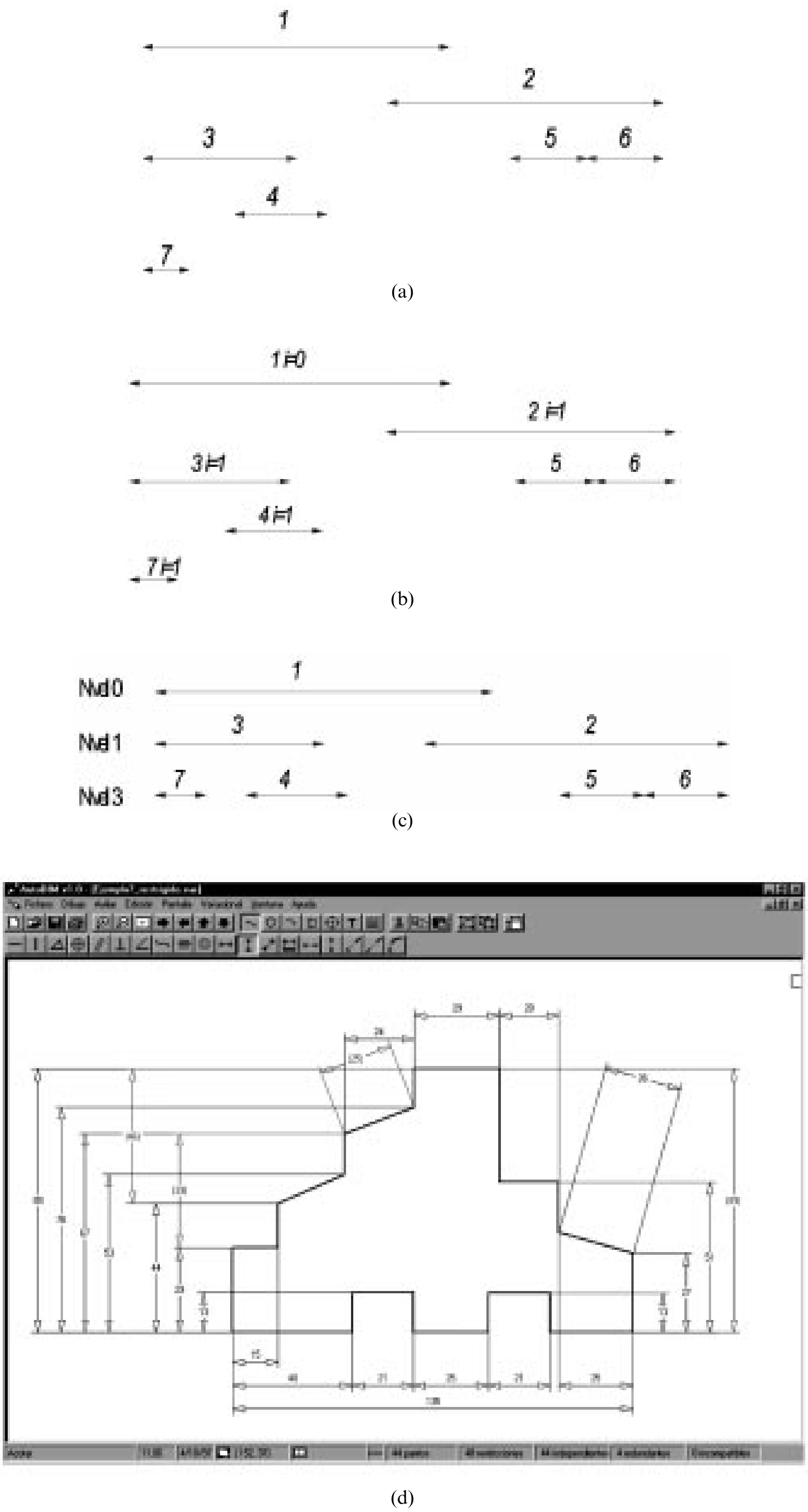


Fig. 10. Dimension nesting algorithm: (a) dimensions before nesting; (b) nesting algorithm, Step 1; (c) results of nesting; (d) an example.

sions to be represented on the nearest side of the element to be dimensioned as well as the nesting of dimensions according to their length.

The algorithms for locating dimensions let you find the nearest end of the element on each orientation. For each of these a system of local co-ordinates is created per dimension. The center of this new system of co-ordinates is the point of reference of the dimension of lesser X-axis, *Yp1*, *Yp2*, *Yp3*, and *Yp4*, characteristic points of the elements to be dimensioned in the new system of co-ordinates. The maximum and minimum Y-axis points (*Ymax* and *Ymin*, respectively) are also calculated for the new system. The distance between *Ymax* and the points *Yp1* and *Yp3* is compared to the distance between *Ymin* and the points *Yp2* and *Yp4*. The dimension will be situated in the nearest end.

Once the dimensions are sorted by type, orientation and layout around the element, they are ordered by length and then follow a nesting algorithm. Each group of similarly oriented dimensions, on the same side of the element, is sorted by length. The longest one has a nesting index of $i = 0$. Shorter ones receive a nesting index of $i = 1$ as long as they are either fully included in or overlap the longer dimension. The next magnitude is then sorted, and all dimensions that are either fully included or overlap it add one to the previous index $i = 1 + 1 = 2$. This procedure is repeated until the lesser dimension is reached. Fig. 10 illustrates this graphically.

Figure 12 shows the constrained sketch of an element and its dimensioning. User-provided redundant dimensions appear in the dimensioning. When displaying redundant dimensions, these are shown as an auxiliary dimension with the numbers in parenthesis; in the case of inconsistent dimensions, the number appears struck-through.

Alternative dimensions

Generally, dimensions established for a mechanical part must be modified because they do not meet the needs of the drawing's users. For example, Fig. 12a, dimensioned following constructive criteria, changes to Fig. 12b when production criteria are followed.

It is possible to detect equivalent dimensions by allowing user dimensions to be entered and eliminating unnecessary ones. It is the user who decides which dimensions get inserted in the sketch. On doing this an additional equation is added to the system. If it is a perfectly constrained system it will thus become over-constrained, the new dimension will be redundant, and it can be expressed as the linear combination of the preceding geometric constraints:

$$R_{n+1} = \lambda_1 R_1 + \lambda_2 R_2 + ... + \lambda_n R_n$$

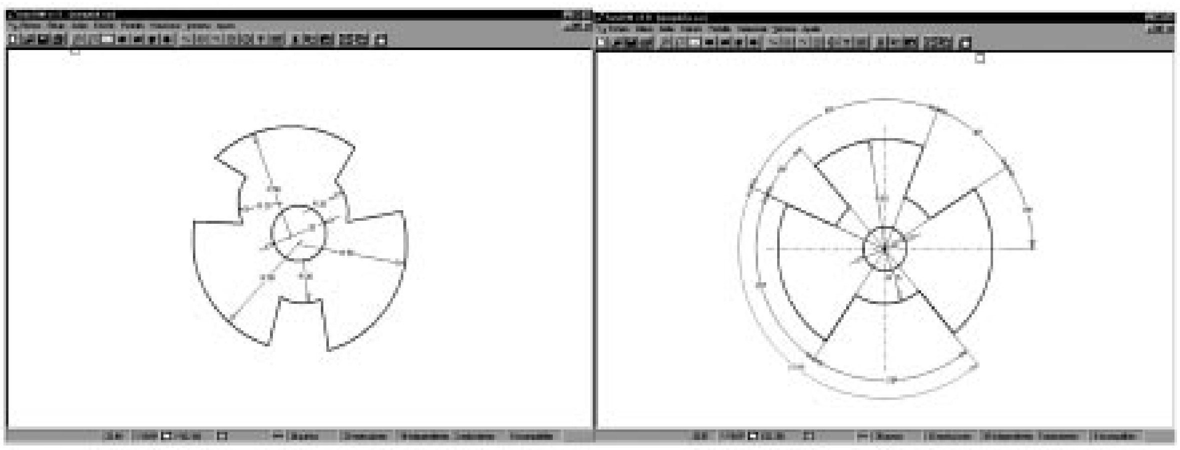
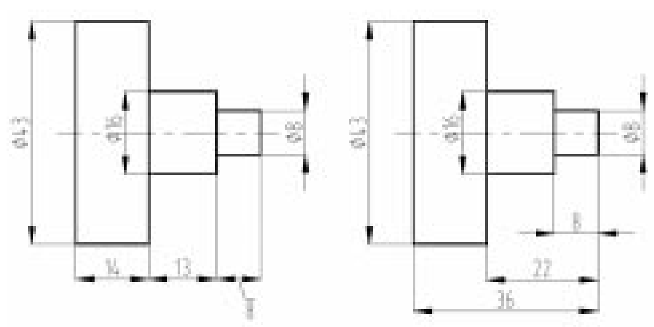


Fig. 11. Sketch and dimensioning of a part.



(a) (b)

Fig. 12. (a) Geometric dimensioning; (b) production-oriented dimensioned.

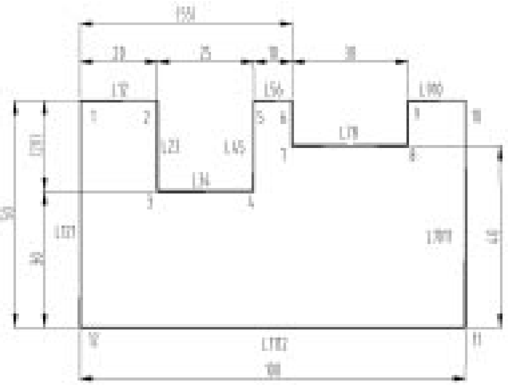


Fig. 13. Element with redundant dimensions.

By solving the system and finding the values of λ_i, as follows:

$$J^T\lambda = R_{n+1}$$

We will obtain $\lambda_i = \pm 1$ coefficients for the metric constraints on which it depends, and other values for λ_i for the rest of the constraint equations if it depends on them. In this way, any of the dimensional equations can be substituted by the dimension that the user wants. Given the following example (Fig. 13):

We want to include a vertical dimension of 20 and a horizontal dimension of 55. Therefore, other dimensions will have to be eliminated to avoid over-constraining. The first column includes the constraint number. In the list, these two dimensions are known as R_{25} and R_{26} respectively. We are looking for values such that:

$$R_{25} = \lambda_1 R_1 + \lambda_2 R_2 + \ldots + \lambda_{24} R_{24}$$
$$R_{26} = \lambda'_1 R_1 + \lambda'_2 R_2 + \ldots + \lambda'_{24} R_{24}$$

Solving the above systems, we get:

$$R_{25} = R_3 + 4.4959R_7 + R_{22} - R_{23}$$

$$R_{26} = 0.436R_3 - 0.0218R_8 + 0.0718R_9 + 0.05R_{10} + R_{18} + R_{19} + R_{20}$$

In the first case, we observe that the values of λ_3, λ_{22}, and λ_{23} are ± 1, as expected. R_3, a horizontality restriction, is assigned to a dimension with a null value, R_{23} and R_{22} are the vertical dimensions defined with values of 30 and 50, respectively. The vertical dimension whose value is 20 can substitute for one of these two. Additionally, the + sign associated with constraint R_{22} and the—sign associated with constraint R_{23} indicate that R_{25} can be obtained by subtracting R_{23} from R_{22}.

In the second case, the dimensional constraints once again offer a coefficient of 1, and additional dependencies appear. If the horizontal dimension with value 55 is added, one of the others, with values 20, 25, and 10, must be eliminated. Additionally, the three plus signs associated to these dimensions indicated that the one with value 55 is obtained by adding the three preceding ones.

Application of the methodology

Students in the first course of engineering have to do basic drawing and advanced drawing (drawing of assemblies). Both subjects take up 130 hours in the year, and study and practice in dimensioning sketches is essential. The basic study plan is shown below in Fig. 14.

During the class, the teacher explains the concepts needed to perform appropriate dimensionings and puts forward different examples to the students for later discussion. The student does these exercises at home with the help of a computer application. This guarantees a solution with no redundant or inconsistent dimensions. The results are later discussed in class where the teacher comments on the advantages or disadvantages of the different solutions proposed by the students. The disadvantages of serial/in-series dimensioning are then explained, along with functional dimensions, and the adaptation of dimensioning to different types of manufacturing.

CONCLUSION

In this paper a new methodology approach of teaching dimensioning based on variational geometry has been presented. With the aim of helping engineering students and reinforcing their learning of sketch dimensioning, a computer application based on variational geometry has been developed. The proposed approach establishes the complete geometry and constraints of a sketch and relates it with the complete dimensioning of the sketch. The developed methodology gives as result a complete and consistent dimensioning of the sketch following the rules established by an international standard like ISO 129.

This approach allows the user's intention to be detected at the time of making the sketches. This

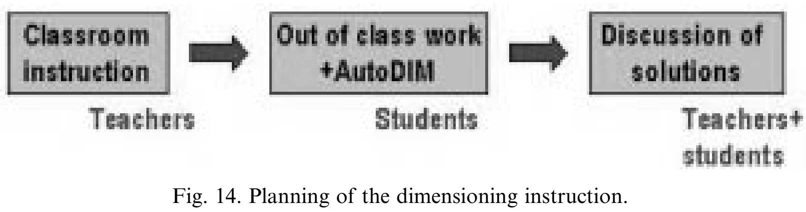


Fig. 14. Planning of the dimensioning instruction.

means that the drawing is simple and similar to what the student would produce on a sheet of paper. The methodology also establishes the most suitable dimensioning but, if the student wants to substitute any dimension for another, the algorithm automatically reconfigures the complete dimensioning scheme and proposes another different one. It is possible to detect redundant and inconsistent dimensions introduced into the sketch by the user. It also includes algorithms to dimensioning the element, following the ISO 129 standard. Lastly, it can detect alternative dimensions to a given one, thus offering several possible dimensionings of one sketch.

REFERENCES

1. B. T. Turner, Design into practice, *Effective Communication,* September 1967.
2. F. Folini, Specifiche funzionalli per un sistema parametrioc-variazionale, Atti del convegno ADM 'Evoluzione del disegno nell'integrazione delle politiche e delle metodologie industriali', *Caserta Aversa,* Italy, September 27–29, 1995, ADM, pp. 149–155.
3. M. L. Mohler, An instructional method for the AutoCAD modeling environment, *Engineering Design Graphics Journal,* **5**(13), Winter 1997.
4. C. L. Miller and G. R. Bertoline, Spatial visualization research and theories: their importance in the development of an engineering and technical design graphics curriculum model, *Engineering Design Graphics Journal,* **55**(3), 1991, pp. 5–14.
5. C. Potter and E. Van derWermer, Perception, imagery, visualization and engineering graphics, *European Journal of Engineering Education,* **28**(1), 2003, pp. 117–133.
6. R. C. Hillyard and I. C. Braid, Analysis of dimensions and tolerances in computer aided design, *Computer Aided Design,* **10**(3), 1978, pp. 161–166.
7. R. C. Hillyard and I. C. Braid, Characterizing non ideal shapes in terms of dimensions and tolerances, *Computer Graphics,* **12**(3), 1978, pp. 234–238.
8. R. A. Light, Symbolic dimensioning in computer aided design, MS Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (1980).
9. V. C. Lin, D. C. Gossard and R. A. Light, A variational geometry in computer aided design, *Computer Graphics,* **15**(3), 1981, pp. 171–177.
10. V. C. Lin, Three dimensional variational geometry in Computer Aided Design, MS Thesis Massachusetts Institute of Technology, Cambridge, Massachusetts, USA (1981).
11. R. A. Light and D. C. Gossard, Modification of geometric models through variational geometry, *Computer Aided Design,* **14**(4), 1982, pp. 209–214.
12. R. A. Light and D. C. Gossard, Variational geometry: a new method for modifying part geometry for finite element analysis, *Computer & Structures*, **17**(5–6), 1982, pp. 903–909.
13. R. J. Joan-Arinyó and A. Soto, A correct rule-based geometric constraint solver, *Computer & Graphics,* **21**, 1997.
14. C. Hoffmann, Geometric *and Solid Modelling,* Morgan Kaufmann, USA (1989).
15. C. Chou, *Mechanical Theorem Proving,* Kluwer, Netherlands (1987).
16. W. Wen-Tsün, Basic principles of mechanical theorem proving in geometries, *J. Syst. Science & Mathematics Science*, **4**, 1986, pp. 207–235.
17. W. Bouma, I. Fudos, J. Hoffmann, J. Cai and R. Paige, Geometric Constraint Solver, *Computer Aided Design,* **27**(6), 1995, pp. 487–501.
18. I. Fudos, Editable representations for 2D geometric design. Master Thesis, Purdue University (1993).
19. M. L. Martínez, J. Félez and A. Carretero, A variational CAD System oriented to automatic dimensioning, *Proc. CompuGraphics'97.*
20. R. Joan-Arinyo and A. Soto-Riera; Combining constructive and equational geometric constraint solving techniques, *ACM Trans. Graph,* **18**(91), 1999, pp. 35–55.
21. Robert Joan-Arinyo, Nuria Mata and Antoni Soto-Riera, A constraint solving-based approach to analyze 2D geometric problems with interval parameters, *Proc. 6th ACM Symp. Solid Modeling and Applications*, 2001.
22. Chen Ke-zang, Feng Xin-an, Lu Quan-sheng, Intelligent dimensioning for mechanical parts based in feature extraction, *Computer Aided Design*, **33**, 2001, pp. 949–965.

M. Luisa Martinez is an Ass. Professor of the Engineering Graphics Group and Simulation of the Technical University of Madrid (Spain). She got her Ph.D. Degree in 1997 working on variational geometry. Formerly she teaches Drawing and Computer Graphics in the Mechanical Engineers School and researches on Drawing education and EAO. During this time she has been involved in different education projects and pilot activities promoted by the European Commission and other Spanish institutions.

Jesus Félez received his Mechanical Engineer and Doctoral degrees from the University of Zaragoza in 1985 and 1989. He started as Associate Professor at the Technical University of Madrid in Spain (UPM) in 1990 and becomes Full Professor in 1997. His main activities and research interests are mainly focused in the field of simulation, computer graphics and

virtual reality. He has published over 50 technical papers and has been actively involved in over 25 research and development projects. He has served as thesis advisor for 30 master's theses and four doctoral dissertations. Prof. Felez is member of ACM, SCS and IEEE having a very active participation. He is also member of the International Program Committee of the Bond Graph Modeling Conference of SCS.