2014

Development of a Geometric Model Retrieval System: A design exemplar case study

Murty Srirangam
Sonos, Inc

Srinivasan Anandan
Dassault Systemes, SolidWorks

Joshua D. Summers
Clemson University, jsummer@clemson.edu

Development of a geometric model retrieval system: a design exemplar case study

Murty Srirangam

Sonos, Inc.,
25 First Street, Suite 300,
Cambridge, MA 02141 USA
E-mail: murtysrirangam@gmail.com

Srinivasan Anandan

Dassault Systèmes, SolidWorks,
175 Wyman Street,
Waltham, MA 02451 USA
E-mail: srinivasan.anandan@3ds.com

Joshua D. Summers*

Department of Mechanical Engineering,
Clemson University,
250 Fluor Daniel Building,
Mail Stop: 0921, Clemson, SC 29634-0921 USA
E-mail: jsummer@clemson.edu
*Corresponding author

Abstract: This paper presents a case study examining design exemplar technology implemented as a search and retrieval tool fortire mould inserts. Limitations of using the geometric-based exemplar approach, such as tediousness of authoring exemplars and time complexity, are identified and addressed through a new parametric-based exemplar approach. Here, the maxima and minima are calculated based upon the specifications of the query mould insert. The design exemplar is demonstrated to be useful primarily in prototyping query mechanisms. Ultimately, customer requirements necessitated implementing the parametric approach as a dedicated software package grounded on the exemplar based prototyped query mechanism.

Keywords: feature recognition; design exemplar; retrieval; mould insert.

Reference to this paper should be made as follows: Srirangam, M., Anandan, S. and Summers, J.D. (xxxx) 'Development of a geometric model retrieval system: a design exemplar case study', *Int. J. Computer Aided Engineering and Technology*, Vol. X, No. Y, pp.000–000.

Biographical notes: Murty Srirangam is currently a Senior QA Automation Engineer for Sonos, Inc. in Cambridge, MA. He received his MS in Mechanical Engineering from Clemson University in 2007, studying geometric retrieval algorithms. In 2005, he earned his BS in Mechanical Engineering from GITAM University in India. His current interests include software engineering, algorithm development, and other computational tools to support and enhance the engineer's design capacity.

Srinivasan Anandan is currently an API Support Engineer with SolidWork where he is tasked with the challenge of defining the future needs of CAD system developers within the SolidWorks platform family. He received his PhD from Clemson University in Mechanical Engineering in 2008, studying challenges of semantic and geometric similarity retrieval to support engineering design. He earned his BS from Mumbai University in 2002 and an MS in Industrial Engineering from Clemson University in 2006. His research interests include software and design enabling technologies to support mechanical product developers.

Joshua D. Summers, Professor in Mechanical Engineering and the College IDEaS Professor at Clemson University, co-directs the CEDAR Group (Clemson Engineering Design Applications and Research). He earned his PhD in Mechanical Engineering from ASU (design automation) and his MS (submarine design) and BS (fluidised bed design) from University of Missouri. He has worked at the Naval Research Laboratory (VR Lab and NCARAI). He research has been funded by government, large industry, and small-medium sized enterprises. His areas of interest include collaborative design, knowledge management, and design enabler development with the overall objective of improving design through collaboration and computation. He has supervised over 40 MS/PhD theses, published more than 220 referenced publications, and been recognised with international, national, and regional awards for research and education excellence.

1 Introduction – role of similarity in engineering design

In engineering design, it is a common practice to develop a solution set to the problems at hand by comparing them with known 'similar' design problems and modifying them to satisfy new design requirements (Goel and Craw, 2005; Iyer et al., 2005; McAdams and Wood, 2002; Summers et al., 2002; Veltkamp and Hagedoorn, 2001). This is evident in that engineers spend significant time, between 30%–60%, searching for information that is relevant to the given design problem (Allard et al., 2009; Culley et al., 1992; Leizerowicz et al., 1996). It has also been determined that approximately 75% of the design activity consists of reusing existing data to address the design problem (Iyer et al., 2005; Ullman, 2010). Reusing design information from similar products is useful in the development of new product design for cost estimation, product platform development, and part reuse (Cardone et al., 2003). For example, in the modern computer era where the design data is maintained digitally, many companies, such as Quote (http://www.mfgquote.com/http://www.mfgquote.com/), Job Shop (http://www.jobshop.com/), and Global Spec (http://www.globalspec.com) allow clients to submit 3D models of the part they wish to produce over the web to obtain manufacturing cost quotations. The automated time to estimate the production cost of the submitted product can be greatly reduced compared manual estimations [5], if a similar previously manufactured product can be retrieved from the database and its cost is recalculated according to new specifications. Further, in manufacturing, the setup time and cost while switching between different products is considerable. Such criteria could be reduced significantly if similarly shaped parts are grouped to share common tools and setups [6]. For firms with databases of previously designed parts, designers may find it convenient and efficient to reuse database designs to create new products. In all such

Comment [t1]: Author: Please provide full reference for McWherter, 2001

Comment [t2]: Author: Please provide full reference for O. Kulak, 2006.

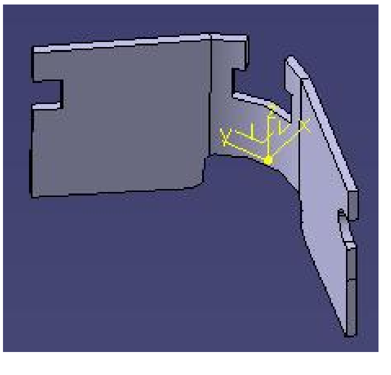
cases, discovering similar parts saves both time and money. With large amounts of data generated from the comprehensive use of CAD systems in engineering design, designers prefer to reuse the CAD data to make better use of their time and labours to create superior designs than in generating CAD models.

One technology that has been proposed in the literature is the design exemplar, which is based on feature recognition technology and graph grammars (Summers et al., 2004). This paper presents the first application of the design exemplar to an industrial problem. We present a case study that explores the true utility of this exemplar in developing a search and retrieval tool for use in tyre design as sponsored by a major tyre manufacturing company. This case study includes the challenges associated with developing an exemplar based solution to the presented problem, the evolution of this primarily geometric approach to a parametric one, and finally the instantiation of the design exemplar inspired solution in a dedicated software package. Finally, we explore the true benefits to using the design exemplar in design automation software development, specifically exploring the aspects of rapid algorithm prototyping.

2 Research motivation: tread insert retrieval

To create tyre treads, manufacturers use stamped mould inserts whose tooling cost several thousand dollars. A typical mold insert (Figure 1) is a metallic insert which can be integrated into the complex tyre mould to create the small, narrow depressions of the tyre treads. The top view of the mould insert for the mould insert (Figure 1) can be simplified to a line-arc-line profile (Figure 2). The top profile is used to define the bending tooling for creating the insert. It is this profile that is the focus of the search and retrieval system.

Figure 1 Example of mould insert (see online version for colours)



The line-arc-line profile of the mould insert may be described as a line followed by an arc which is tangential to the line. The arc in turn is followed by a line which is tangential to the arc itself. Mould inserts manufactured to create treading on any given tyre may consist up to 18 such line-arc-lines as their top profiles. The tooling cost for each type of mould insert is approximately $2,000, with three to five different inserts required for each tyre. The tooling cost of the mould inserts for a newly designed tyre is approximately $6,000 to $10,000. The company manufactures numerous types of tyres and hence the total tooling cost is significant. As the company has been manufacturing tyres for decades, a significant database of mould insert designs has evolved and their associated stamping tool sets. The tooling cost of a new mould insert can be reduced by searching and retrieving a 'similar mould insert' from this database for reuse in lieu of the new mould insert. In this case, a previously designed mould insert is considered similar to a new mould insert if it lies within a tolerance envelope around the new one. Thus, exact matches for the profiles are not sought, but rather similar matches that have the general shape and profile can be used without impacting negatively the tyre tread performance properties.

Figure 2 Line-arc-line profile of mould insert (see online version for colours)

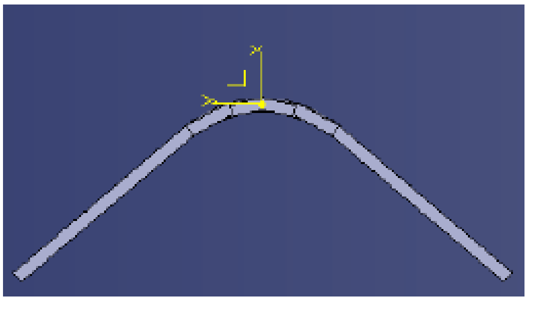
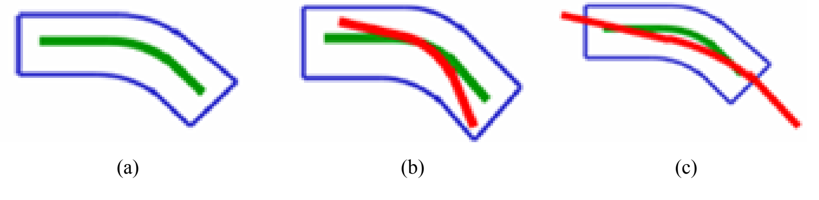


Figure 3 Lamelle with tolerance envelope (see online version for colours)



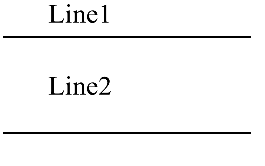
The concept of similarity based on tolerance envelope may be explained with an example. Consider a typical mould insert (green colour) as shown in Figure 3(a). This mould insert is initially designed to create treading on a newly designed tyre and is called target mould insert. To find a 'similar' mould insert for this target mould insert, a tolerance envelope is defined around the target mould insert, which is shown in blue in Figure 3. The tolerance is defined by the tyre designer. A mould insert that falls within this tolerance envelope is considered to be a mould insert similar to the target mould insert. Figure 3(b) shows a similar mould insert, shown in red, which fits in the tolerance envelope, with the original insert that defines the tolerance envelope shown in green. Figure 3(c) shows a mould insert that does not fit inside the tolerance envelope and hence is not considered similar.

Once a similar mould insert is retrieved, it can be inspected by the tyre designer to see if it can be reused, thereby saving on the tooling cost. Manually searching through the database for similar mould inserts can be time consuming because of the database consisting of more than 5,500 inserts for the North American design group of the manufacturer. As well, the designer searching the database might miss a potential candidate for reuse. Hence, the tyre manufacturer identified the need for developing an automatic search and retrieval system. A potential challenge in automating the search and retrieval process is the absence of algorithms that deal specifically such a geometric similarity. In the present case, all the mould inserts are either line-arc-line models or extended line-arc-line models which differ primarily only in dimensions. In order to search and retrieve similar mould inserts, an algorithm that can deal with geometric similarity of this kind is needed. The design exemplar is proposed as a solution to the problem and the resulting development effort is the focus of this case study.

3 The design exemplar: a CAD query language

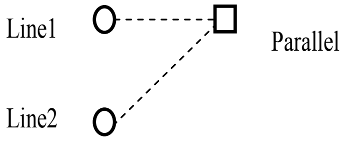
The design exemplar is a CAD query language that can be used to search and retrieve geometric models (Summers et al., 2006). Specifically it is a data structure used to represent design data, with an integral generic constraint solving algorithm that facilitates querying, solving, and modification of design data represented in a geometric model (Summers et al., 2004). It provides a standard representation of mechanical engineering design knowledge for topological and geometric design problems based on a canonically derived set of entities and relations. Finally, the design exemplar uses a bipartite graph representation to model design data, implying that two entities can be connected only through a relation. Implicit relations or entities have the attribute ‘extract’ (shown by the dotted lines), whereas relations or entities that are explicit in the CAD model have attribute ‘match’ (shown by solid lines). The match portion supports the explicit data that is being interrogated in the model by the user whereas the extract portion supports and evaluates the implicit data represented in the model. Thus, the extract component is conditional, holding the relations which should be satisfied by the match part. In this way it facilitates reasoning about the match part of the exemplar. For example, the exemplar representation of a model consisting of a pair of parallel lines shown in Figure 4 is presented in Figure 5.

Figure 4 Simple model



In this exemplar the pair of lines, Line1 and Line2, are represented with two entities. Since these lines are explicit they have the attribute ‘match’. These nodes are grouped together and are connected through the parallel relation. The edge is represented by dotted line since this relation is not explicit.

Figure 5 Exemplar for model shown in Figure 4



The modification capability of design data of the design exemplar is due to its ability to support alpha and beta states. The alpha state represents the constraints and entities that exist in the model before modification, while the beta state represents entities and relations that exist in the model after modification. Any entities existing in a model both before and after modification are said to be in an alpha-beta state.

4 Search and retrieval of mould inserts using the design exemplar

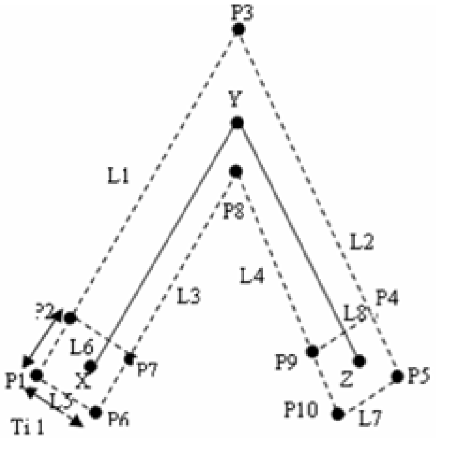
This section presents two different efforts for defining a generic query mechanism based on the design exemplar: a geometric approach and a parametric approach. These two approaches are discussed, specifically with respect to their complexity for development and their computational efficiency.

4.1 The geometric approach

In the first approach, relations are imposed on the salient vertices or points of the geometry of the mould insert to constrain it fully outside or inside a bounding geometry so as to satisfy the ‘similarity’ condition. The most common condition entails verifying if a given point lies within a particular area of the tolerance envelope or bounding geometry. This is known here as the ‘boundary constraint’ approach. The exemplar includes two components: the match and the extract. The match component is associated with the explicit information contained within the geometry models being queried, typically composed of lines, arcs, and points. The extract portion includes the relations that determine whether a point in the model being queried lies within the acceptable range. To determine if a line segment can fit within a tolerance envelope of rectangular shape, conditions can be imposed on the entities of the line segment with respect to the entities of the rectangle to form a fully constrained problem. In this example, the line segments forms the match portion exemplar of the query as conditions are imposed on it and the rectangular envelope forms the extract portion of the exemplar as line segment is constrained about it. If all the salient points or vertices of geometry satisfy the conditions imposed, then the queried model is similar to the target as defined by the exemplar.

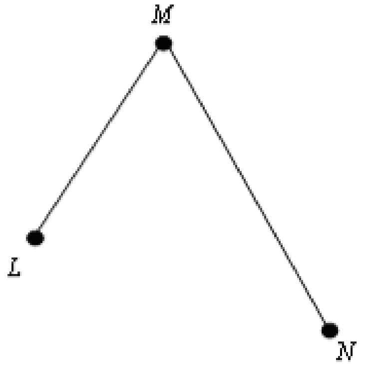
The tolerance envelope drawn around a target mould insert has two tolerance boxes; one at the beginning and one at the end. For a mould insert to be considered similar, the starting and ending points should lie within these end tolerance boxes of the envelope. In the example (Figure 6) the quadrilaterals defined by points P1, P2, P7, P6, and P9 P4 P10 P5 form the end tolerance boxes for the target mould insert XYZ.

Figure 6 Target mould insert with tolerance envelope



For a mould insert LMN (Figure 7) to be considered similar to XYZ, it should fit within the envelope of Figure 6 and its endpoints *L* and *N* should lie within these tolerance boxes. The tolerance envelope of dashed lines around XYZ is defined by two length parameters (width is the distance between P1 and P6 and length is the distance between P1 and P2).

Figure 7 Mould insert LMN



Verifying if a mould insert can fit within a tolerance envelope is possible using two different geometric methods. In the first, the tolerance is positioned around the mould insert to determine if at least one configuration exists to meet the necessary conditions. In the second, the mould inset is adjusted such that it fits within the envelope satisfying the necessary conditions. Though these methods appear similar, there is a difference in the number of constraints applied, in term impacting the solving complexity and time.

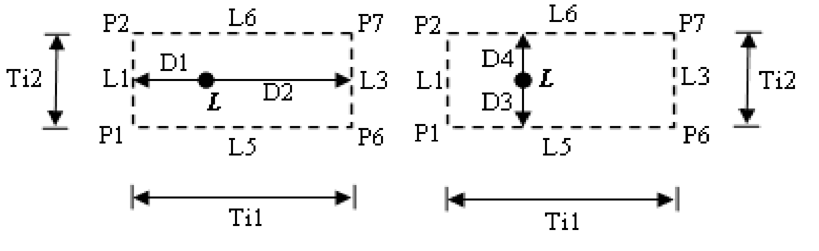
4.1.1 Geometric boundary constraint approach 1

The tolerance envelope is adjusted around the mould insert retrieved from the database to determine the existence of a configuration such that the two necessary conditions are met. Since conditions are imposed on the tolerance envelope, it forms the extract part of the query and the mould insert forms the alpha match part of the query. In order to determine whether the mold insert LMN (Figure 7) is similar to XYZ (Figure 6), the following conditions must be satisfied.

- the point *L* should be within the end tolerance box either P1 P2 P7 P6 or P4 P5 P10 P9
- the point *N* should be within the other box P1 P2 P7 P6 or P4 P5 P10 P9
- *M* should be within the envelope.

Figure 8 shows an enlarged view of the tolerance boxes of the tolerance envelope shown in Figure 6.

Figure 8 Conditions that need to be satisfied by point L



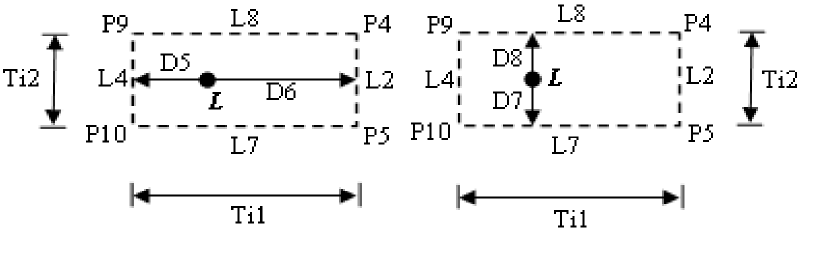
For point *L* to be within the box P1 P2 P7 P6, which is a necessary condition for the mould insert *LMN* to be similar, it should satisfy the following equations:

$$Ti1 = d1 + d2 \tag{1}$$

$$Ti2 = d3 + d4 \tag{2}$$

Also, there is a possibility that the point *L* is within the tolerance box on the other side of the tolerance envelope, which is P4 P5 P10 P9. Similar equations are formed using Figure 9.

Figure 9 Conditions that need to be satisfied by point L



Ti1 = d5 + d6 (3)

Ti2 = d7 + d8 (4)

These equations enforce the end points of the lines of the mould insert profile to be within the tolerance envelope. The lines may not be parallel to lines of the tolerance envelope. A similar set of equations are used to constrain point *N* such that it lies within the tolerance box. The equations needed to constrain point *M* within the tolerance envelope are derived using Figure 10. Referring to Figure 10, point *M* should lie between L1 and L3 or L2 and L4. If d9 and d10 are the distances of *M* from L1 and L3, and d11 and d12 are the distances of *M* from L4 and L2, then for *M* to lie within the envelope, it should satisfy at least one of the two conditions stated below:

d9 + d10 = Ti1 AND d11 $\leq$ L1 + Ti1

OR

d12 + d13 = Ti1 AND d11 $\leq$ L2 + Ti1

Figure 10 Conditions that need to be satisfied by point *M* to lie within the tolerance envelope

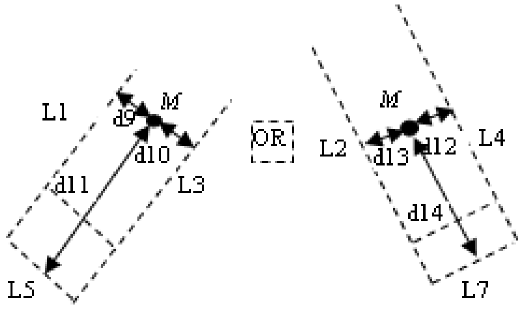
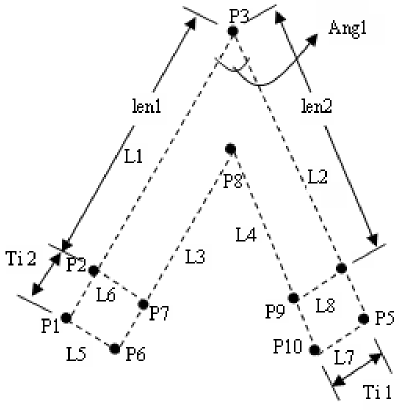


Figure 11 Tolerance envelope for mould insert LMN



The exemplar authored for the retrieval of line-line mould insert LMN (Figure 7) is shown using Figure 11.

In Figure 11, the tolerance envelope is represented by dashed lines as the extract portion of the exemplar and mould insert LMN which is verified for similarity to form the match part of the exemplar. The complete bi-partite graph for the exemplar is illustrated in Figure 12. This figure includes all the geometric and parametric entities and relations necessary to form the boundary envelop conditions for the query.

Figure 12 Exemplar authored for mold insert retrieval with envelope as extract (see online version for colours)

Alpha Match:
1. Parameter Len1;
2. Parameter Len2;
3. Line L_A;
4. Line L_B;
5. Point L;
6. Point M;
7. Point N;
8. Incident (L_A L);
9. Incident (L_A M);
10. Incident (L_B M);
12. Incident (L_B N);
13. Angle "Angl" (L_A L_B);
14. Distance "Len1" (L, M);
15. Distance "Len2" (M, N);

Alpha Extract:
16. Parameter len1 ;
17. Parameter len2;
18. Parameter Angl ;
19. Parameter Ti 1;
20. Parameter Ti 2;
21. Line " L1";
22. Line "L2";
23. Line "L3";
24. Line "L4";
25. Line "L5";
26. Line "L6";
27. Line "L7";
28. Line "L8";
29. Point "P1";
30. Point "P2";
31. Point "P3";
32. Point"P4";
33. Point "P5";
34. Point "P6";
35. Point "P7";
36. Point "P8";
37. Point "P9";
38. Point "P10;
40. Incident (L1, P3);
41. Incident (L1, P2);
42. Incident (L1, P1);
43. Distance "len1+Ti 1" (P1,P3)
44. Distance "len1" (P1, P 2);
45. Angle (Angl, L1);
46. Incident (L2, P3);
47. Incident (L2, P4);
48. Incident (L2, P5);
49. Distance "len2+Ti 2" (P3 P5);
50. Distance "len2" (P3, P4);
51. Angle (Ang2, L1);
52. Incident (L3, P6);
53. Incident (L3, P7);
54. Incident (L3, P8)
55. Distance "len11" (P6, P8);
56. Distance "len11-Tol2" (P7, P8);
57. Incident (L4, P8);
58. Incident (L4, P9);
59. Incident (L4, P10);
60. Distance "len2" (P8, P10);
61. Distance "len2" (P9, P10);
62. Parallel (L4, L2);
63. Distance "Ti1" (L1, L3);
64. Distance "Ti1" (L2, L4);
65. Parallel (L3, L1);
66. Incident (L5, P1);
67. Incident (L5, P6);
68. Perpendicular (L1, L5);
69. Incident (L6, P2);
70. Incident (L6, P7);
71. Perpendicular (L1, L6)
72. Distance "Tol2" (L6 L5);
73. Incident (L8, P4);
74. Incident (L8, P9);
75. Perpendicular (L8, L2);
76. Incident (L7, 10);
77. Incident (L7, P5);
78. Perpendicular (L7, L2);
79. Distance "Tol2" (L7, L8);
80. Distance "d1" (L, L1);
81. Distance "d2" (L, L3);
82. Distance "d3" (L, L5);
83. Distance "d4" (L, L6);
84. Distance "d5" (L, L4);
85. Distance "d6" (L, L2);
86. Distance "d7" (L, L7);
87. Distance "d8" (L, L8);
88. Distance "d9" (M, L1);
89. Distance "d10" (M, L3);
90. Distance "d11" (M, L5);
91. Distance "d12" (M, L2);
92. Distance "d13" (M, L4);
93. Distance "d14" (M, L7);
94. Distance "d15" (N L4);
95. Distance "d16" (N L2);
96. Distance "d17" (N L7);
97. Distance "d18" (N L8);
98. Distance "d19" (N L1);
99. Distance "d20" (N L3);
100. Distance "d21" (N L5);
101. Distance "d22" (N L6);
102. Equation "EQ1" (Ti 1, d1, d2);
103. Equation "EQ2" (Ti 2, d3, d4);
104. Equation "EQ3" (Ti 1, d9, d10);
105. Equation "EQ4" (d11, Len1, Tol2);
106. Equation "EQ5" (Ti 1, d13, d12);
107. Equation "EQ6 (d14, Len2, Tol2);
108. Equation "EQ7" (Ti 1, d15, d16);
109. Equation "EQ8" (Ti 2, d17, d18);
110. Equation "EQ9" (Ti 1, d5, d6);
111. Equation "EQ10" (Ti 2, d7, d8);
112. Equation "EQ11" (Ti 1, d19, d20);
113. Equation "EQ12" (Ti 2, d21, d422);

Blocks
Sub Block "Block1" (EQ1, EQ2, EQ7, EQ8);
Sub Block "Block 2" (EQ9, EQ10, EQ11, EQ12);
OR Block "Block 3" ("Block1", "Block 2");
Sub Block "Block 4" (EQ3, EQ4);
Sub Block "Block 5" (EQ5, EQ6);
OR Block "Block 6" ("Block 4", "Block 5");

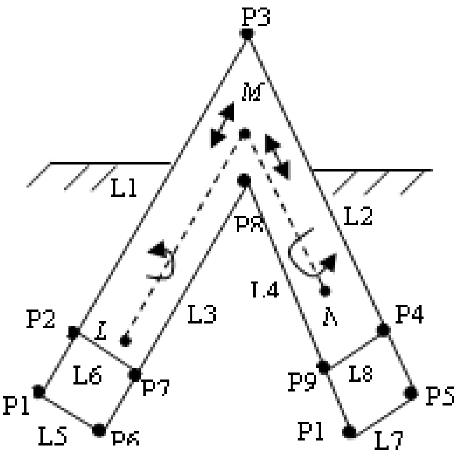
Equations:
Equation "EQ1" (Ti 1= d1 + d2);
Equation "EQ2" (Ti 2= d3 + d4);
Equation "EQ7" (Ti 1= d15+ d16);
Equation "EQ8" (Ti 2= d17 + d18);
Equation "EQ9" (Ti 1= d5+ d6);
Equation "EQ10" (Ti 2= d7 + d8);
Equation "EQ11" (Ti 1= d19+ d20);
Equation "EQ12" (Ti 2= d21 + d22);
Equation "EQ3" (Ti 1= d9 + d10);
Equation "EQ4" (d11 ≤ Len1+Ti 2);
Equation "EQ5" (Ti 1= d12 + d13);
Equation "EQ6 (d14 ≤ Len2+Ti 2);

In Figure 12, lines 1–7 are the entities that form the mould insert and lines 8–15 describe the incident, tangency, and distance constraints such that the entities form they form the mould insert. Lines 1–15 are all alpha-match, indicating that these entities must exist within the model that is being queried. Lines 16–78 describe the entities of the mould insert in the database which form the extract portion or query of the exemplar. From line 79, a set of distances are calculated. The tolerance values are specified with these equations. These equations are formed into blocks such that when a mould insert satisfies these set of blocks of equations and logical conditions, it is said to be a query match (Putti and Summers, 2006; Summers et al., 2006).

4.1.2 Geometric boundary constraint approach 2

In the next approach, the mould insert from the database is adjusted within the tolerance envelope to determine the existence of at least one configuration to meet necessary conditions (Figure 13).

Figure 13 Match and extract parts of the exemplars are interchanged in the approach 2



In Figure 13, the grounded tolerance envelope is drawn in solid lines indicating that it forms the match portion of the exemplar. The mould insert LMN is drawn in dotted lines indicating that it forms the extract part of the exemplar. The various degrees of freedom of the mould insert within the envelope are also schematically shown above. Though the conditions that must be satisfied by the mould insert remain the same, such as points *L*, *M*, *N* should satisfy the same conditions mentioned in the previous exemplar approach, with the difference now that these conditions are applied on the mould insert rather than the tolerance envelope. Here, the tolerance envelope forms the match part of the query whereas the mould insert from the database forms the extract portion. The exemplar is shown in Figure 14. Lines 1–23 describe the entities that form the envelope and lines 24–78 describe the incident, tangency, and distance constraints of the tolerance envelope. Lines 79–94 describe the entities of the mould insert in the database that form the extract part of the exemplar. From line 95, a set of distances are calculated to form a set of equations. These equations are used to verify the match.

Figure 14 Exemplar authored for mold insert retrieval with envelope as match (see online version for colours)

```
Alpha Match:
1. Parameter len1;
2. Parameter len2;
3. Parameter Ang1;
4. Parameter Tol1;
5. Parameter Tol2;
6. Line "L1";
7. Line "L2";
8. Line "L3";
9. Line "L4";
10. Line "L5";
11. Line "L6";
12. Line "L7";
13. Line "L8";
14. Point "P1";
15. Point "P2";
16. Point "P3";
17. Point "P4";
18. Point "P5";
19. Point "P6";
20. Point "P7"
21. Point "P8";
22. Point "P9";
23. Point "P10;
24. Incident (L1, P3);
25. Incident (L1, P2);
26. Incident (L1, P1)
. Distance "len1+Tol1" (P1,P 3)
28. Distance "len1" (P1, P 2);
29. Angle (Ang1, L1);
30. Incident (L2, P3);
31. Incident (L2, P4);
32. Incident (L2, P5);
Distance "len2+Tol2" (P3, P5);
34. Distance "len2" (P3, P4);
35. Angle (Ang2, L1);
36. Incident (L3, P6);
37. Incident (L3, P7);
38. Incident (L3, P8)
39. Distance "len11" (P6, P8);
Distance "len11-Tol2" (P7,P8);
41. Incident (L4, P8);
42. Incident (L4, P9);
43. Incident (L4, P10);
44. Distance "len2" (P8, P10);
45. Distance "len2" (P9, P10);
46. Angle (Ang1, L1);
47. Incident (L2, P3);
48. Incident (L2, P4);
49. Incident (L2, P5);
Distance "len2+Tol2" (P3, P5);
51. Distance "len2" (P3, P4);
52. Angle (Ang2, L1);
53. Incident (L3, P6);
54. Incident (L3, P7);
55. Incident (L3, P8)
56. Distance "len11" (P6, P8);.
Distance "len11-Tol2" (P7, P8);
57. Incident (L4, P8);
58. Incident (L4, P9);
59. Incident (L4, P10);
60. Distance "len2" (P8, P10);
61. Distance "len2" (P9, P10);
62. Parallel (L4, L2);
63. Distance "Tol1" (L1, L3);
64. Distance "Tol1" (L2, L4);
65. Parallel (L3, L1);
66. Incident (L5, P1);
67. Incident (L5, P6);
68. Perpendicular (L1, L5);
69. Incident (L6, P2);
70. Incident (L6, P7);
71. Perpendicular (L1, L6);
72. Distance "Tol2" (L6, L5);
73. Incident (L8, P4);
74. Incident (L8, P9);
75. Perpendicular (L8, L2);
76. Incident (L7, 10);
77. Incident (L7, P5);
78. Perpendicular (L7, L2);
Alpha Extract:
79. Parameter Len1;
80. Parameter Len2;
81. Line L_A;
82. Line L_B;
83. Point L;
84. Point M;
85. Point N;
86. Incident (L_A, L);
87. Incident (L_ A, M);
88. Incident (L_B, M);
89. Incident (L_B, N);
90. Distance "Len1" (L, M);
91. Distance "Len2" (M, N);
92. Angle "Ang" (L_A, L_B);
93. Distance "Len1" (L, M);
94. Distance "Len2" (M, N);
95. Distance "d1" (L, L1);
94. Distance "d2" (L, L3);
95. Distance "d3" (L, L5);
96. Distance "d4" (L, L6);
97. Distance "d5" (L, L4);
98. Distance "d6" (L, L2);
99. Distance "d7" (L, L7);
100. Distance "d8" (L, L8);
101. Distance "d9" (M, L1);
102. Distance "d10" (M, L3);
103. Distance "d11" (M, L5);
104. Distance "d12" (M, L2);
105. Distance "d13" (M, L4);
106. Distance "d14" (M, L7);
107. Distance "d15" (N, L4);
108. Distance "d16" (N, L2);
109. Distance "d17" (N, L7);
110. Distance "d18" (N, L8);
111. Distance "d19" (N, L1);
112. Distance "d20" (N, L3);
113. Distance "d21" (N, L5);
114. Distance "d22" (N, L6);
115. Equation "EQ1" (Tol1, d1 , d2);
116. Equation "EQ2" (Tol2, d3, d4);
117. Equation "EQ3" (Tol1, d9, d10);
118. Equation "EQ4" (d11, Len1, Tol2);
119. Equation "EQ5" (Tol1, d13, d12);
120. Equation "EQ6 (d14, Len2, Tol2);
121. Equation "EQ7" (Tol1, d15, d16);
122. Equation "EQ8" (Tol2, d17, d18);
123. Equation "EQ9" (Tol1, d5, d6);
124. Equation "EQ10" (Tol2, d7, d8);
125. Equation "EQ11" (Tol1, d19, d20);
126. Equation "EQ12" (Tol2, d21, d422);

Blocks
  Sub Block "Block1" (EQ1, EQ2, EQ7, EQ8);
  Sub Block "Block 2" (EQ9, EQ10, EQ11,
  EQ12);
  OR Block "Block 3" ("Block1", "Block 2");
  Sub Block "Block 4" (EQ3, EQ4);
  Sub Block "Block 5" (EQ5, EQ6);
  OR Block "Block 6"("Block 4", "Block 5");

  Equations:
    Equation "EQ1" (Tol1 = d1 + d2);
    Equation "EQ2" (Tol2= d3 + d4);
    Equation "EQ7" (Tol1 = d15+ d16);
    Equation "EQ8" (Tol2= d17 + d18);
    Equation "EQ9" (Tol1 = d5+ d6);
    Equation "EQ10" (Tol2= d7 + d8);
    Equation "EQ11" (Tol1= d19+ d20);
    Equation "EQ12" (Tol2= d21 + d22);
    Equation "EQ3" (Tol1 = d9 + d10);
    Equation "EQ4" (d11 ≤ Len1+Tol2);
    Equation "EQ5" (Tol1 = d12 + d13);
    Equation "EQ6 (d14 ≤ Len2+Tol2);
```

A comparison of exemplars authored for the two approaches shows that the second approach has fewer constraints in the extract portion, thus reducing the computational effort required of the constraint solver. The tolerance envelope forms the extract portion of the exemplar in approach 1; eight lines and ten points of the tolerance envelope are constrained around three points and two lines of the mould insert which is retrieved from the database. In approach 2, the mould insert forms the extract part of the exemplar query; three points and two lines of the mould insert retrieved from the database are constrained within eight lines and ten points of the tolerance envelope. Hence, the constraint solver requires more time to check for the similarity condition in approach 1 when compared to approach 2.

4.1.3 Retrieval of line-arc-line mould inserts

The approach discussed for line-line mould inserts is extended for the retrieval of line-arc-line mould inserts. Apart for the curve introduced between the two lines, the example discussed in the previous section has similar geometric characteristics to the line-arc-line profile. Hence, most of the conditions that must be applied to constrain a line-arc-line mould insert within a line-arc-line tolerance envelope remain the same. Figure 15 shows a mould insert PQRS of line-arc-line profile that must be verified as being within the tolerance envelope shown in Figure 16.

Figure 15 Line-arc-line mould insert PQRS

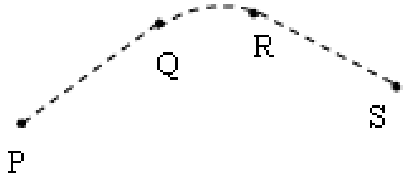
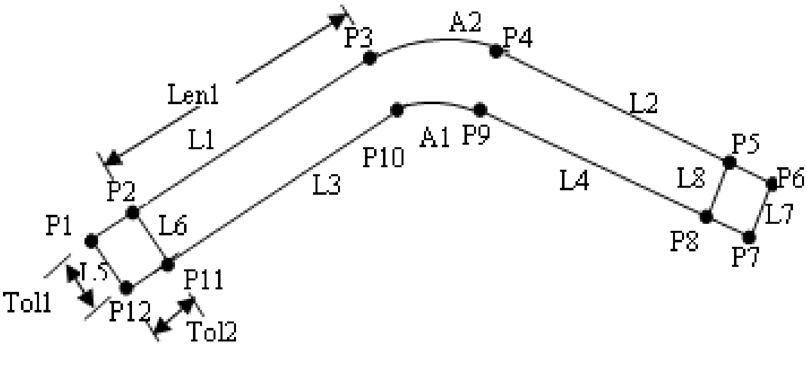


Figure 16 Tolerance envelope from target mould insert



To be considered as a similar mould insert with reference to the envelope shown in Figure 16, PQRS should satisfy the following conditions:

- point *P* should lie within the end tolerance box P1 P2 P11 P12
- point *S* should lie within the end tolerance box P5 P6 P7 P8
- points *Q* and *R* should lie within the tolerance envelope.

The exemplar for retrieving line-arc-line mould inserts can be authored similarly to the exemplars authored for the line-line models with additional arc entities. The exemplar consists of 40 entities and 80 relations (Figure 17).

Figure 17 Exemplar authored for line arc line mold insert retrieval (see online version for colours)

```
Alpha Match:
1. Parameter Len1;
2. Parameter Len2;
3. Parameter Ang;
4. Parameter Tol1;
5. Parameter Tol2;
6. Parameter rad;
7. Line "L1";
8. Line "L2";
9. Line "L3";
10. Line "L4";
11. Line "L5";
12. Line "L6";
13. Line "L7";
14. Line "L8";
15. Line CP3;
16. Line CP4;
17. Point "P1";
18. Point "P2";
19. Point "P3";
20. Point "P4";
21. Point "P5";
22. Point "P6"
23. Point "P8";
24. Point "P9";
25. Point "P10;
26. Point "P11";
27. Point "P12";
28. Point P;
29. Point Q;
30. Point R;
31. Point S;
32. Point C;
33. Arc "A1";
34. Arc "A2";
35. Tangency (A1, L1);
36. Tangency (A1, L2);
37. Tangency (A2, L4);
38. Tangency (A2, L3);
39. Incident (L1, P3);
40. Incident (L1, P2);
41. Incident (L1, P1);
Distance "len1+Tol1" (P1, P 3);
42. Distance "Tol1" (P1, P 2);
43. A1 (Rad1+Tol2, Ang1);
44. Incident (A1, P3);
45. Incident (A1, P4);
46. A2 (Rad1+Tol2, Ang1);
47. Incident (L2, P6);
48. Incident (L2, P4);
49. Incident (L2, P5);
. Distance "len2+Tol2" (P4, P 6);
51. Distance "len2" (P4, P 5);
52. Incident (L4, P9);
53. Incident (L4, P7);
54. Incident (L4, P8);
Distance "len2+Tol2" (P7, P 9);
55. Distance "len2" (P8, P 9);
56. Parallel (L4, L2);
57. Incident (L7, P6);
58. Incident (L7, P7);
59. Perpendicular (L7, L2);
60. Perpendicular (L7, L4);
61. Incident (L8, P8);
62. Incident (L8, P5);
63. Parallel (L8, L7);
64. Concentric (A1, A2);
65. Incident (L3, P 10);
66. Incident (L3, P11);
67. Incident (L3, P12);
68. Distance "len2+Tol2" (P10, P 12);
69. Distance "len2" (P11, P 10);
70. Parallel (L3, L1);
71. Incident (L5, P1);
72. Incident (L5, P12);
73. Perpendicular (L7, L1);
74. Perpendicular (L7, L3);
75. Incident (L6, 2);
76. Incident (L8, 11);
77. Parallel (L8, L7);
Alpha Extract
78. Point P;
79. Point Q;
80. Point R;
81. Point S;
82. Line Li1;
83. Line Li2;
84. Incident (P, Li1);
85. Incident (Q, Li1);
86. Incident (Q, Ar2);
87. Incident (R, Ar2);
88. Incident (R, Li2);
89. Equation "EQ1" (Tol1= d1 +d2);
90. Equation "EQ2" (Tol2=d3+d4);
91. Equation "EQ3" (Tol1= d5 +d6);
92. Equation "EQ4" (Tol2=d7+d8);
93. Equation "EQ5" (Tol1= d22+d23);
94. Equation "EQ6" (Tol2, d2, d25);
95. Equation "EQ7" (Tol1, d26, d27);
96. Equation "EQ8" (Tol2 d28, d29);
97. Equation "EQ9" (Tol1, d9, d10);
98. Distance "d1" (P, L1);
99. Distance "d2" (P, L2);
100. Distance "d3" (P, L5);
101. Distance "d4" (P, L6);
102. Distance "d5" (S, L4);
103. Distance "d6" (S, L2);
104. Distance "d7" (S, L8);
105 Distance "d8" (S, L7);
106. Distance "d9" (Q, L1);
107. Distance "d10" (Q, L3);
108. Distance "d11" (Q, L5);
109. Distance "d12" (Q, L2);
110. Distance "d13" (Q, L4);
111. Distance "d14" (Q, L7);
112. Distance "d15" (Q, C);
113. Distance "d16" (R, L1);
114. Distance "d17" (R, L3);
115. Distance "d18" (R, L5);
116. Distance "d19" (R, L2);
117. Distance "d20" (R, L4);
119. Distance "d21" (Q, L7);
120. Distance "d22" (Q, C);
121. Equation "EQ10" (d1, L1, Ti1);
122. Equation "EQ11" (Tol1, d12, d13);
123. Equation "EQ12" (d14, L1, Ti1);
124 . Equation "EQ13" (Ang, Ang1, Ang2)
125. Equation "EQ14" (rad, Ti1, d15);
126. Equation "EQ15" (Tol1, d16, d17);
127. Equation "EQ16" (d18, L1, Ti1);
128. Equation "EQ17" (Tol1, d19, d20);
129 Equation "EQ18" (d21, L1, Ti1);
130 Equation "EQ19" (Ang, Ang3, Ang4);
131 Equation "EQ20" ((rad, T, d30);
132. Equation "EQ6" (Tol2, d2, d25);
133. Equation "EQ7" (Tol1, d26 d27);
134. Equation "EQ8" (Tol2 d28, d29);
135. Equation "EQ9" (Tol1, d9, d10);
136 Equation "EQ10" (d1, L1, Ti1);
137. Equation "EQ11" (Tol1, d12, d13);
138. Equation "EQ12" (d14, L1, Ti1);
139. Equation "EQ13" (Ang, Ang1, Ang2);
140. Equation "EQ14" (rad, Ti1, d15);
141. Equation "EQ15" (Tol1, d16, d17);
142 Equation "EQ16" (d18, L1, Ti1);
143. Equation "EQ17" (Tol1, d19, d20);
144. Equation "EQ18" (d21, L1, Ti1);
145. Equation "EQ19" (Ang, Ang3, Ang4);
146. Equation "EQ20" ((rad, T, d30);

Blocks:
    Sub Block "Block1" (EQ1, EQ2, EQ3, EQ4);
    Sub Block "Block2" (EQ5, EQ6, EQ7, EQ8);
    OR Block ("Block1", "Block2");
    Sub Block "Block3" (EQ9, EQ10);
    Sub Block "Block4" (EQ11, EQ12);
    Sub Block "Block5" (EQ13, EQ14);
    OR Block ("Block3", "Block4", "Block5");
    Sub Block "Block6" (EQ15, EQ16);
    Sub Block "Block7" (EQ17, EQ18);
    Sub Block "Block8" (EQ19, EQ20);
    OR Block ("Block6", "Block7", "Block8");

Equations:
  Equation "EQ1" (Tol1= d1 +d2);
  Equation "EQ2" (Tol2=d3+d4);
  Equation "EQ3" (Tol1= d5 +d6);
  Equation "EQ4" (Tol2=d7+d8);
  Equation "EQ5" (Tol1= d22+d23);
  Equation "EQ6" (Tol2= d2+ d25 );
  Equation "EQ7" (Tol1= d26+d27);
  Equation "EQ8" (Tol2= d28+d29);
  Equation "EQ9" (Tol1= d9+d10);
  Equation "EQ10" (d11≤= L1+Ti1);
  Equation "EQ11" (Tol1= d12+d13);
  Equation "EQ12" (d14≤= L1+Ti1);
  Equation "EQ11" (Ang= Ang1+Ang2);
  Equation "EQ13" ((rad +Ti)≤d15 ≤(rad-Ti1));
  Equation "EQ14" (Tol1= d16+d17);
  Equation "EQ15" (d18≤= L1+Ti1);
  Equation "EQ16" (Tol1= d19+d20);
  Equation "EQ17" (d21≤= L1+Ti1);
  Equation "EQ18" (Ang= Ang3+Ang4);
  Equation "EQ19" ((rad +Ti)≤d30 ≤(rad-Ti1));
```

4.1.4 Limitations of the exemplar geometric approach

There are several limitations with these two approaches. First, the anecdotal experience of developing exemplars for the retrieval of mould inserts for profiles line-line and line-arc-line exposes the tedious nature of authoring exemplars. The exemplar built for the retrieval of a simple line-arc-line profile consists of 40 entities and 80 geometric constraints. Therefore, to author an exemplar to obtain a generic solution for mould insert retrieval problem, 18 different exemplars must be authored, meaning that the number of geometric entities and constraints to be handled is of the order of thousands. The complexity of exemplar construction also means the evidence of errors in authoring the exemplars (Summers, 2005; Summers et al., 2009).

A second observation is that a typical mould insert can have up to 18 line-arc-line profiles. Though the exemplars authored work only for a line-arc-line profile, they did not so for mould inserts with a different profile. Therefore, to author an exemplar for the retrieval of similar mould inserts of all profiles, different exemplars must be written for each increment of line-arc-line. Thus, with each increment of arc and line in the profile of the mould insert, there is a significant increase in the number of entities that must be handled. Further, all the exemplars written for different profiles of mould inserts must be sequenced to obtain a general solution for mould insert retrieval (Putti and Summers, 2006). Worse-case time consuming scenarios require checking the mould insert 18 times with 18 different exemplars in step one, which is then checked for the constraints applied.

Since the design exemplar has the limitations mentioned above, a parametric-based exemplar has been developed to address the search and retrieval problem of similar mould inserts. The goal is to reduce the degree of freedom associated with the design exemplars, and thereby reduce the computational complexity of the constraint problems.

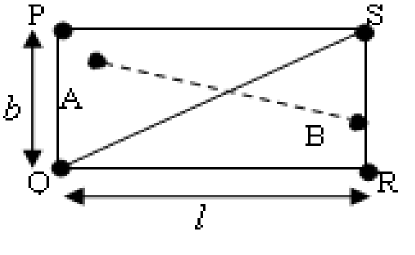
4.2 The parametric approach

To address the limitation of tediousness of authoring exemplars in Sections 4.1.1 and 4.1.2, a 'max-min' exemplar approach was developed. The principle behind this approach is illustrated with an example. Consider a line segment AB of certain length (Figure 18) and a tolerance envelope P Q S R of rectangular shape (Figure 19). To determine if the line fits within the rectangular tolerance envelope, the minimum length of the line within the envelope is zero and the maximum length of the line within the envelope is the diagonal of the rectangle, QS (or PR) which is given by the formula $\sqrt{l^2 + b^2}$.

Comment [t3]: Author: Please confirm if this is correct.

Figure 18 Model

Figure 19 Rectangular envelope



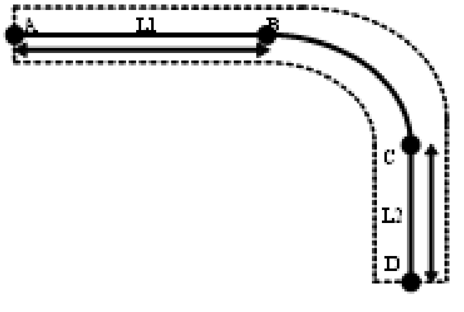
This approach entails determining the length of the longest line that can fit within the rectangle in terms of known quantities, which is the length and breadth of the rectangle. A similar approach is used for retrieving mould inserts. A set of equality and inequality relations that classify a mold insert are incorporated into a query exemplar. To develop this parametric-based exemplar algorithm, one must:

- divide the target mould insert and the tolerance envelope around it into line-arc-line entities
- find maximum and minima of the parameters that fit within individual tolerance envelopes obtained by division of the whole tolerance envelope
- determine the similarity of a mould insert by verifying if all the specifications of the mould insert are within the maxima and minima calculated.

4.2.1 Step 1: division of mould inserts into line-arc-line profiles

Before developing expressions for maxima and minima for mould inserts, it is necessary to ensure that the expressions are not specific to a certain configuration. If different configurations of mould inserts formed by each increment of geometric entities arc and line, such as line-arc-line or line-arc-line-arc-line, require different formulae, then the development of the mathematical model becomes tedious and difficult. Hence, it is necessary to perform a degree of freedom analysis on the geometry of the mould inserts to verify if general expressions can be developed for extension to all mould inserts.

Figure 20 Line-arc-line model of a mould insert



A simple mould insert of line-arc-line profile is shown in Figure 20. Geometric entities, such as points, lines, and arcs, form the insert; as is a tolerance window that encompasses the insert. The mould inserts that fall within this tolerance envelope are considered to be similar to this mold insert. Table 1 shows the total degrees of freedom associated with the entities of the line-arc-line profile. The mould insert consists of entities such as point A, point B, point C, and point D, each of which has two degrees of freedom (in 2D); translation along the x-axis and translation along the y-axis. Similarly, each line has three degrees of freedom and translation along the axes and the angle. Each arc has seven degrees of freedom; x and y coordinates of the center, start, and end points and the radius. The degrees of freedom of all the entities contained in the model are 21.

Table 1 Degree of freedom analysis for line –arc-line mould insert

Entities	*DOF*	*Description*
Point A	2	X and Y coordinates
Point B	2	X and Y coordinates
Point C	2	X and Y coordinates
Point D	2	X and Y coordinates
Line 1 (L1)	3	X and Y coordinates, angle of line
Line 2 (L2)	3	X and Y coordinates, angle of line
Arc	7	X and Y coordinates of centre, radius and end points
Total	21	

These degrees of freedom are constrained through the set of relations. Table 2 shows the constraints that must be satisfied by these entities to form a line-arc-line profile. The number of scalar equations represented by the angle, radius, and distance relations is two since each relation specifies a maximum and a minimum value that each parameter takes. Thus, these constraints arrest 18 of the 21 degrees of freedom (DOF) of the entities. Therefore, because a mould insert of a line-arc-line profile, has three degrees of freedom in the space, it is considered to be a rigid body. The three degrees of freedom associated with the mould insert in a tolerance envelope are the

1 rotation of the mould insert in the two dimensional space within the envelope

2 the translation of the moUld insert within the envelope in horizontal and vertical directions.

A similar DOF analysis is done for other profiles.

Table 2 Degrees of freedom that can be arrested

Constraints	*DOF*	*Description*
Incident (A, L1)	2	Constraints the x and y coordinates
Incident (D, L2)	2	Constraints the x and y coordinates
Distance (A, D)	2	Constraints the distance between A and D in a range (lower limit < L(A, D) < upper limit)
Distance (AB + BC + CD)	2	Constraints the total length of the profile in a range (lower limit < L(AB+BC+CD) < upper limit)
Incident (Arc, B, L1)	2	Constraints the x and y coordinates
Incident (Arc, C, L2)	2	Constraints the x and y coordinates
Angle (L1, L2)	2	Constraints the angle between L1 and L2 in a range (lower limit < angle(A, D) < upper limit)
Tangent (arc and L1, at B)	1	
Tangent (arc and L2, at C)	1	
Radius of arc	2	Constraints the radius of the arc in a range
		(lower limit < rad(arc) < upper limit)
Total	18	

The degrees of freedom for the line-arc-line-arc-line mould insert were 35. The number of degrees of freedom controlled for the line-arc-line-arc-line model was 32, a difference of three. Thus, the line-arc-line-arc-line insert may also be considered a rigid body with translation in the x and y directions and rotation about z. Therefore, for any mould insert, the number of degrees of freedom associated with the mould insert within a tolerance envelope is three. As no extra degrees of freedom are present for the mould insert presented in a two dimensional space within the tolerance envelope, the formulae developed for a simple mould insert with a line-arc-line will be extended to all the mold inserts. Following the principle of the parametric model, these three DOF can be arrested by calculating the maxima and minima of the known parameters of the mold insert: the radius of the mould inserts, the angle between the legs of the mould insert, and the lengths of the legs of the mould insert.

Therefore, to assess if a line-arc-line mould insert from the database can fit within the tolerance envelope, the maximum and minimum values of the radius of the arc, the angle between the legs, and the length of the legs are needed. Also, the requirement that the end points of the legs lie within end tolerance boxes constrains the maximum and minimum distance between these end points. The specifications of the mould insert are then checked to determine if they fall within the numerical bounds. If all the specifications fall within calculated range, the mould insert is considered similar. As stated before, since the mould insert of any general line-arc-line profile has three degrees of freedom, the formulae developed for a line-arc-line profile are easily extendable to the remaining profiles.

A simple observation suggests that all the mould inserts consist of line-arc-line profile as a fundamental element. Therefore, if a mould insert is divided into line-arc-line elements, the formulae can be repeated sequentially to handle the next – arc-line elements. To apply the approach to a target mould insert with a profile of multiple line-arc-lines, a tolerance envelope is drawn. The tolerance envelope is then divided into line-arc-line profiles, and the maxima and minima of the parameters that fit into each of these tolerance envelopes with line-arc-line profiles are calculated. To determine if a mould insert fits within the whole tolerance envelope, the specifications of the mould insert are checked to determine if they fall within the maxima and minima calculated for each individual tolerance envelopes.

Figure 21 A complex mould insert divided into line-arc-line profiles (see online version for colours)

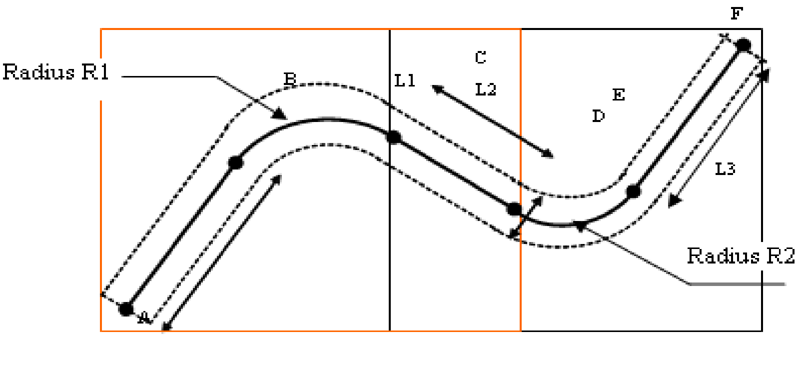


Figure 21 shows a mould insert with a line-arc-line-arc-line profile. To apply the formulae developed for a line-arc-line profile, the mould insert is divided into two line-arc-line profiles. Then the formula is applied to each line-arc-line profile separately to obtain their maxima and minima.

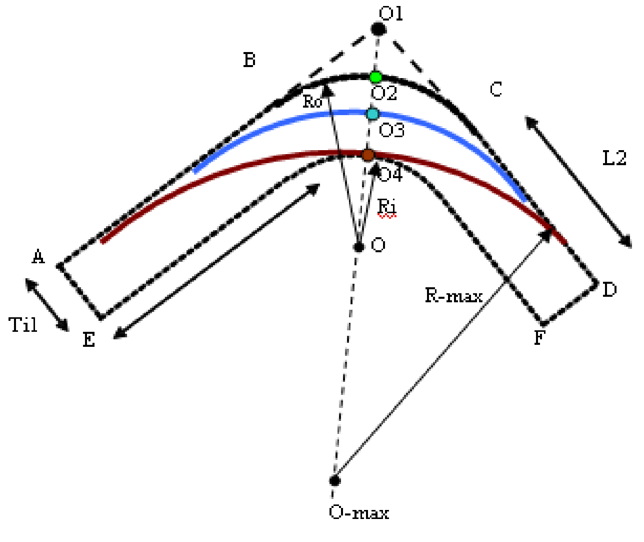
4.2.2 Step 2: calculation of the parameters

Next, the formulae to estimate maximum and minimum radius of the arc, the maximum and minimum distance between the end points of the leg, and maximum and minimum lengths of the leg that can fit with a given tolerance envelope are developed in terms of known parameters of the mould insert. Again, the known parameters are tolerance values, the length of the leg, and the radius of the arc of the target mould insert.

4.2.2.1 Expression for maximum radius of the circle that can fit in the tolerance envelope

To determine the formula for the maximum radius of the arc that fits within a given tolerance envelope, the configuration representing the maximum arc radius must be determined. Consider a tolerance envelope ABCDEF as shown in Figure 22. At a given point on the axis O1-Omax of the tolerance envelope, the arc with the greatest radius is that which passes through the point O4 and is tangential to the lines of the outer boundary of the envelope. The arc with the maximum radius is that which is tangent to the either of the legs of the outer boundary of the envelope and is also tangent to the arc of the inner boundary of the tolerance envelope.

Figure 22 Configuration of the arc of the maximum radius that can fit within a tolerance envelope (see online version for colours)



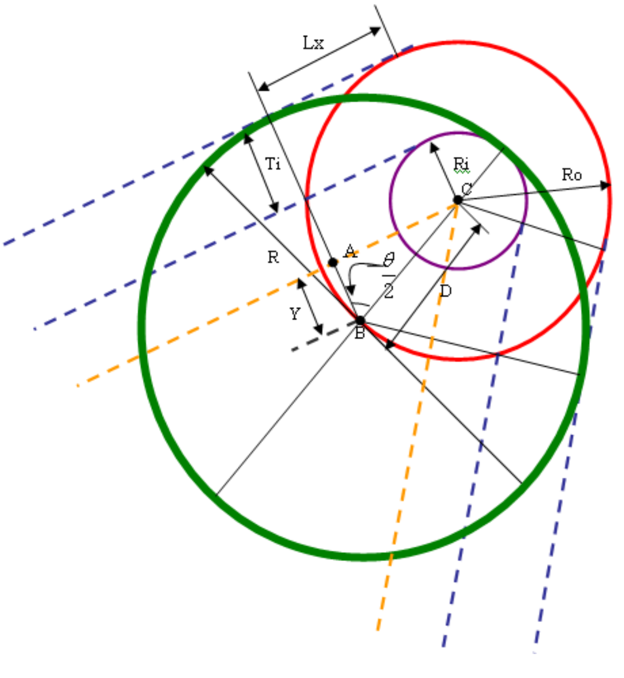
In Figure 23, let Ro and Ri be the radii of the outer and inner arcs of the tolerance envelope, shown in red and purple colours respectively. The legs of the tolerance envelope are represented by blue dashed lines. The green circle with the radius R that is tangential to the legs of the outer envelope and the arc of the inner envelope represents the circle with the largest radius that can fit within the given tolerance envelope. From Figure 23 it can be seen that,

$$R = D + Ri = Ri + Ti + Y \tag{5}$$

where R is the radius of the green circle and Ti is the value of the primary tolerance. Using trigonometry it can be seen that,

$$R = Ri + Ti + Lx / \tan(\theta / 2)$$

Figure 23 Configuration of the circle with maximum radius that can fit within a tolerance envelope



where *Lx* is the distance between the point where the green circle is tangent to the outer tolerance boundary and the point where the line is tangent to the outer curve. This implies that,

$$R = \sqrt{\left(Lx\left(\tan\left(\frac{\theta}{2}\right)\right)^{-1}\right)^2 + Lx^2} + Ri = Lx * \sqrt{\left(\tan\frac{\theta}{2}\right)^{-2} + 1} + Ri$$

From equation (5), however, we have,

R = Ri + Ti + Y;

This implies that,

$$Lx \frac{Ti}{\sqrt{(\tan(\frac{\theta}{2}))^{-2}+1}-\tan(\frac{\theta}{2})^{-1}}$$

Therefore,

$$R = \frac{Ti * \left(\tan\left(\frac{\theta}{2}\right)\right)^{-1}}{\sqrt{\left(\tan\frac{\theta}{2}\right)^{-2}+1}-\left(\tan\frac{\theta}{2}\right)^{-1}} + Ti + Ri$$

Thus the maximum radius of the circle is dependent upon the radius of the inner tolerance envelope and the angle between the lines.

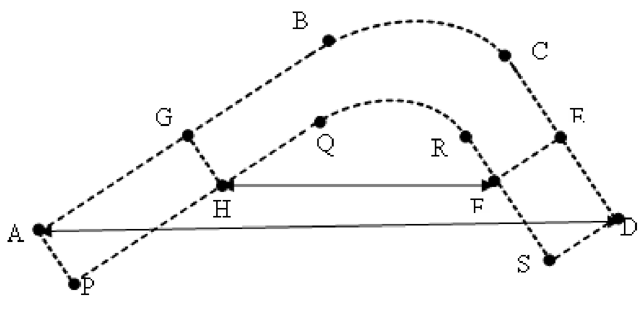
4.2.2.2 Expression to calculate minimum radius of the circle that can fit in the tolerance envelope

Theoretically, the minimum radius of the circle that can fit within the tolerance envelope is zero. In the case of mould inserts, however, the minimum radius of the arc is determined by the stamping process limitations. For a tolerance envelope of tolerance Ti, the minimum radius of the arc that can be manufactured is Ti/2.

4.2.2.3 Expression to find maximum and minimum distance between end points of the legs

The next DOF to be arrested is the angle between the lines. Since the angle between the lines is directly related to the distance between the end points of the mould insert, the maximum and minimum distance between the end points of the legs of the mould insert that can fit within tolerance envelope arrests this second degree of freedom (Figure 24).

Figure 24 Maximum and minimum distance between end points of a similar mould insert



The possible maximum and minimum distances between two points that either lie on or within the end tolerance boxes is the maximum and minimum distance between the endpoints of the mould insert that can fit within the tolerance envelope. Similar to the derivation of the expression shown above, the expression for the maximum distance between the two legs (D) was derived using trigonometry.

$$\text{Dmax} = 2\text{L1} * \text{cosine}\ \alpha_1 + 2\text{R1} * \text{sine}\ \alpha_1;$$

where R1 = radius of included arc; α_1 = angle subtended by R1; β_1 = angle made by the legs with each other; L1 = length of first leg. Similarly the expression for the minimum distance between the two legs is given by:

$$\text{Dmin} = (\text{L1} - \text{Ti}) * \cos\alpha_1 + (\text{L2} - \text{Ti}) * \cos\alpha_2 + (\text{R} - \text{Ti}) * \sin\ \alpha_2$$

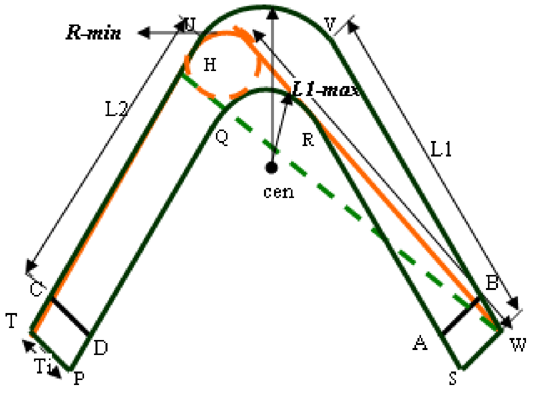
4.2.2.4 Expression for the maximum length of the leg that can fit within the tolerance envelope:

Because the insufficient number of variables makes it impossible to determine a formula for the tangent of maximum length, a general procedure is presented here:

1 assigning a local coordinate system to the envelope

2 finding the maximum length of the tangent that can fit within the envelope.

Figure 25 Configuration of a line-arc-lime model such that the length of the line is maximum (see online version for colours)



First, as the objective is to find the length that is an absolute quantity and does not depend on the coordinate system, a local coordinate system is assigned to the mould insert. For convenience, the centre of the mould insert that defines the tolerance envelope is fixed as the origin. Next, the configuration providing the maximum length of the tangent is considered (Figure 25). The leg begins from W and extends to the outer boundary of the tolerance envelope such that it is tangent to the arc of the inner boundary of envelope. If L2 is greater than L1, then the leg starts from the other corner of the outer boundary. For the leg to be of maximum length, the radius of the arc should be as small as possible. The minimum possible radius of the arc that can be manufactured for an envelope of a given tolerance value Ti is Ti/2 which is fixed by the manufacturing limitations (Section 4.2.2.2). Calculating the length of the tangent requires.

Comment [t4]: Author: Please confirm if this is correct.

- finding the locus of the centre of the circle
- finding the centre of the arc of the configuration
- solving the equations of the circle and tangent to find the point of tangency
- calculating the length of the leg using the Euclidian distance formula.

4.3 *Testing results*

To test algorithm functionality, 15 mould inserts were selected at random from the database. The database was then searched for similar mould inserts for each of the randomly selected using the developed algorithms. The results were then verified visually for false positives (matches that were not correct). The experiments were conducted on a computer with an Intel Pentium M 1.73 GHz processor and 512 MB of RAM. The operating system was Windows XP. The code was implemented in C++ and compiled using Intel® C++ Compiler.

4.3.1 *An example query*

A mould insert, taken from the database, was queried for similar mould inserts, and the algorithm results were then cross checked using CAD models. CAD models of the tolerance envelope of the target mould insert and CAD models of the retrieved mould inserts were then drawn and superimposed to determine if the CAD model fits within the tolerance envelope to satisfy the rest of the conditions.

Query mould insert considered: mould insert name in database: ACR13431

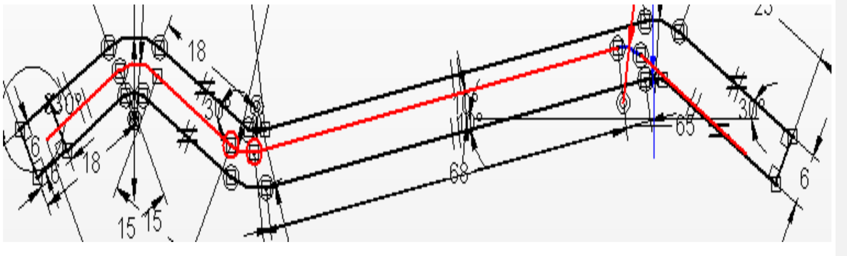
Description of the mould insert: 4 1.500001 329.999997 0.600002 1.500000 30.000090 0.599994 6.499998 9.999983 0.599994 2.000021 330.000353 0.

Figure 26 Program output

```
Enter the number of legs4
Enter the lengths of legs and their corresponding angles and the radius 1.5
330
1.5
30
6.5
10
2
30
radii.6
.6
.6
lamelle that fits in is ACR13432
lamelle that fits in is ACR13431
lamelle that fits in is ACR13428
lamelle that fits in is ACR13427
Press any key to continue_
```

The description of the mould insert was a proprietary format used by the tyre manufacturer to describe the inserts. Fouris the number of legs, 1.5 is the length of leg, 319.99 is the angle made by the leg with positive x-axis, 0.6 is the radius of the arc the leg is tangent to, and the length of the next leg is 1.5 and so on. Thus, the order of the description is the number of legs of the mould insert, the length of the leg, the slope of the leg, and the radius of the arc to which it is tangential. The results obtained by the implementation of the algorithms are shown in Figure 26 and Figure 27.

Figure 27 Verification of the retrieved mould insert ACR13432 using CAD model (see online version for colours)



4.3.2 Interpretation of results

ACR13431 is the target mould insert itself, and ACR13432, ACR13428 are mould inserts rotated through 180 degrees (flipped). ACR13427 is an exact match for the query in the database. This suggests that four inserts could be manufactured and classified as one mould insert.

4.4 Advantages and disadvantages of the parametric approach

One major advantage of the parametric approach is the low complexity of the algorithm, on the order O (n). When compared to manual retrieval, which takes several days to navigate through the database to retrieve similar mould inserts as reported by the mould designers, the algorithm is quite rapid. While it may be implemented using design exemplar tool, there is a large time gain when implemented in a standalone C++ program. This gain is due to the elimination of the requirement in handling spatial data in the C++ program. Implementing the exemplar in the design exemplar tool requires dedicating much of the computer’s processing time to applying the incident and tangential constraints on the geometric entities to form a line-arc-line profile. The primary task of the algorithm, however, entails verifying if the mould insert in the database satisfies the set of developed conditions rather than building a mould insert from line-arc-line entities. Totally eliminating need for a line-arc-line profile in the hard coded program yields a huge time gain. Also, the rotational and translational invariance of the algorithm permits identifying and retrieving mirror images. The parametric approach is limited, however, in that it provides only rough definitions of similarity without a ranking of the matches. Though a mould insert with both the largest arc radius and largest length of the leg will not fit within the tolerance envelope, the algorithm nonetheless indicates its suitability for a similar mould insert. While results are satisfying at low tolerance values, an increase in tolerance values causes a corresponding increase in the number of false positives.

5 Conclusions and future work

In case study, the applicability and suitability of the design exemplar tool for an industrial scenario was investigated with the search and retrieval of mould inserts. The core research question is: *Can design exemplar be implemented to find potential candidates of mould inserts for a given target mould insert?* The limitation of the exemplar approach is the tediousness of authoring exemplars for real world problems. Another difficulty involves managing the geometric entities while authoring exemplars. As the complexity of the exemplar increases in terms of the number of conditions imposed and geometric entities, it becomes difficult to handle all entities and relations concurrently. In the max-min approach, a set of maxima and minima conditions are calculated based on the specifications of the target mould insert. Though this approach yielded favourable results, its implementation with the design exemplar added an unreasonably high time complexity. The true utility of the design exemplar was its enhancement of prototyping the query mechanism. The capability of the design exemplar to query CAD models for parametric information made it possible to test the validity of the proposed approach. The time involved, however, in using the design exemplar for the actual database queries of more than 5,500 mould inserts inspired the implementation of this approach into an independent software package.

The following additions to the present system are suggested to enhance its capabilities:

- An automatic exemplar generator that generates exemplars of the geometric models either when selected or dropped down into CAD system can ease the massive load of generating exemplars. This feature makes the design exemplar more user friendly and also saves much time when building queries. Versions of this concept have been explored in previous work (Summers et al., 2002; Venkataraman et al., 2001).
- Methods to improve the time complexity of exemplars when applied as a search and retrieval tool on huge databases.
- Integration of the design exemplar into commercially available CAD systems to provide CAD users with a superior CAD query language customised to their needs.

References

Allard, S., Levine, K.J. and Tenopir, C. (2009) 'Design engineers and technical professionals at work: observing information usage in the workplace', *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 3, pp.443–454, doi:10.1002/asi.21004.

Cardone, A., Gupta, S.K. and Karnik, M. (2003) 'A survey of shape similarity assessment algorithms for product design and manufacturing applications', *Journal of Computing and Information Science in Engineering*, Vol. 3, No. 2, p.109, doi:10.1115/1.1577356.

Culley, S.J., Court, A.W. and McMahon, C.A. (1992) 'The information requirements of engineering designers', *Engineering Designer*, Vol. 18, No. 3, pp.21–23.

Goel, A.K. and Craw, S. (2005) 'Design, innovation, and case based reasoning', *Knowledge Engineering Review*, Vol. 20, No. 3, pp.271–276.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K. (2005) 'Three-dimensional shape searching: state-of-the-art review and future trends', *Computer-Aided Design*, Vol. 37, pp.509–530.

Leizerowicz, W., Bilgic, T., Lin, J. and Fox, M.S. (1996) 'Collaborative design using WWW', *Proceedings of WET-ICE*, IEEE, Morgantown, WV.

McAdams, D. and Wood, K. (2002) 'A quantiative similarity metric for design-by-analogy', *ASME Transactions Journal of Mechanical Design*, Vol. 124, No. 2, pp.173–182.

Putti, S. and Summers, J.D. (2006) 'Dynamic networks: towards a mechanical design visual programming language', *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, p.DETC–99669, ASME, Philadelphia, PA.

Summers, J.D. (2005) *Comparative Study of CAD Interrogation Capabilities: Commercial CAD vs. Design Exemplar*.

Summers, J.D., Bayanker, S. and Gramopadhye, A. (2009) 'Experimental comparison of CAD input devices in synthesis, analysis, and interrogation tasks', *Computer-Aided Design and Applications*, Vol. 6, No. 5, pp.595–612, doi:10.3722/cadaps.2009.595-612.

Summers, J.D., Bettig, B. and Shah, J.J. (2004) 'The design exemplar: a new data structure for embodiment design automation', *Journal of Mechanical Design*, Vol. 126, No. 5, pp.775–787.

Summers, J.D., Divekar, A. and Anandan, S. (2006) 'Towards establishing the design exemplar as a CAD query language', *Computer-Aided Design and Applications*, Vol. 3, Nos. 1–4, pp.523–532.

Summers, J., Lacroix, Z. and Shah, J. (2002) 'Case-based design facilitated by the design exemplar', *Proceedings of the 7th International Conference on Artificial Intelligence in Design*, pp.453–476.

Ullman, D.G. (2010) *The Mechanical Design Process*, 4th ed., McGraw-Hill, New York.

Veltkamp, R.C. and Hagedoorn, M. (2001) *State of the Art in Shape Matching*, in M.S. Lew (Ed.): *Principles of Visual Information Retrieval*, p.87, Springer, London, UK.

Venkataraman, S., Shah, J.J. and Summers, J.D. (2001) 'An investigation of integrating design by features and feature recognition', *International Conference FEATS*.

Comment [t5]: Author: Please provide issue number.

Comment [t6]: Author: Please provide the publisher and place of publication.