Singularity Analysis of Geometric Constraint Systems

PENG Xiaobo (彭小波), CHEN Liping (陈立平), ZHOU Fanli (周凡利) and ZHOU Ji (周 济)

National CAD Support Software Engineering Research Center
Huazhong University of Science and Technology, Wuhan 430074, P.R. China

E-mail: pengxiaobo@hotmail.com; lpchen@mail.hust.edu.cn

Received January 11, 2001; revised August 21, 2001.

Abstract Singularity analysis is an important subject of the geometric constraint satisfaction problem. In this paper, three kinds of singularities are described and corresponding identification methods are presented for both under-constrained systems and over-constrained systems. Another special but common singularity for under-constrained geometric systems, pseudo-singularity, is analyzed. Pseudo-singularity is caused by a variety of constraint matching of under-constrained systems and can be removed by improving constraint distribution. To avoid pseudo-singularity and decide redundant constraints adaptively, a differentiation algorithm is proposed in the paper. Its correctness and efficiency have been validated through its practical applications in a 2D/3D geometric constraint solver CBA.

Keywords geometric constraint satisfaction, parametric design, singularity, redundant constraint, pseudo-singularity

1 Introduction

Singularity analysis of constraint is a key technology for geometric constraint satisfaction problems (GCSP). However, so far most researches and papers on GCSP have been focusing on constraint solving planning[1−6], i.e., constraint matching, sorting and decomposition, with only few papers involving singularity analysis. DCM is one of the best constraint solvers in the world. But according to our experience, definitions of constraint and dimension in DCM are affected by the operation order for under-constrained systems. It seems that DCM has not perfectly resolved the singularity problem for under-constrained systems. Light[1] tried to differentiate the redundancy of a constraint according to the singularity of its Jacobian matrix. Wang Boxing[7] indicated that if there is any redundant constraint, the Jacobian matrix of the system must be singular.

Our research is based on the directed constraint graph. Generally, graph-based singularity analysis is fit for well-constrained systems and over-constrained systems. In this paper, this method is extended to under-constrained systems. The reasons leading to a singular Jacobian matrix of a geometric constraint system are comprehensively analyzed and three types of singularities and corresponding judging methods are further presented.

2 Basic Concepts

A geometric constraint system can be described with a directed graph G[4], in which a directed arc a uniquely stands for a constraint c, and c matches with the head vertex of a. The process of creating a directed graph is in fact an iterative process of finding the matching vertex for each constraint and is essentially the bipartite matching of $\langle c, e\rangle$, where e is a geometric element. The following are several important concepts used in the paper.

Compound Vertex. In a directed graph G, a strongly connected sub-graph can be reduced to a single vertex called compound vertex.

This research is partly supported by the National '863' High-Tech Programme of China under grant No.9842-003.

Residual DOF (RDOF). The difference between the Degree of Freedom (DOF) of a vertex and the number of its matching constraints is called its RDOF; RDOF of a compound vertex is the sum of RDOF of all the vertices in it. If RDOF of a vertex is more than 0, it is free.

Propagation Set and Predetermination Set. In a Directed Acyclic Graph (DAG), the vertex set found by Depth First Ergodicity (DFE) from vertex v is called the propagation set of v, while the vertex set found by inverse DFE from v is called its predetermination set.

Transitive Path. Constraints c_1 and c_2 correspond to an in-arc and an out-arc of vertex v respectively. The explicit parameter sets in algebra expressions of c_1 and c_2 are P_1 and P_2 respectively. If $P_1 \cap P_2 \neq \emptyset$, then c_1 and c_2 are transitive at v and any change of either one will affect the other. If all the constraints along a path are transitive and the end vertex is free, then the path is transitive.

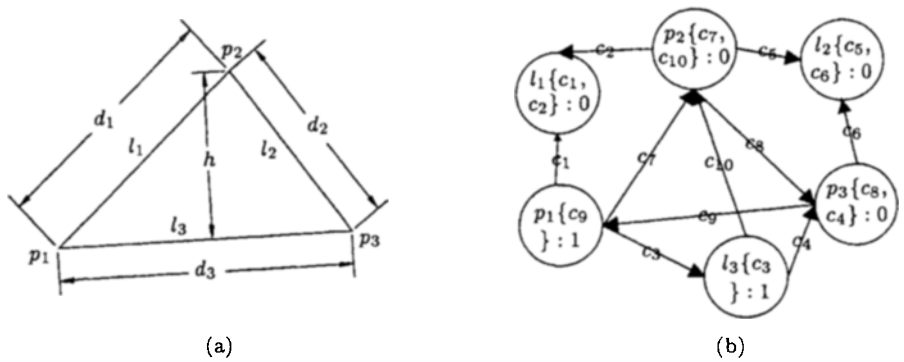
The remainder of this paper is organized as follows. Section 3 describes three types of singularity and analyzes pseudo-singularity for under-constrained systems. Section 4 advances the symbol method and the numerical method for singularity differentiation. In the section, a singularity differentiation algorithm is presented and some examples are illustrated. Section 5 gives a conclusion.

3 Singularity Types

A geometric constraint system can be expressed as a set of non-linear equations. Based on the algorithm for the blocking triangular form of sparse matrix, many dividing methods[1,3–5,8] have been proposed to decompose geometric constraint systems so that each geometric system is reduced to a set of sub-systems that can be solved orderly. In these methods, singularity analysis of constraint is very important.

3.1 Saturated Singularity

Chen Liping[3] presented the concept of Saturated Set: if under geometric constraint set C in which there is no redundant constraints, relative DOF between any two geometric entities in entity set E is zero, then E is called a Saturated Set under C. A saturated set can be regarded as a rigid body composed of its inner geometric entities. Any constraint c added into a saturated set will lead to a singular Jacobian matrix and thus is called a saturated redundant constraint. In this case, the bipartite matching operation of c, which is to find its matching vertex in the directed graph, will fail.



(a) (b)

Fig.1. Saturated singularity.

In Fig.1(a), when the lengths of the three sides are decided, the triangle is decided and can be regarded as a rigid body. The constraints are as follows:

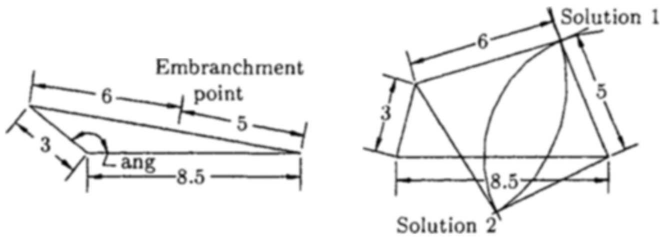
Geometric entity set: $E = \{p_1, p_2, p_3, l_1, l_2, l_3\}$; Constraint set: $C = \{c_i | i = 1, \ldots, 9\}$

$c_1 = PntOnLine(p_1, l_1)$; $c_2 = PntOnLine(p_2, l_1)$; $c_3 = PntOnLine(p_1, l_3)$; $c_4 = PntOnLine(p_3, l_3)$; $c_5 = PntOnLine(p_2, l_2)$; $c_6 = PntOnLine(p_3, l_2)$; $c_7 = DistPP(p_1, p_1, d_1)$; $c_8 = DistPP(p_2, p_3, d_2)$; $c_9 = DistPP(p_1, p_3, d_3)$.

If a new height constraint $c_{10} = Height(p_2, l_3, h)$ is added between p_2 and l_3, c_{10} will cause saturated singularity. Fig.1(b) is the corresponding directed graph.

3.2 Embranchment Singularity

According to mechanism kinematics, if at the time t^* there is a position q^* of the system where the Jacobian matrix $\Phi_q(q^*, t^*)$ is not singular, then according to the Implicit Function Theorem, for $\delta > 0$, the system has a stable unique solution $q(t)$ near q^* in the period of $|t - t^*| < \delta$. Solution 1 and Solution 2 in Fig.2(b) are two examples of q^*. On the other hand, if there exists locking or embranchment at time t^*, then when $t \to t^*$ there is no solution and the motion of the system loses its certainty. For example, at the position shown in Fig.2(a), the next moving direction of the system cannot be decided and the system becomes singular. In the research of multi-body systems, E. J. Haug[9] and Hong Jiazhen[10] considered that this kind of singularity occurs only at isolated points, e.g., the position in Fig.2(a), and called it isolated singularity.



(a) (b)

Fig.2

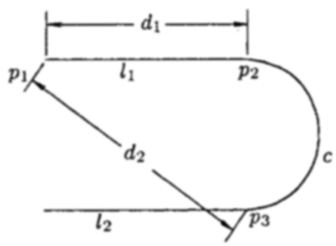


Fig.3. Embranchment singularity caused by tangency.

This case also occurs in geometric constraint satisfaction problems such as parametric design. Consider that two curves are constrained by a tangency constraint. The constraint system is non-linear and the Jacobian matrix Φ_q is dependent on the orientation parameter vector $\boldsymbol{q}$ of the system[11]. The system is singular. Obviously the singularity occurs only when tangency constraint exists or rather at the isolated point. We consider the tangent point as the isolated point of the system.

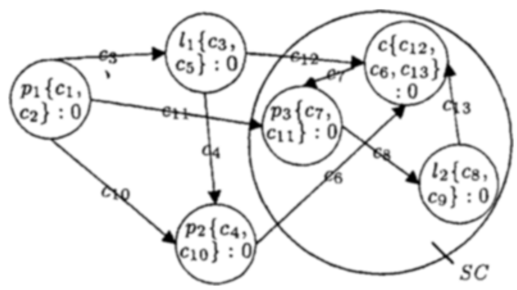


Fig.4. A directed graph of embranchment singularity.

There is only one difference between embranchment singularity of mechanism analysis and that of GCSP. Kinematics analysis can be considered as GCSP based on time sequence. But for static GCSP like parametric design, when we say a system is of embranchment singularity, we mean that the system is at a position where there are constraints leading to isolated singular points. For example, if the influence of time is not considered in Fig.2(a), when some constraints make the quadrilateral to be a triangle, the system is of isolated singularity. Similarly, in Fig.3, p_1 is fixed, l_1 and l_2 are horizontal, and c is tangent with l_1 and l_2. The constraints can be written as follows:

$c_1 = FixX(p_1)$; $c_2 = FixY(p_1)$; $c_3 = PntOnLine(p_1, l_1)$; $c_4 = PntOnLine(p_2, l_1)$; $c_5 = Horizontal(l_1)$;
$c_6 = PntOnCircle(p_2, c)$; $c_7 = PntOnCircle(p_3, c)$; $c_8 = PntOnLine(p_3, l_2)$; $c_9 = Horizontal(l_2)$;
$c_{10} = DistPP(p_1, p_2, d_1)$; $c_{11} = DistPP(p_1, p_3, d_2)$; $c_{12} = TanLC(l_1, c)$; $c_{13} = TanLC(l_2, c)$.

Fig.4 is the directed graph of Fig.3. The strongly connected sub-graph $SC = \{p_3, l_2, c\}$ has a 7×7 Jacobian matrix Φ_q. Φ_q is singular and $Rank(\Phi_q) = 5$. Its rank deficit is 2, being equal to the number of tangent points. Obviously, there is no redundant constraint here and the singularity comes from the tangencies between c and l_1, l_2.

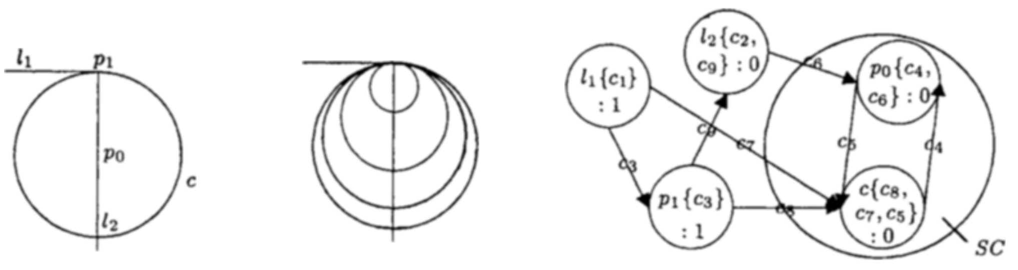
3.3 Redundancy Singularity

There is no essential difference between redundancy singularity and saturated singularity. They both are caused by redundant constraints. When a redundant constraint can be decided by a symbol method, the singularity is called saturated singularity. Otherwise it is called redundancy singularity. Redundancy singularity is often produced by the existence of equivalent geometric constraints.

Unlike saturated singularity, redundancy singularity is common when system is under-constrained, especially during the initial phase of design. For an under-constrained vertex, the symbol method cannot be used to judge its singularity by simply calculating its RDOF, because in this case, singularity might still occur even if the vertex is free. For instance, when a vertex DOF of 3 is constrained by two geometrically equal constraints, the vertex is free and the directed graph is normal, but singularity occurs. Consider the constrained geometry in Fig.5(a):

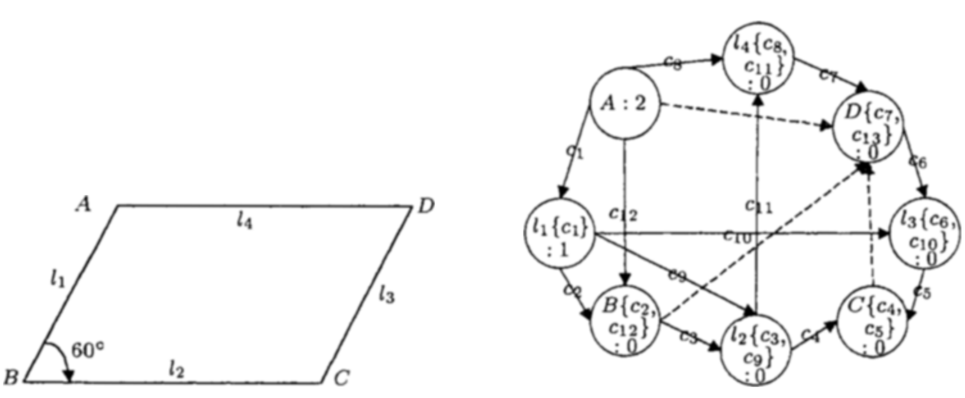
$c_1 = Horizontal(l_1)$; $c_2 = Vertical(l_2)$; $c_3 = PntOnLine(p_1, l_1)$; $c_4 = Cent\_X(p_0, c)$; $c_5 = Cent\_Y(p_0, c)$; $c_6 = PntOnLine(p_0, l_2)$; $c_7 = Tangent(l_1, c)$; $c_8 = PntOnCircle(p_1, c)$; $c_9 = PntOnLine(p_1, l_2)$.

Fig.5(c) is the directed graph of Fig.5(a). It can be seen that circle c and its center point p_0 constitute a strongly connected component SC. Although RDOF of SC seems to be zero in the graph, when the radius of c changes, all the constraints can still be satisfied (Fig.5(b)). That is to say, SC is not fixed and the system is singular. The singularity is just caused by the existence of equivalent geometric constraints. This kind of singularity cannot be decided by the symbol method and numerical method should be adopted.



(a) (b) (c)

Fig.5. Redundancy singularity.



(a) (b)

Fig.6. Redundancy singularity.

Fig.6(a) is another example: $AB//DC$, $AD//BC$, $|AB| = |CD|$, $\angle B = 60°$, $|AB| = 5$. The constraint set is:

$c_1 = PntOnLine(A, l_1)$; $c_2 = PntOnLine(B, l_2)$; $c_3 = PntOnLine(B, l_2)$; $c_4 = PntOnLine(C, l_2)$; $c_5 = PntOnLine(C, l_3)$; $c_6 = PntOnLine(D, l_3)$; $c_7 = PntOnLine(A, l_4)$; $c_8 = PntOnLine(D, l_4)$; $c_9 = AngBtLine(l_1, l_2, 60)$; $c_{10} = ParalLL(l_1, l_3)$; $c_{11} = ParalLL(l_2, l_4)$; $c_{12} = DistPP(A, B, 5)$; $c_{13} = EqualLen((A, B), (C, D))$.

$ABCD$ is a parallelogram. Fig.6(b) is the directed graph of Fig.6(a). Three dashed arcs in Fig.6(b) denote the four-element constraint $c_{13} = EqualLen((A, B), (C, D))$, which is different from binary constraints. Three arcs of c_{13} mean that c_{13} involves 4 vertices. But they work as a single one and reduce DOF of the vertex D by 1. It can be seen that the total RDOF of the system is 3 and there is no local over-constraint in the graph. The system seems to be well-constrained. However, it is obvious that CD can move along AD. That is to say, the system is under-constrained. So in the system there exists singularity. Here it is also impossible to judge the singularity by calculating DOF of the vertices. In fact, constraints $AB//DC$ and $AD//BC$ are partly equivalent to $|AB| = |CD|$, which leads to the singularity.

3.4 Pseudo-Singularity

For an under-constrained system, the matching modes between constraints and entities are various. Different constraint matching modes produce different solving sequences. Some of these solving sequences may contain singular units and are called ill-conditioned solving sequences. However, this kind of singularity can be removed by improving the constraint distribution. If a kind of singularity can be avoided by adjusting constraint distribution, it is called pseudo-singularity.

In the example of Fig.7, l_1 is horizontal and fixed. p is the intersection point of l_1 and l_2. A is the angle between l_1 and l_2. This under-constrained system has two possible directed graphs (Fig.7(a-1) and Fig.7(b-1)). Now add an angle constraint $c_3 = AngLL(l_1, l_2, 180)$ between l_1 and l_2. Then (a-1) and (b-1) develop to (a-2) and (b-2) respectively. In the case of (b-2), intersection point p cannot be solved from two superposed lines l_1 and l_2. The system is singular. However, for the same system, the solving sequence in (a-2) is normal. (b-2) is of pseudo-singularity. To avoid the pseudo-singularity in (b-2), adjust constraint c_2 and let it be matched with $v(l_2)$, then the directed graph is optimized to (a-2) and the solving sequence $v(l_1) \to v(p) \to v(l_2)$ becomes normal.

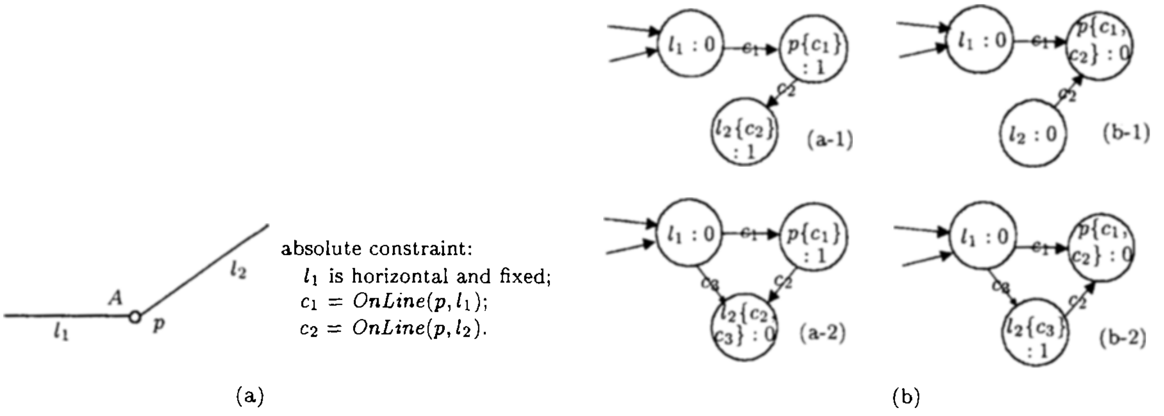


Fig.7. An example of pseudo-singularity.

Then we can conclude that: (1) For an under-constrained system, singularity of a constraint cannot be simply determined by the singularity of a single vertex; (2) For an under-constrained system, if a constraint has at least a normal solving sequence, then it is normal.

We adopt the steps as follows to eliminate pseudo-singularity. Once an ill-conditioned vertex v is obtained, constraint matching adjustment[8] will be implemented on the non-propagation set of v to find an inverse transitive path P. Reverse the arcs along P to change the constraints matched with v so that the propagation set of v is enlarged and a new solving sequence is obtained. To ensure the new solving sequence is well-conditioned, the algorithm needs to check all the vertices in the new solving sequence. Please refer to [8] for details of constraint matching adjustment.

4 Singularity Differentiation

4.1 Graph Differentiation Method

A geometric constraint system has the following three properties:

Property 1. *If constraint c is not redundant, then there must be a geometric entity e in the system, $DOF(e) > 0$ and c can be matched with e $(\langle c, e\rangle | e \in E)$.*

This property comes from the definition of directed geometric constraint graph. A normal constraint must have a matching vertex which corresponds to a geometric entity. Otherwise, the directed graph cannot be created correctly.

Property 2. *If constraint c is redundant, then the Jacobian matrix Φ_q of the matching vertex of c is singular. In our research, it means Φ_q has row rank deficit.*

Any redundant constraint belongs to a dependent constraint set. In our research, each row vector of the Jacobian matrix of the dependent constraint set is composed of the partial derivatives of orientation parameters of one constraint equation in the set. Because of the dependency of the constraint set, the row vectors of its Jacobian matrix are also dependent and the Jacobian matrix must have row rank deficit.

Property 3. *Assume that v_s is a compound vertex and there is no unary constraint that describes an absolute position in v_s. If (i) the predetermination set of v_s is empty and RDOF of v_s is less than that of a rigid body or (ii) the predetermination set of v_s contains only one vertex v_p, the RDOF of v_s is zero and the number of out-arcs from v_p to v_s is more than DOF of v_p, then there must be saturated singularity.*

Generally, there are two methods to position v_s. One is using unary constraints that describe absolute positions such as fixed position, horizontal, vertical and so on, which is excluded by Property 3. The other method is to position by its predetermination set PD_s. If PD_s is empty or includes only one geometric entity whose DOF is less than that of v_s, obviously it is impossible for v_s to be positioned and there must be saturated singularity among the constraints of v_s.

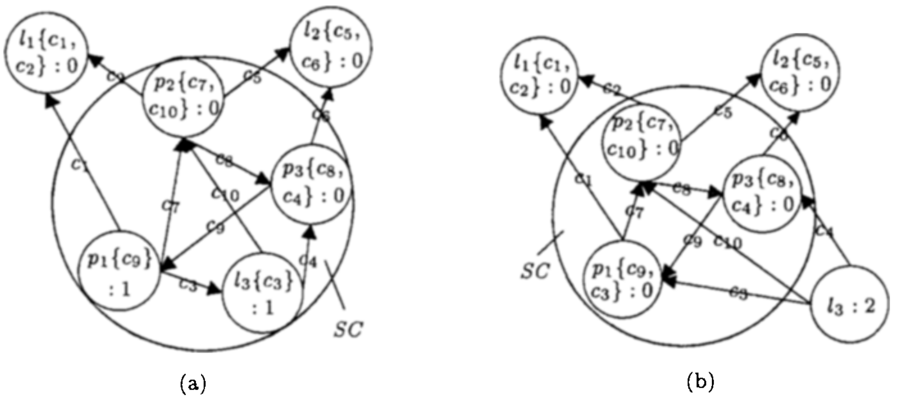


Fig.8. Graph characteristics of saturated singularity.

Properties 1 and 2 are two necessary conditions for a normal constraint. Property 3 indicates the graph characteristics of saturated singularity and becomes an efficient symbol differentiation method for that. For instance, Fig.8(a) is the strongly connected graph of Fig.1(b), in which the only strongly connected sub-graph SC has 4 entities and its RDOF is $2 \times 4 - 6 = 2$. After the optimal matching adjustment[8], which can ensure RDOF of any compound vertex v_i: $RDOF(v_i) \to 0$, SC contains 3 entities $SC = \{p_1, p_2, p_3\}$ and $RDOF(SC) = 2 \times 3 - 6 = 0$ (Fig.8(b)). The predetermination set of

SC is $PD_s = \{l_3\}$. DOF of l_3 is 2. However, it can be seen that there are three arcs from l_3 to SC, corresponding to constraints c_3, c_4, c_{10}, which means rigid body SC whose DOF is 3 is fully determined by l_3 whose DOF is 2. It is obviously impossible. According to Property 3, there must be saturated singularity for SC.

4.2 Constraint Residue Perturbation Method

For redundancy singularity and embranchment singularity, the numerical judging method should be applied. E. J. Haug[9] presented a variable perturbation method to decide singular points. This method is efficient. However, it applies perturbation to all entities in the system indiscriminately and the scope of perturbation does not vary for different entities, which increases the expense of time of the method.

Considering that singularity is caused only by redundant constraints, we advance the Constraint Residue Perturbation Method (CRPM) which only involves the redundant constraint set.

If Jacobian matrix Φ_q of a solving unit v is singular, then redundant constraint set $C_r \subset C$ can be obtained easily by the Gaussian Elimination Method. Here C is the constraint set of v. Set a perturbing value δ for each constraint $c_i \in C_r$ and then adopt Newton-Raphson iteration. For

$$\forall c_i \in C_r \rightarrow F_i = F_i \pm \delta,$$

where $\delta > \varepsilon$, ε is the iteration precision; $\pm$ means perturbation direction; F is the constraint equation. Then compute with the iteration formula

$$q_{j+1} = q_j - \Phi_{q_j}^{+} \bullet F_j,$$

where q is the orientation parameter vector of v, $\Phi_{q_j}^{+}$ is the general inverse of Φ_q. Because the perturbation is small, after a few steps of iteration we can determine whether there is a solution or not. If there is a solution for either perturbation direction for c_i and row rank deficit of Φ_q decreases, the singularity caused by c_i is embranchment, or it is redundancy. By this means, the types of singularity of all the constraints in C_r can be decided.

In Fig.5, there is redundancy among the constraints between compound vertex SC and its predetermination set $PD_s = \{l_1, l_2, p_1\}$. $C_r = \{c_6, c_7, c_8\}$. For c_6, the perturbation is:

$$a_2 x_0 + b_2 y_0 + c_2 = \pm\delta,$$

where a_2, b_2, c_2 are parameters of l_2. After perturbation, l_2 will have a slight translation and thus the intersection point of l_1 and l_2 will not be on circle c (Fig.9(a)). There is no solution for this perturbation. Consider c_7, the perturbation is:

$$a_1 x_c + b_1 y_c + c_1 \pm r = \pm\delta,$$

where a_1, b_1, c_1 are parameters of l_1. It means that l_1 moves along l_2 (Fig.9(b)). Obviously, the perturbing result also makes p_1 not on circle c, which is inconsistent with c_8 and there is also no solution. The perturbation for c_8 is

$$\sqrt{(x_c - x_1)^2 + (y_c - y_1)^2} - r = \pm\delta.$$

It means that the radius of circle c changes for the perturbation. From Fig.9(c), it can be seen that after perturbation, all the constraints are still satisfied and perturbing solution can be easily obtained. The Jacobian matrix Φ_q of SC is:

$$\begin{array}{cccccc}
 & x_0 & y_0 & x_c & y_c & r \\
c_4: & 1 & 0 & -1 & 0 & 0 \\
c_5: & 0 & 1 & 0 & -1 & 0 \\
c_6: & a_2 & b_2 & 0 & 0 & 0 \\
c_7: & 0 & 0 & a_1 & b_1 & 1 \\
c_8: & 0 & 0 & (x_c - x_1)/R & (y_c - y_1)/R & -1
\end{array}$$

where $E = \{p_0, c\}$, $C = \{c_4, c_5, c_6, c_7, c_8\}$, $q = [x_0, y_0, x_c, y_c, r]^{\mathrm{T}}$, $R = \sqrt{(x_c - x_1)^2 + (y_c - y_1)^2}$.

Because p_0 and p_1 are always on l_2 before and after perturbation and the values of $(x_c - x_1)/R$ and $(y_c - y_1)/R$ do not change, the row rank deficit of Φ_q is unaltered. According to CRPM, c_8 causes redundancy singularity.

Similarly, in Fig.10(b), line l is tangent with circle c. The system is singular. After either perturbation shown in Fig.10(a) or Fig.10(c), the row rank deficit of the Jacobian matrix of the system decreases to 0, which shows the system has embranchment singularity.

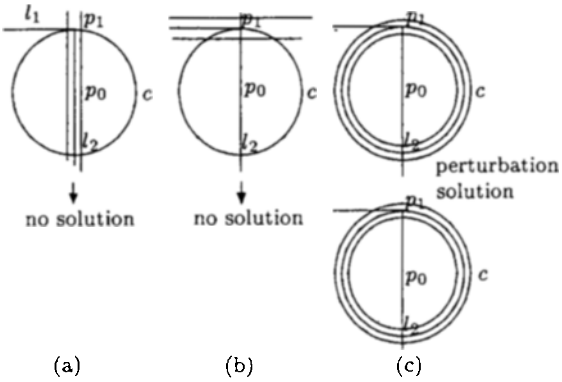


Fig.9. Redundancy singularity.

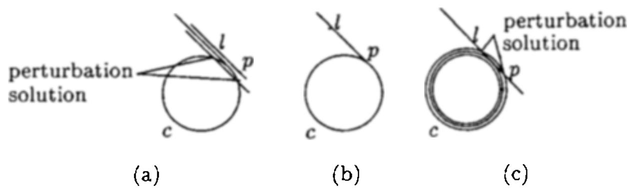


Fig.10. Embranchment singularity.

The algorithm of singularity differentiation can be summed up as follows:

Algorithm 1. Globally Judging the Redundancy of Constraint c in an Under-Constrained System

Step 1. If the bipartite matching of c fails, c is redundant; return FALSE.

Step 2. Initialize directed graph G; get the propagation set PP_s of c.

Step 3. Adopt the optimal adjustment algorithm[8] to obtain the strongly connected sub-graphs on PP_s and make G a DAG. Then the solving sequence $S = \{v_i\}$ from c can be obtained along the out-arcs of c.

Step 4. If (there is $v \in S$ and v is of saturated singularity) then c is redundant and return FALSE;
else if (there is $v \in S$ and v is of redundancy singularity) then {search for inverse transitive path P on the non-propagation set of v};
else {print("c is normal constraint"); output the solving sequence; return TRUE}.

Step 5. If (P is empty) c is redundant; return FALSE;
else{apply reversing operation on P; reset the solving sequence S; goto Step 2}.

In Algorithm 1, Step 4 needs to judge the singularity type of vertex $\boldsymbol{v}$. Algorithm 2 gives this judging method:

Algorithm 2. Judging the Singularity Type of Vertex $\boldsymbol{v}$

Step 1. If ($\boldsymbol{v}$ satisfies Property 3) return "$\boldsymbol{v}$ is of saturated singularity".

Step 2. Compute the Jacobian matrix Φ_q of $\boldsymbol{v}$.

Step 3. If (Φ_q is normal) return "$\boldsymbol{v}$ is normal".

Step 4. Compute dependent constraint set C_r of $\boldsymbol{v}$ by the Gaussian Elimination Method.

Step 5. $i = 1$, r is the number of constraints in C_r.

Step 6. $c_i \in C_r$, $r_0 = Rank(\Phi_q)$.

Step 7. Add a perturbation δ on c_i.

Step 8. Solve $\boldsymbol{v}$.

Step 9. If (there is no solution) return "$\boldsymbol{v}$ is of redundancy singularity";
else if ($r_1 = Rank(\Phi_q) \leq r_0$) return "$\boldsymbol{v}$ is of embranchment singularity".

Step 10. If ($i < r$) {$i + 1$, go to Step 6}.

Step 11. return "$\boldsymbol{v}$ is normal".

Algorithm 1 adaptively enlarges the propagation set of constraint c according to the singularity types of the vertices in the solving sequence SC and eliminates the singularity. Because to repeat Step 2 is limited, the complexity of the algorithm is approximately equal to that of DFE: $O(n+e)$[12]. The algorithm works well for under-constrained systems. However, it is also efficient for well-constrained and over-constrained systems.

4.3 Some Examples

Using the O-O method we have presented a united modeling method for both 2D and 3D geometric constraints and developed a 2D/3D geometric constraint solver CBA (Constraint Broadcasting Automation). Fig.11 illustrates the hierarchy of CBA.

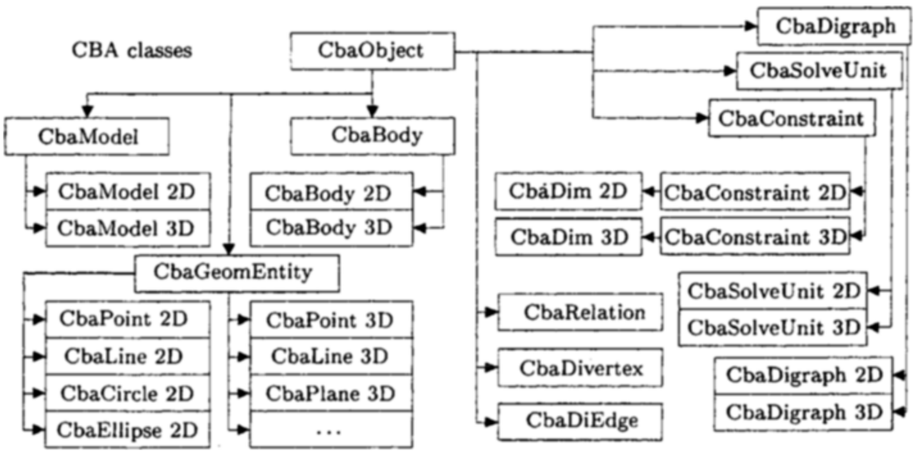
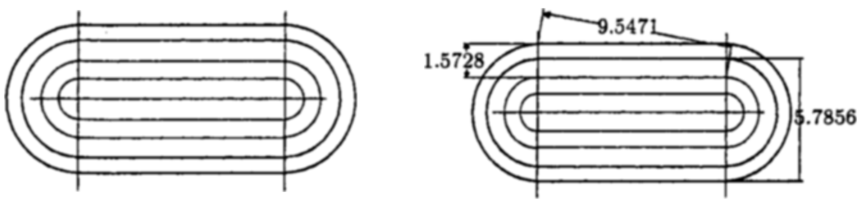


Fig.11. The hierarchy of CBA.



(a) (b)

Fig.12. An example of constraint identification.

(a) (b)

Fig.13. Tripod assembly.

The module of constraint identification in CBA2D transforms general engineering graphs to parametric ones and identifies geometric constraints according to relative positions of geometric entities. Fig.12(a) shows a key way that has n loops. By geometric detection, $2n+1$ horizontal constraints, 2 vertical constraints, $4n$ tangent constraints and $4n+4$ point-on-line constraints are identified. The algorithm also identifies successfully that there are $n-1$ redundant tangency and $4n+1$ redundant point-on-line constraints in these constraints. Finally, $2n+1$ horizontal constraints, 2 vertical constraints, $4+3(n-1)=3n+1$ tangent constraints and 3 point-on-line constraints are obtained by CBA2D. In this example, all the types of singularity mentioned above occur. Particularly, after the constraint identification, labeling some additional dimensions (e.g., Fig.12(b)) will lead to sophisticated embranchment singularity. CBA2D also can deal with such kind of problems.

In another example, in Fig.13(a), pole 2 mates with pole 1 and pole 3 by two co-axis relations. Each co-axis relation includes 4 constraints. If another co-axis relation is added between pole 1 and pole 3, CBA3D will close the tripod and find out two redundant constraints (Fig.13(b)). (Please refer to [11] for our research on geometric constraint expression and decomposition). Practice has proved that the differentiation method presented in this paper is fast and efficient.

5 Conclusion

Singularity is an important factor that affects directly the solving ability of a geometric constraint solver. In this paper, three types of singularities are presented, and pseudo-singularity, which is very common in under-constrained systems, is analyzed. As one part of the kernel of our geometric constraint solver CBA, the graph differentiation method and the constraint residue perturbation method are advanced to differentiate efficiently singular constraints. All the algorithms and methods proposed in this paper can also be applied to 3D geometric constraint systems and have proved efficient and fast in CBA 3D.

Acknowledgement The authors are very grateful to the suggestions for improving the paper given by the anonymous reviewers.

References

[1] Light R, Lin V, Gossard D. Variational geometry in CAD. *Computer Graphics,* 1981, 15(3): 171–177.
[2] Kramer G A. A geometric constraint engine. *Artificial Intelligence,* 1992, 58: 327–360.
[3] Chen Liping, Zhou Ji. Research of geometric constraint system reasoning. *Journal of Huazhong Univ. of Sci & Tech,* 1995, 6: 70–74. (in Chinese)
[4] Dong Jinxiang. A new thought of constraint solving in variable drawing systems. *Journal of Computer-Aided Design & Computer Graphics,* 1997, 9(6): 513–519. (in Chinese)
[5] Jae Yael Lee, Kwangsoo Kim. A 2-D geometric constraint solver using DOF-based graph reduction. *Computer-Aided Design,* 1998, 30(11): 883–896.
[6] Gao Xiaoshan, Chou Shangching. Solving geometric constraint systems. Part II, A symbolic approach and decision of Re-constructibility. *Computer-Aided Design,* 1998, 30(20): 115–122.
[7] Wang Boxing, Chen Liping, Zhou Ji. Study and practice of key technologies of geometric constraint driven function in traditional drafting systems. *Computer Research & Development,* 1998, 35(10): 935–940. (in Chinese)
[8] Chen Liping, Wang Boxing. An optimal method of bipartite graph matching for underconstrained geometry solving. *Chinese Journal of Computers,* May, 2000, 23(5): 523–530. (in Chinese)
[9] Haug E J. Computer Aided Kinematics and Dynamics of Mechanical Systems. Allyn and Bacon, Vol.I, 1989.
[10] Hong Jiazhen. Computational Dynamics of Multibody Systems. Advanced Education Press, China, 1996, (4): 70. (in Chinese)
[11] Chen Liping, Peng Xiaobo. An approach to a 2D/3D geometric constraint solver. In *Proceedings of ASME DETC'00/DAC-14515,* Baltimore, Maryland, September 10–13, 2000.
[12] Yan Weimin, Wu Weiming. Data Structure. Tsinghua University Press, China, 1992. (in Chinese)

PENG Xiaobo received his B.S. and M.S. degrees from Huazhong University of Science and Technology, Wuhan, in 1995 and 1998 respectively. Now he is a Ph.D. candidate in the National CAD Support Software Engineering Research Center, Huazhong University of Science & Technology. He specializes in geometric constraint solving, solid modeling and design automation. His current research activities focus on the research of key technologies of 3D variable geometry, whose primary goal is to provide an effective geometric constraint solver for 2D/3D CAD system applications.

CHEN Liping received his Ph.D. degree from Huazhong University of Science and Technology, Wuhan, in 1995. He is currently a professor in the National CAD Support Software Engineering Research Center, Huazhong University of Science & Technology. His research interests are in the areas of geometric constraint solving, kinematics and dynamics, feature-based and dimension-driven solid modeling and CAD/CAM integration.

ZHOU Fanli is a Ph.D. candidate in the National CAD Support Software Engineering Research Center, Huazhong University of Science & Technology, Wuhan. His research interests include kinematics, dynamics and stimulation.

ZHOU Ji received his Ph.D. degree from the State University of New York at Buffalo in 1984. He is a member of Chinese Academy of Engineering and a professor at Huazhong University of Science & Technology, Wuhan. His research interests focus on CAD/CAM systems, geometric modeling, optimization and intelligent design.