Sensing Strategies Based on Manufacturing Knowledge

Thomas C. Henderson, Tarek M. Sobh, Frédéric Zana, Beat Brüderlin, and Ching-Yao Hsu [1]

UUCS-94-023

Department of Computer Science
University of Utah
Salt Lake City, UT 84112 USA

August 30, 1994

Abstract

We propose an approach for the inspection of machined parts that is based on knowledge of the actual manufacturing process for the parts to be inspected. A principal benefit of this approach is that sensing can be focused on those areas of parts where violations of tolerance specifications are most likely. NC toolpaths are used as a low-level unifying representation for the analysis of geometry and tolerances in design, manufacturing, and inspection.

[1] This work was supported by the Advanced Research Projects agency under Army Research Office grant number DAAH04-93-G-0420. All opinions, findings, conclusions or recommendations expressed in this document are those of the author and do not necessarily reflect the views of the sponsoring agencies. This report appears as a paper in the proceedings of the 1994 ARPA Image Understanding Workshop.

Sensing Strategies Based on Manufacturing Knowledge*

Thomas C. Henderson, Tarek M. Sobh, Frédéric Zana,
Beat Brüderlin, and **Ching-Yao Hsu**
Department of Computer Science
University of Utah
Salt Lake City, UT 84112

Abstract

We propose an approach for the inspection of machined parts that is based on knowledge of the actual manufacturing process for the parts to be inspected. A principal benefit of this approach is that sensing can be focused on those areas of parts where violations of tolerance specifications are most likely. NC tool paths are used as a low-level unifying representation for the analysis of geometry and tolerances in design, manufacturing, and inspection.

1 Introduction

Sensing for inspection involves making measurements of an object or process and comparing the results to a desired specification. There are unavoidable errors involved in the processes generating the phenomena being measured. In manufacturing, costs are often inversely related to the precision with which this variability must be controlled. As a result, minimally sufficient tolerances are often associated with specifications. Process engineering in part involves the choice of a manufacturing plan which can meet these specifications with the least expense. Inspection operations insure that the specifications have in fact been achieved.

The importance of quantifying tolerance in the specification, design, manufacturing and inspection process is obvious. Unfortunately, adequate representations of tolerance do not exist which permit dialog between these various aspects of the manufacturing process. This lack is particularly acute in systems which tightly integrate all of the aspects of prototyping. Tolerance specifications, whatever their form, must have the same meaning for all steps in the process. Maximum acceptable deviations must be optimized across all steps so that productivity is not adversely impacted by requirements for unneeded precision; information about the tolerance required and the manufacturing processes also make it possible to focus sensing on areas of higher systematic or statistical concern. We propose to use the tolerance specification in conjunction with knowledge of the manufacturing process plans to determine more optimized sensing strategies.

While published standards such as ANSI-Y14.5M [ANSI, 1982] exist for describing manufacturing tolerances, they are not yet sufficiently complete and unambiguous to precisely specify the information needed for automated inspection. Furthermore, these standards provide no help in determining how to optimize the inspection process so as to increase the confidence that out of tolerance features are discovered with a reasonable expenditure of resources. In this paper, we give preliminary examples of an approach to this problem based on a detailed understanding of the manufacturing processes involved in the production of machined parts.

2 Tolerances

Our goal is to develop a methodology which helps to guarantee that the intended tolerance specification is met as efficiently as possible. There are two crucial issues related to tolerance specifications:

- *validation of the design, subject to imposed tolerances,* and
- *validation that tolerances have been achieved in the actual part.*

The first of these is concerned primarily with the analysis of tolerance stack-up in assembly using geometries. The second of these involves sensor measurements either during the manufacturing phase or post-manufacture inspection, and is the subject of this paper. To ensure that the tolerance has been met, sensors are used to:

- measure the manufacturing process (e.g., table position during NC milling),
- measure parameters of manufacturing features (e.g., hole diameter), and
- measure points on the surface directly and analyze them.

Of course, sensor error/uncertainty must be accounted for [Noble and Mundy, 1993].

We propose to synthesize process monitoring and inspection strategies based on detailed knowledge of geometry, tolerance specification, manufacturing features and processes,

*This work was supported by the Advanced Research Projects agency under Army Research Office grant number DAAH04-93-G-0420.

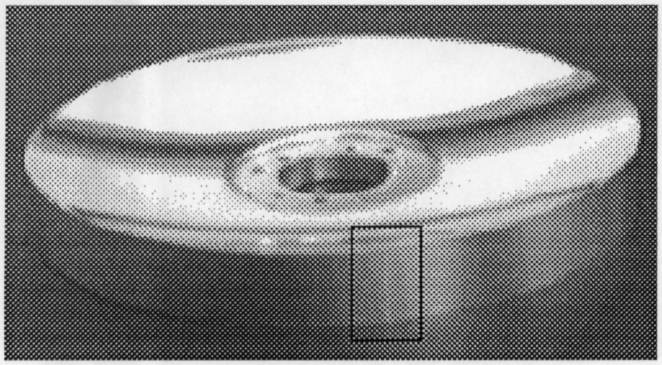

Figure 1: Mirror

and the sensors involved. As an example, consider the laser spot or heat source tracker currently being prototyped and manufactured as part of the ARPA Madefast program. Figure 1 shows the mirror part of the device; the box outlines a rough bump in the supposedly circular outer edge of the mirror. This is caused by the fact that the 3-axis mill sometimes performs differently during startup/slowdown which occurs at this point. We believe that this kind of systematic problem is a perfect example of something that our approach can help avoid or detect.

The question here is whether parts of uncertain shape fulfill certain functional specifications. The question is expressed as geometrical relationships between toleranced objects. Unfortunately, tolerance based relations are often inconsistent, unlike relations between exactly represented objects. We have presented elsewhere a survey of tolerance representation and analysis methods [Hsu and Bruderlin, 1994], and have derived an intuitionistic tolerance handling method for robust modeling [Bruderlin and Fang, 1992, Fang and Bruderlin, 1991, Zhu *et al.*, 1993].

Our methods allow us to simulate manufacturing tolerances, and thus simulate the validity of the design under these tolerances. For instance, we can find out whether a functional feature can be manufactured, and whether it has certain relationships with other features (within tolerance), and whether the relationships are logically consistent. The geometric operations that need to be carried out can be very similar to the solid modeling operations done in the design stage. We geometrically construct the object by machinable features (e.g., drilling a hole corresponds to a Boolean subtraction of a cylinder, etc.); however, for this we associate tolerances that correspond to the tool precision to the geometric elements, rather than the floating point, or design tolerances, used previously (see [Hsu and Bruderlin, 1994] for details).

After the geometric construction we can query relationships between geometric elements and features to test the validity of the functional features (e.g., to determine whether two holes manufactured by two independent drilling operations line up within the tolerance of the design specifications). The adaptive tolerance method of the intuitionistic geometry approach facilitates tolerance analysis and synthesis in that it can be used to determine whether certain relationships can be achieved unambiguously under the current tolerances, and it provides the necessary feedback, indicating that the precision of the individual objects in the relation need to be tightened in some cases, or that the features need to be manufactured with a more precise tool, or an additional finishing stage may become necessary. In other cases the analysis may tell us that the tolerances can be relaxed, or that the clearance needs to be increased.

To validate that tolerances have been achieved requires the allocation and management of sensing resources in order to monitor specific parts of the machining process and afterwards to measure particular features of the part. We can exploit earlier work in process monitoring [Okafor *et al.*, 1993, Spyridi and Requicha, 1994], both to predict likely deviation from the process, as well as to determine the most appropriate sensing to detect such deviations. Furthermore, our technique provides the high-level goals (e.g., features to be inspected) to drive sensing strategies such as developed at Columbia [Abrams and Allen, 1992, Abrams *et al.*, 1993] and others.

The usual approach to validation is to simply measure the geometry resulting from the manufacturing process and compare it to the nominal geometry from the CAD model. We believe that a stronger approach, exploiting knowledge of the process plan and the particular manufacturing process, is possible, and that this approach permits the automatic synthesis of sensing strategies.

To achieve this requires a tolerance specification which:

- specifies design geometry tolerance as well as tool path tolerances, and
- helps locate high payoff (i.e., maximal information gain) inspection regions.

We are working with the Alpha_1 Computer Aided Geometric Design system and exploit its ability to generate process plans for 3 and 5-axis NC milling machines. For these machines, the process is a set of tool paths with appropriate tools, speeds, etc., specified. Thus, a sensing strategy is a set of sensing operations carried out at particularly high risk parts of the tool path or places on the completed part.

3 Process Plans

The standard representations for Computer Aided Design include volumetric, boundary and CSG models. Current advanced modelers, can produce process plans for specific machines in order to manufacture the object. We believe that the process plan and associated information (e.g., the tool path, the tool to be used, its speed, etc.) provide a strong basis for analyzing the manufacturing and inspection steps with respect to tolerances.

A tolerance specification on the shape geometry must be transformed into the corresponding tolerance on the machining operation and vice versa. This in turn can be used to select an appropriate manufacturing process, given knowledge of the manufacturing accuracy of the process. This yields direct methods for deciding on sensing strategies both to monitor the manufacture of the part, as well as for post-manufacturing inspection. These sensing strategies are derived from an analysis of where the tool path is most likely to deviate from the tolerance specification.

These must all be done as efficiently as possible; in particular, it must be:

- straightforward to choose the cheapest manufacturing process, to go as fast as possible on that machine,
- to make as few sensed measurements as possible, and
- to perform as little computation as possible.

The keys to our approach are:

- have/use knowledge about each feature and machining process for that feature, and
- exploit the tool path representation to guide analysis and sensing strategies.

In order to structure the analysis process, we focus here on NC milling, and use the tool path as the basis upon which design and manufacturing tolerance and sensor measurements will be compared. Much as operational semantics allows the meaning of a high level program to be defined in terms of the particular architecture upon which it executes, so can the CAD specification of a part be defined in terms of the machining operations which produce its shape.

Given the CAD geometry for a part, a tolerance specification, and a class of NC mill to be used, generic knowledge about such mills can be used to generate a desired tool path with its associated tolerance (call it TP_d). Once a specific mill is selected, the nominal tool path from TP_d together with the accuracy of the mill determine the actual tool path (call this TP_a). These two tool paths allow us to determine a great deal about the efficiency and uncertainty regions of the process.

First, if $TP_a \subset TP_d$ is true, then we know that the tolerance should, in principle, be achieved. If $TP_d - TP_a$ is large, then the selected machine may be too precise, and therefore, too expensive. If the boundary of TP_a is close to that of TP_d, this signals places where sensing may be necessary to guarantee the inclusion relation. This also gives insight into how accurate the sensing needs to be. Even if TP_a is not contained in TP_d, this approach allows us to estimate what percentage of milled parts will be out of spec, and thus an informed decision can be made whether to tighten the accuracy of the machine, or where to sense with high probability of part error. Thus, the tool path representation allows insight into design, manufacture and inspection in a common framework.

For this approach to work well requires a clear and efficient implementation technique, and we propose the interval spline for this purpose. The use of interval Bezier curves for a complete description of approximation errors was proposed by Sederberg and Farouki [Sederberg and Farouki, 1992] (see paper for details). The basic idea is to extend splines to polynomials whose coefficients are intervals with well defined arithmetic operations. Such splines define a region in space rather than a curve. This notion captures very nicely the semantics of a tolerance specification. We have developed interval curves for both 2D and 3D.

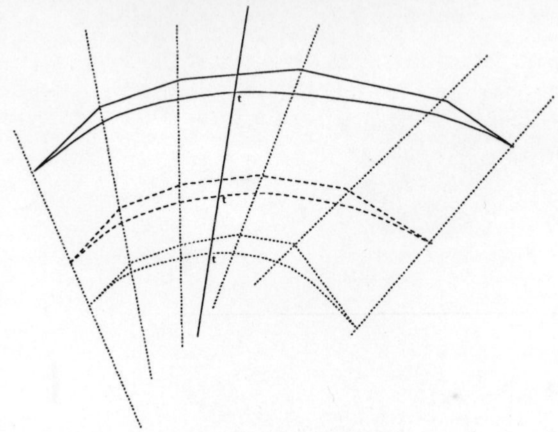

Figure 2: One Interval Spline

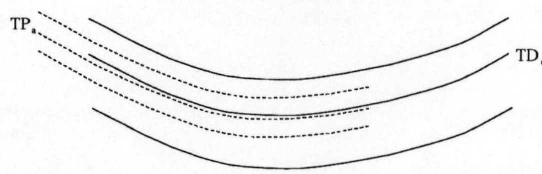

Figure 3: $TP_a \subset TD_d$

For curves in 2D, an interval is a set of 3 points (corresponding to the nominal point and two bounds). The spline interpolation is done (on 6 consecutive points) separately on each of the 3 corresponding curves (see Figure 2). Note that since the initial 3 points are on a line, evaluation at any parameter $t \in [0, 1]$ yields 3 points on a line. As indicated above, the determination of inclusion of one interval spline within another is an important question. Figure 3 shows the case where inclusion is true. We have developed a technique to answer this question (see Appendix A). Moreover, if one interval contains another, then the area of the difference of the two intervals is also determined. In 3D, we assume that the uncertainty around a point is described by an ellipse (in the plane normal to the curve). The problem becomes how to determine if one ellipse is inside another. We have developed an algebraic solution to this problem (see Appendix B).

4 Conclusion

We propose tool paths with tolerances as a unifying approach to dealing with tolerance issues across design, manufacturing and inspection. Not only does this permit us to answer questions concerning design and manufacturing processes, but also gives a way to determine places in the process and on the part where sensing is useful to ensuring that tolerances are met. We have developed algorithms based on interval splines and implementation is currently underway. We consider our major contributions to be:

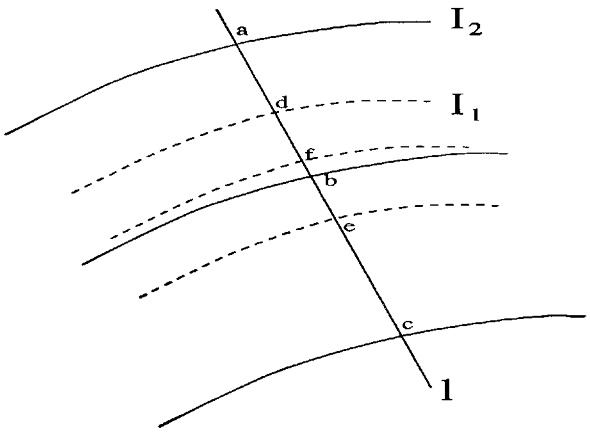

Figure 4: Included Interval Spline

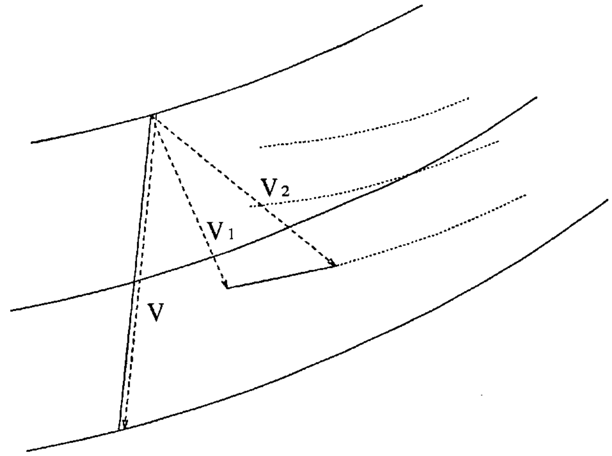

Figure 5: Two Interval Splines

- Proposing inspection strategies based on manufacturing knowledge, and
- Showing that lower-level manufacturing features such as tool paths provide a unified framework to analyze tolerances in design and manufacture of machined parts.

Acknowledgments

Samuel Drake provided important information about tolerance issues in machining.

A Interval Spline Inclusion

Given two interval splines I_1 and I_2, we want to determine if $I_1 \subset I_2$. Figure 4 gives an example where this is true, and defines our notation.

First, note that if corresponding control points of the three splines of the interval spline are collinear, then so are corresponding points on the spline curve (e.g., a, b, and c). To determine inclusion, we sample as finely as necessary across the interval I_2. At each sample point, t, we determine the line, l, passing through a,b, and c. We then intersect l with the interval I_1 and obtain d,e, and f. If d, e, and f lie between a and c, then $I_1 \subset I_2$ at t.

This requires a method to intersect a line with the interval I_1. This intersection is done with a divide and conquer algorithm, checking the sign of (while watching for 0):

$$\det \begin{vmatrix} x(t) & x_1 & x_2 \\ y(t) & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix}$$

where, $a = (x_1, y_1)$, $b = (x_2, y_2)$, and $(x(t), y(t))$, is a point on the spline. To avoid multiple solutions, the process begins around the closest points to the center point of the nominal interval with a change of sign.

However, as can be seen in Figure 5, interval splines don't necessarily begin or end at the same time. So, once the former algorithm fails (as well as the first time it succeeds), it is necessary to check that the process is actually on one ending of the common piece of curve. It is necessary to check - on the whole spline - that 2 consecutive points are on the same part of the first interval to fail. That can be done by looking at a determinant: If you consider the 2D point (x,y) to be the same as the 3D point (x,y,0), then the last coordinates of the cross products $V \otimes V_1$ and $V \otimes V_2$ will be of the same sign if and only if the 2 points at the end of V_1 and V_2 are on the same side of the plane $z = 0$. This implies that the 2D determinants $det(V, V_1)$ and $det(V, V_2)$ have the same sign.

B Algebraic Solution to Ellipse Inclusion

The implicit equation of the ellipse with center X, and which goes through the extreme points X_1 and X_2 along the 2 orthogonal axes is given by the following:

take $\vec{V_1} = \frac{(X_1 - X)}{\|X_1 - X\|^2}$ and $\vec{V_2} = \frac{(X_2 - X)}{\|X_2 - X\|^2}$ then:

$$M \in ellipse \iff (\vec{XM}.\vec{V_1})^2 + (\vec{XM}.\vec{V_2})^2 = 1$$

This is true if $V_1.V_2 = 0$, so we need to ensure that it stays the same along the path. Using the recursive algorithm $P_i := tP_i + (1-t)P_{i+1}$ it is necessary to check that

$$(tV_1^{(1)} + (1-t)V_1^{(2)}).(tV_2^{(1)} + (1-t)V_2^{(2)}) = 0$$

for any t (assuming that $V_i^{(1)}.V_i^{(2)} = 0$), which leads to

$$V_1^{(1)}.V_2^{(2)} + V_1^{(2)}.V_2^{(1)} = 0$$

This condition expresses a requirement on the orthogonal projection of $(V_1^{(2)}, V_2^{(2)})$ onto $(V_1^{(1)}, V_2^{(1)})$ which is not always true. But we will assume that the equation of the ellipse is still given by those formulas and the following.

Take the following parametric equation:

$$M(t) = X' + \frac{2t}{1+t^2}\vec{X_1'X'} + \frac{(1-t^2)}{1+t^2}\vec{X_2'X'}$$

and put this point in the implicit equation of the other ellipse. That gives the following polynomial of degree 4:

$$(X\vec{X'}.\vec{V_1} + 2tX_1\vec{X'}.\vec{V_1} + (1-t^2)X_2\vec{X'}.\vec{V_1})^2$$

$$+(X\vec{X'}.\vec{V_2} + 2tX_1\vec{X'}.\vec{V_2} + (1-t^2)X_2\vec{X'}.\vec{V_2})^2 = (1+t^2)^2$$

The real roots – if they exist – give the 4 points of intersection of those 2 ellipses. If the two ellipses do not intersect and the center of one is inside the other, then one is contained by the other one; this is checked first to avoid computation. The Sturm theorem on polynomials gives an algorithm to find the number of roots of any polynomial. If this algorithm is applied to a polynomial with symbolic variables as its coefficients, there is a condition that determines when (and only when) the polynomial has a real root. If that is done for the polynomial $X^4 + aX^2 + bX + c$, you find[1]:

$$\begin{aligned}\Gamma &= 2a^3 - 8ac + 9b^2 \\ \Delta &= 16a^4c - 4a^3b^2 - 128a^2c^2 + 144ab^2c \\ &\quad -27b^4 + 256c^3\end{aligned}$$

$X^4 + aX^2 + bX + c$ as no real roots if and only if ($a \geq 0$ and $\Delta > 0$) or ($a > 0$ and $\Gamma = 0$) or ($a < 0$ and $\Gamma > 0$ and $\Delta > 0$)

If you see the polynomial $X^4 + dX^3$ as the beginning of the expansion of $(X+\alpha)^4$ then you see that a good translation transforms any degree 4 polynomial into a polynomial $T^4 + aT^2 + bT + c$ with $T = X - \alpha$. For our problem, the resulting values of a,b and c are given by the equations:

$$\begin{aligned}A_1 &= (X\vec{X'} - X_2\vec{X'}).\vec{V_1} & B_1 &= 2X_1\vec{X'}.\vec{V_1} \\ A_2 &= (X\vec{X'} - X_2\vec{X'}).\vec{V_2} & B_2 &= 2X_1\vec{X'}.\vec{V_2} \\ C_1 &= (X\vec{X'} + X_2\vec{X'}).\vec{V_1} & C_2 &= (X\vec{X'} + X_2\vec{X'}).\vec{V_2} \\ A &= \sqrt{A_1^2 + A_2^2} & B &= \sqrt{B_1^2 + B_2^2} \\ C &= \sqrt{C_1^2 + C_2^2} & & \end{aligned}$$

then $P(t) = c_4t^4 + c_3t^3 + c_2t^2 + c_1t + c_0$ with

$$\begin{aligned}c_4 &= A^2 - 1 \qquad c_3 = 2(A_1B_1 + A_2B_2) \\ c_2 &= B^2 + 2(A_1C_1 + A_2C_2 - 1) \\ c_1 &= 2(B_1C_1 + B_2C_2) \qquad c_0 = C^2 - 1\end{aligned}$$

and finally, you find α and then a,b and c:

$$\alpha = \frac{c_3}{4c_4} \qquad a = \frac{c_2 - 6c_4\alpha^2}{c_4}$$

$$b = \frac{c_1 - 4c_4\alpha^3 - 2\alpha(c_2 - 6c_4\alpha^2)}{c_4}$$

$$c = \frac{c_0 - c_4\alpha^4 + \alpha^2(c_2 - 6c_4\alpha^2) - \alpha(c_1 - 4c_4\alpha^3)}{c_4}$$

References

[Abrams and Allen, 1992] S. Abrams and P. Allen. Sensor planning in an active robotic work cell. In *Proc. DARPA Image Understanding Workshop*, November 1992.

[Abrams *et al.*, 1993] S. Abrams, P. Allen, and K.A. Tarabanis. Dynamic sensor planning. In *Proc. DARPA Image Understanding Workshop*, November 1993.

[ANSI, 1982] American Society of Mechanical Engineers (ASME). *Engineering Drawing and Related Documentation Practices*, 1982. (Reaffirmed 1988).

[Bruderlin and Fang, 1992] B. Bruderlin and S. Fang. Intuitionistic geometry: A new approach for handling geometric robustness. *International Journal of Computational Geometry and Applications*, submitted 1992.

[Fang and Bruderlin, 1991] S. Fang and B. Bruderlin. Robustness in geometric modeling. In *Computational Geometry: Methods, Algorithms and Applications*, Bern, Switzerland, 1991. Springer-Verlag.

[Hsu and Bruderlin, 1994] C.Y. Hsu and B. Bruderlin. A new approach to tolerance analysis. Unpublished manuscript, June 1994.

[Noble and Mundy, 1993] J.A. Noble and J. Mundy. Bayesian part tolerancing with measurement error. In *Proc. SPIE Conf. – Neural and Stochastic Methods in Image & Signal Processing II*, San Diego, 1993.

[Okafor *et al.*, 1993] A.C. Okafor, Y.M. Ertekin, and S. Arora. Intelligent machining, monitoring, and diagnostic system for quality assurance based on multi-sensor integration. In *Proc. 1993 Design and Manufacturing Systems Conference*. NSF, January 1993.

[Sederberg and Farouki, 1992] T.W. Sederberg and R.T. Farouki. Approximation by interval bezier curves. *IEEE Computer Graphics and Applications*, pages 87–95, September 1992.

[Spyridi and Requicha, 1994] A.J. Spyridi and A.A.G. Requicha. Automatic programming of coordinate measuring machines. In *IEEE Robotics and Automation Conference*, pages 1107–1112, San Diego, California, May 1994. IEEE.

[Zhu *et al.*, 1993] X. Zhu, S. Fang, and B. Bruderlin. Obtaining robust boolean set operations for manifold solids by avoiding and eliminating redundancy. In *Proceedings of the Second Symposium on Solid Modeling and Applications*, 1993.

[1] result taken from the course "geométrie semie-algébrique" from Professor Coste (University of Rennes, France), DEA IMA.