Proceedings of TMCE 2016, May 9-13, 2016, Aix-en-Provence, France, edited by I. Horváth, J.-P. Pernot, Z. Rusák.

ON THE DETECTION OF OVER-CONSTRAINED SUBPARTS OF CONFIGURATIONS WHEN DEFORMING FREE-FORM CURVES

Hao Hu
Arts et Métiers ParisTech.
LSIS UMR CNRS 7296
Hao.hu@ensam.eu

Mathias Kleiner
Jean-Philippe Pernot
Arts et Métiers ParisTech.
LSIS UMR CNRS 7296
{ mathias.kleiner@ensam.eu, jean-philippe.pernot@ensam.eu}

ABSTRACT

Today, designers use CAD modelers to define and modify NURBS surfaces involved in the design of complex shapes like car bodies or turbine blades. The generated shapes often result from the use of variational modeling techniques where user-specified constraints define the shapes. However, for free-form curve/surfaces, if too much constraints are added to subparts of a configuration, the system will not be solvable even if it is globally well-/under-constrained. When this happens, it is useful to identify locally unsatisfiable subparts of configurations and provide the user feedback for adjustment. Currently, in the domain of geometric constraint solving, techniques are mainly developed for Euler geometries rather than parametric entities like free-form curves/surfaces. In this paper, we apply the Dulmage-Mendelsohn decomposition method to isolate structural over-constrained subparts of configurations. Since structural over-constraints do not necessarily mean unsatisfiable, a Jacobian matrix analysis approach is taken to further detect the inconsistent constraints. Indeed, these numerical methods can be generalized to detect overconstraints on free-form curves. We illustrate our approach on different examples where results show that Gauss elimination, though restricted to linear cases, is more relevant in our context than Dulmage-Mendelsohn decomposition.

KEYWORDS

Free-form curves, locally unsatisfiable or redundant configurations, Dulmage-Mendelsohn decomposition, Jacobian matrix analysis

1. INTRODUCTION

With the evolution of manufacturing technologies and advances in the domain of new materials, today's CAD modelers need to develop high quality and complex shapes of products. Free-form shapes, especially in the consumer products industry, not only need to meet aesthetic criteria but also satisfy functional criteria.

The shape of a dashboard, a wing or a turbine blade are examples of complex shapes obtained by modifying

NURBS surfaces to satisfy user-specified constraints[12]. To satisfy these requirements, designers are relying on computer aided design (CAD) software.

Free-form curves are often designed using deformation through position, distance tangency, and/or continuity constraints. However, unlike 2D sketching where inconsistent constraints can be identified interactively during the design process, there is no indicator to analyze the constrained status of free-form curves. This is partly due to the absence of suitable techniques for solving algebraic problems that underlie more complicated specifications. In this paper, we seek to extend the shape vocabulary of geometric constraint solvers to free-form class by applying methods to analyze the structure of system of equations.

Therefore, our goal is to provide a diagnosis to the user through the identification of inconsistent constraints, subparts, and corresponding explanations.

Plan of the article

Section 2 summarizes over-constraints detection methods from the geometric constraint solving domain. In section 3, we manually apply a degree of freedom analysis on different B-spline curves in order to introduce the examples used in the following sections. In section 4, we illustrate the use of the

Dulmage-Mendelsohn (D-M) decomposition algorithm to identify over-constrained parts of the corresponding equation systems. In section 5, we show how a Gauss elimination can be used in the linear case to further discriminate inconsistent and redundant constraints. Finally, we discuss about future work.

2. RELATED WORK

2.1. Selection of geometric constraint methods for overconstraints detection

An over-constrained system is a system which contains redundant or conflicting constraints. We propose to classify the existing constraint solving methods into two categories with respect to the type of over-constraints detected. Graph based decomposition allow to identify structural over-constraints. However these methods will not detect constraints that are numerically inconsistent with others in the system [15]. Algebraic methods, such as Gauss elimination or Grobner bases, thus generalize the notion of over-constraints. However time complexity is exponential in the non-linear case, thus limiting their use to small systems[8]. For a general constraint solver, common practice is to first exploit geometric domain knowledge to transform a constraint system into a set of small solvable subsystems and then compute and assemble the solutions of its subsystems. Our presentation in the following part is intended to reflect schemes that are more suited to isolate over-constraints within the two phases.

Geometric constraint system decomposition

Jermann and Trombettoni[16] have proposed to classify the existing decomposition approaches into 4 categories according to the way they operate.

1) The recursive division methods

These methods, first introduced by Owen [17], work by recursively splitting the constraint system into subgraphs. It handles 2D constraint systems formed by points and lines linked by distance and angle constraints. These methods are used to detect whether a system is solvable, but will not provide explanations on which constraints cause the inconsistencies.

2) The recursive assembly methods

These methods adopt a bottom-up scheme by firstly finding rigid subsystems, called cluster and then iteratively aggregate rigid components into bigger ones[18]. This allows to find minimal rigid (dense) subgraphs. Later, Hoffman & al. [5] extends the method to locate 1-overconstrained dense subgraphs by modifying the incremental network maximum flow algorithm. Although this is sufficient when incrementally placing constraints during the design, it cannot be used when combining or editing constrained designs.

3) The single-pass methods

These methods decompose a system in a single iteration where all the components are produced simultaneously. There are two kinds of methods: maximum matching and weighted maximum matching. Serrano[26] describes the detection of over-constraints by finding unmatched constraints after applying maximum matching to the constraint system, where all constraints and entities have one degree of freedom. Also, D-M decomposition algorithm can decompose an equation system into over-,well- and under-constrained subsystem based on maximum matching[4]. Latham and Middleditch extends the work of Serrano by proposing weighted maximum matching to identify overconstraints with an arbitrary number of degrees of freedom[6]. It is performed directly on the constraint graph of the geometric system, avoiding the need to translate it into an equation system.

4) The propagation of degree of freedom approaches(PDOF)

According to Trombettoni[19], these approaches are more suitable for under-constrained systems. Redundant equations must be detected and removed before launching PDOF. Therefore, they can not be used for overconstraints detection.

Algebraic methods for detecting inconsistent system of equations

In general, almost all of geometric constraints can be translated mechanically into a set of non-linear equations. The equations are usually algebraic, and non-algebraic formulations involving trigonometric functions can be avoided in nearly all cases[21]. Therefore, at the equation level, detecting geometric overconstraints turns into identifying a set of inconsistent or redundant equations among a system of equations.

1) Numerical methods

Equations are said independent if the Jacobian matrix has a full rank at the searched root. For linear systems, the Jacobian matrix is composed of coefficients of equation systems, which is independent of the roots. Light and Gossard use Gauss Elimination(GE) to compute the rank as well as further identify invalid equations[14]. Serrano extends this work to check existence of overconstraints within strong connected components of equation systems[7]. However, such roots are hard to find in non-linear systems, therefore computing the Jacobian matrix is a difficult task. In rigidity theory, classical Numerical Probabilistic Method(NPM) studies the structure of the Jacobian at a random configuration[1]. Michelucci extends the NPM to study Jacobian structure at a configuration satisfying the incidence constraints (collinearities and coplanarities)[2].Based on the extended NPM, he then develops the Witness Configuration Method(WCM) to identify subtle dependences between geometric constraints[20].

2) Symbolic methods

Thanks to the consistency algorithm from algebraic geometry[8], for polynomials $f_1,\cdots,f_s \in \mathbb{C}[x_1,\cdots,x_n]$,we compute a reduced Grobner basis(GB) of the ideal with respect to any ordering. If this basis is {1}, then the polynomials have no common zero in $\mathbb{C}$; otherwise, they must have a common zero. In theory, this algorithm can be used to identify all the over-constraints, but the computation cost is exponential and thus only limited to small systems. Other symbolic methods like Wu-ritt as well as theory of determinants(TD) are also able to characterize over-constrained sets but have the same practical limitation as Grobner base.

Constraint solving in terms of curves

D.Podgorelec extends geometric elements to Bezier curves and ellipses by mapping them into auxiliary lines and points and uses local propagation to solve the resulting constraint system[9]. Y. J. Ahn implements the construction of quadratic Bezier curves subject to tangency and length or energy minimization constraints , which offers the ability to find curves that are to have prescribed clearance from points, circles or straight-line borders[10]. Iddo Hanniel & al. use symbolic polynomials to represent inequality expressions defined by conditions on curves and apply inequality constraints to coefficients of symbolic polynomials[11]. These methods are focusing on solving specific issues of CSP(constraint satisfaction problems) on curves, thus lacking the generality. To the best of our knowledge, general methods have not emerged yet.

2.2. Dulmage-Mendelsohn Decomposition

Dulmage-Mendelsohn decomposition algorithm allows to decompose a system (based on bipartite graph) into over-constrained, well-constrained and under-constrained part [22]. The well-constrained part can be further induced into irreducible subparts. Here we extend this method to analysis system of

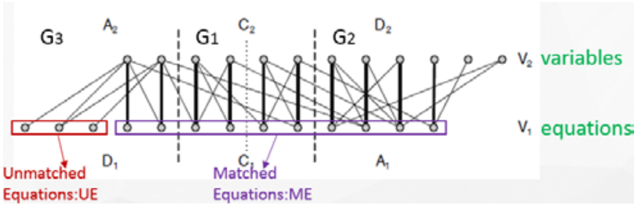


Figure 1 D-M decomposition

constraints represented in equation level.

The decomposition has the following properties:

1) G3 is structural over-constrained part, G1 is structural well-constrained part, G2 is structural under-constrained part

2) The unmatched equations(UE) are either conflicting or redundant

3) G1 , G2 and G3 are independent of the choice of the maximum matching

Since here the numerical information of equation system is ignored, structural over-constrained part is subset of over-constrained parts of a system, that is to say, except G3, it is also possible for G1 to have overconstraints. Algebraic methods are proposed to further analysis the irreducible subparts contained in G3. For a discussion on this, the reader is referred to the work of Serrano[7]. In section 4.4, we will give such examples on B-splines.

2.3. Jacobian structure of linear constraints

The Jacobian matrix

The Jacobian matrix is a matrix (n*m) containing the partial derivatives of each constraint equation with respect to each degree of freedom. It is usually defined and arranged as following:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (1)$$

Where i^{th} constraint is in the form:

$$f_i\left(x_1, x_2 \cdots x_j\right) = 0 \quad (2)$$

Identify inconsistent equations

Equations (2) are independent if the Jacobian has full rank at the searched root. For a linear constraint system, Jacobian matrix is the coefficient matrix of the equation system and by using Gauss Elimination methods, we can check the rank as well as further analyze its structure.

Constructing the Jacobian of (2), the following system is obtained:

$$x' = x - J^{-1} f$$

(3)

Reformulate equation (3), we have:

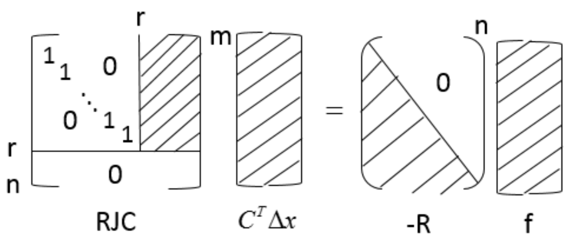
$$J\Delta x = -f \quad (4)$$

Where $\Delta x = x' - x$.

Use the transformation R and C, equation (4) becomes:

$$RJCC^T\Delta x = -Rf \quad (5)$$

The structure of equation (5) is as following:



(6)

Equation (7) represents the i^{th} row of equation (6), where subscript i = r+1…n, indicating a specific row of the matrix.

$$(RJC)_i(C^T\Delta x)_i = -R_i f \quad (7)$$

Since these rows of the product RJC are all zeros, in order to have a solution for $(C^T\Delta x)_i$, the following must be satisfied:

$$-R_i f = 0 \quad \text{i=r+1…n} \quad (8)$$

Hence, the non-zero terms of the i^{th} *row* in equation (8) indicate an inconsistent set of constraint equations.

3. TEST CASES

In this section, we consider constraints satisfaction problems only on curves since it allows us to illustrate our problems clearly.

3.1. Simple Curves

In this section, we set up three constraint satisfaction problems on B-splines with control points as variables. For the three configurations, constraints of curve 1 are the same while for curve 2, constraints are added incrementally. The objective is to minimize the deformation of the two curves, while satisfying the constraints. However, none of the

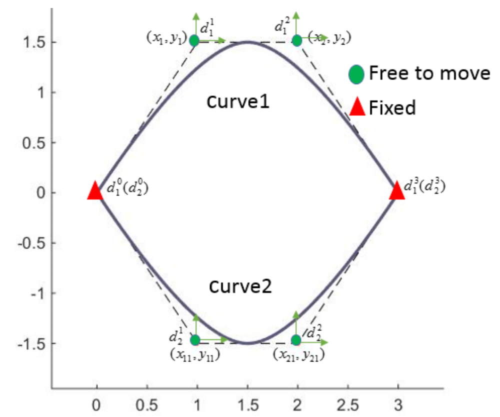


Figure 2: Configuration One

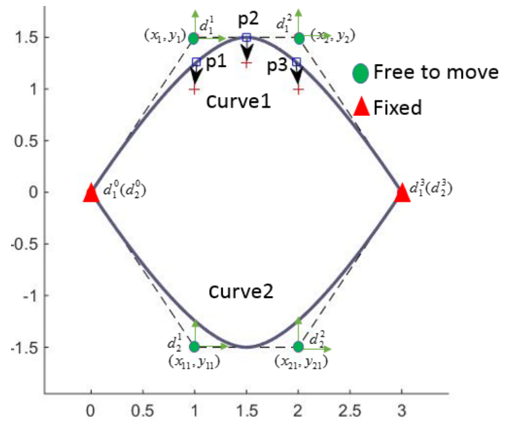


Figure 3: Configuration Two

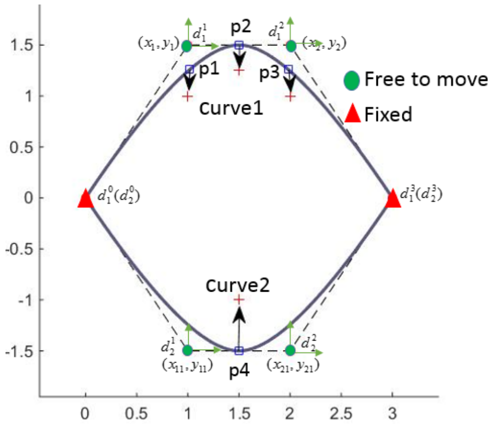


Figure 4: Configuration Three

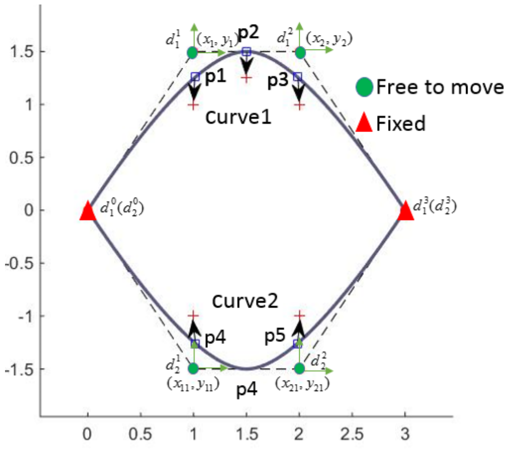


Figure 5: Configuration Four

configurations are satisfiable. In section 3.3, by applying degree of freedom(DOF) analysis based on local support property[23], these examples will be used to illustrate several key concepts.

3.2. Complex curves

Here is a B-spline curve with 13 control points(Fig 7). 10 of them (green ones) are free to move in order to satisfy 9 position constraints (p1-p9). Again, we want to minimize the deformation of the curve while satisfying the 9 constraints. Here again, the constraints can not be satisfied.

3.3. Degree of freedom analysis based on local support property

We apply degree of freedom(DOF) analysis on these examples manually based on local support property.

DOF analysis of configuration two-configuration four

According to the local support property of B-splines, constrained points on curve 1 and curve 2 are influenced by certain area of control points (Table 1). Each free control point has a DOF of 2 and position constraints constrains 1 degree of freedom. Table 2 summarizes the DOF analysis results.

3.3.1. DOF analysis of configuration six

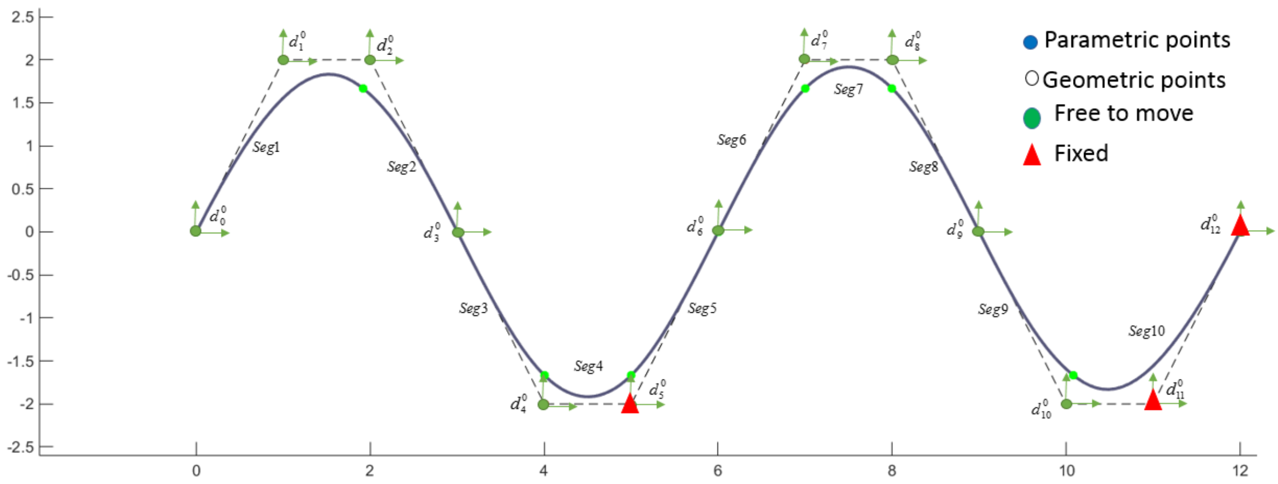


Figure 6: Configuration five-a B-spline curve

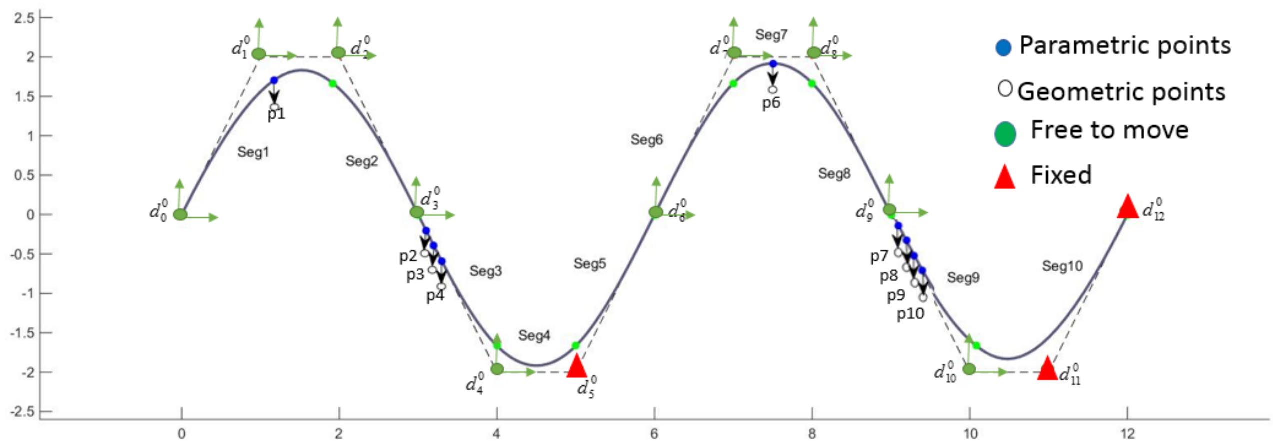


Figure 7: Configuration six-The curve with 9 position constraints

	p1	p2	p3	p4	p5
Influenced by	$d_1^0 - d_1^2$	$d_1^1 - d_1^3$	$d_1^1 - d_1^3$	$d_2^0 - d_2^2$	$d_2^1 - d_2^3$

Table 1: Area of control points that influence the constrained points (configure two – configure four)

	Configuration two		Configuration three		Configuration four	
	Curve 1	Curve 2	Curve 1	Curve 2	Curve 1	Curve 2
DOF	4	4	4	4	4	4
DOC	6	0	6	2	6	4
Status(local)	Over	Under	Over	Under	Over	Well
Status(global)	Under		Well		Over	

Table 2: Constrained status of 10 segments (over/under/well is short for over-/under-/well-constrained)

	Seg1	Seg2	Seg3	Seg4	Seg5	Seg6	Seg7	Seg8	Seg9	Seg10
Affected by	$d_0^0 - d_3^0$	$d_1^0 - d_4^0$	$d_2^0 - d_5^0$	$d_3^0 - d_6^0$	$d_4^0 - d_7^0$	$d_5^0 - d_8^0$	$d_6^0 - d_9^0$	$d_7^0 - d_{10}^0$	$d_8^0 - d_{11}^0$	$d_9^0 - d_{12}^0$
Dof	2*4	2*4	2*3	2*3	2*3	2*3	2*4	2*4	2*3	2*2
Doc	2*1	0	2*3	0	0	0	2*1	0	2*4	0
Status	Under	Under	Well	Under	Under	Under	Under	Under	Over	Under

Table 3: Constrained status of 10 segments (over/under/well is short for over-/under-/well-constrained)

The curve is divided into 10 segments (Seg1-Seg10). End points of each segment are calculated by P(Ui) and P(Ui+1), where Ui and Ui+1 are knots. Analysis result is shown in Table 3.

From Table 2 to Table 3, we know that a constraint system can be locally over-constrained but globally be under-, well-and over-constrained. Since all these constraint systems have no solution, an over-, under- and well-constrained system described from degree of freedom point of view does not reflect its solvability.

4. IDENTIFY LOCAL UNSATISFIABLE PARTS OF CONFIGURATIONS

In this section, we try to automatically identify additional overconstraints in the previous configurations. As our initial trial, identification process is conducted at the level of equations.

4.1. Equation-based representation of the constraint systems

Representation of constraints for configuration two

$$0.4(0. + 0.6x1) + 0.6(0.7x1 + 0.3x2) = 1 \qquad (e1)$$

$$0.4(0. + 0.6y1) + 0.6(0.7y1 + 0.3y2) = 1 \qquad (e2)$$

$$(0.5x1 + 0.5x2) = 1.5 \qquad (e3)$$

$$(0.5y1 + 0.5y2) = 1.25 \qquad (e4)$$

$$0.4\,(1.2 + 0.6\,x2) + 0.6\,(0.3\,x1 + 0.7\,x2) = 2.0 \qquad (e5)$$

$$0.4\,(0. + 0.6\,y2) + 0.6\,(0.3\,y1 + 0.7\,y2) = 1.0 \qquad (e6)$$

4.1.1. Representation of constraints for configuration six

Here we use equations to represent the 9 position constraints. They are:

$$0.125\,x1 + 0.59\,x2 + 0.26\,x3 + 0.02\,x4 = 1.0 \qquad (e1)$$

$$0.125\,y1 + 0.59\,y2 + 0.26\,y3 + 0.02\,y4 = 1.5 \qquad (e2)$$

$$0.085\,x3 + 0.63\,x4 + 0.28\,x5 = 3.5 \qquad (e3)$$

$$0.085\,y3 + 0.63\,y4 + 0.28\,y5 = -0.5 \qquad (e4)$$

$$\cdots \qquad (e18)$$

Where e1 and e2 correspond to p1 position constraint, e3 and e4 correspond to p2 position constraint. For the other constraints, we use ei to represent them orderly.

For brevity, only configuration two and configuration six are taken as examples for this method of identification.

4.2. Bipartite graph representation of the constraint systems

Representation of constraints for configuration two

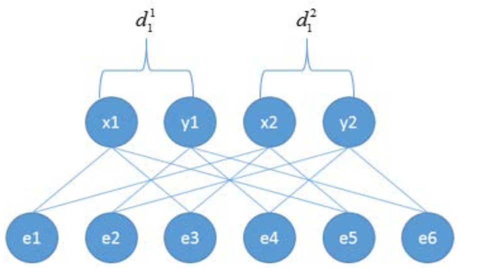


Figure 8: Bipartite graph of configuration two

Representation of constraints for configuration six

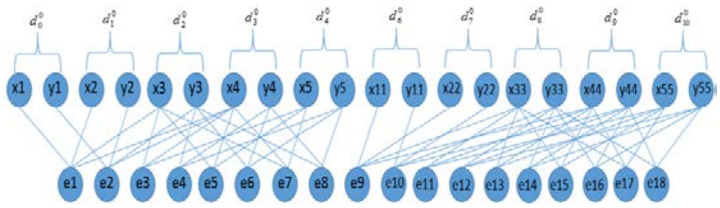


Figure 9: Bipartite graph representation of configuration six

Based on the equations described in section 4.1, bipartite graph representation of both systems is shown in Fig.8 and Fig.9, where vertices are used to represent variables and equations while edges indicate their relationship.

4.3. D-M decomposition of the systems

Decomposition of configuration two

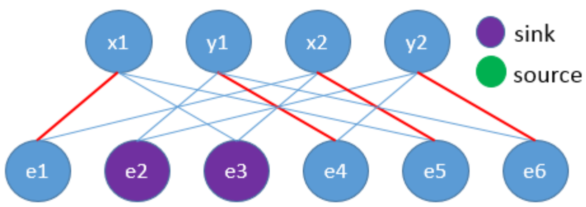


Figure 10: Maximum matching of configuration two

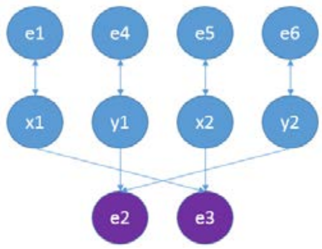


Figure 11: Associated directed graph of the over-constrained sub-graph

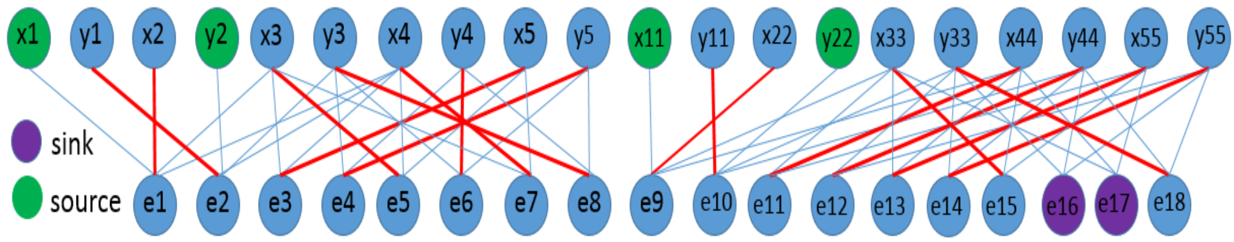


Figure 12: Maximum matching of configuration six

Decomposition of configuration six

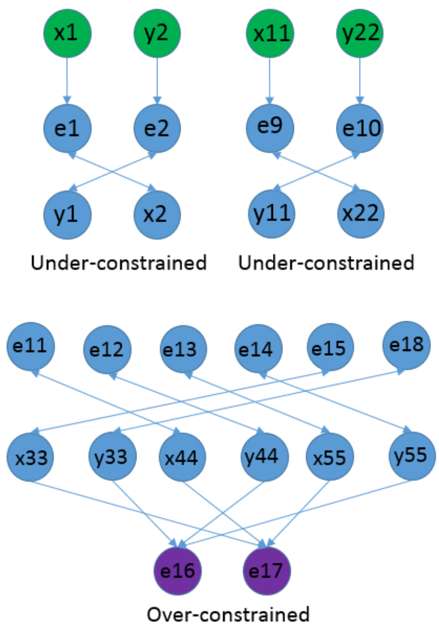


Figure 13: Associated directed graph of over-constrained/under-constrained sub-graph

Fig. 11 shows that the curve 1 is over-constrained which is consistent with the result shown in Table 2.

For configuration six, Fig.13 shows that the constraint system is divided into four parts: two under-constrained parts, one over-constrained part and the rest is well constrained. The result matches well with Table 3.

4.4. Limitation of applying D-M decomposition algorithm

The over-/under-/well-constrained sub-systems decomposed by D-M algorithm is structurally over-/under-/well-constrained. As stated in section 2.2, it is also possible for well-constrained sub-systems to have overconstraints. Here we show such examples on B-splines.

On the segment 3 of configuration seven, the constrained point is restricted to two different positions at the same time(conflicting constraints:p3 and p4); On the segment 9 of configuration eight, the constrained point is restricted to the same positions twice(redundant constraints:p8 and p9)

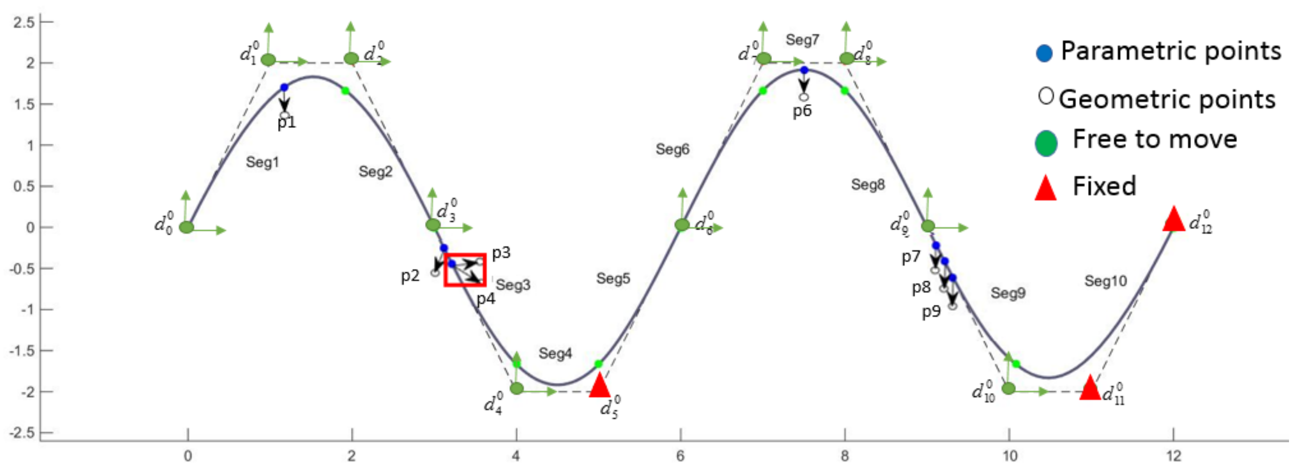


Figure 14: Configuration seven- a b-spline curve with 8 position constraints

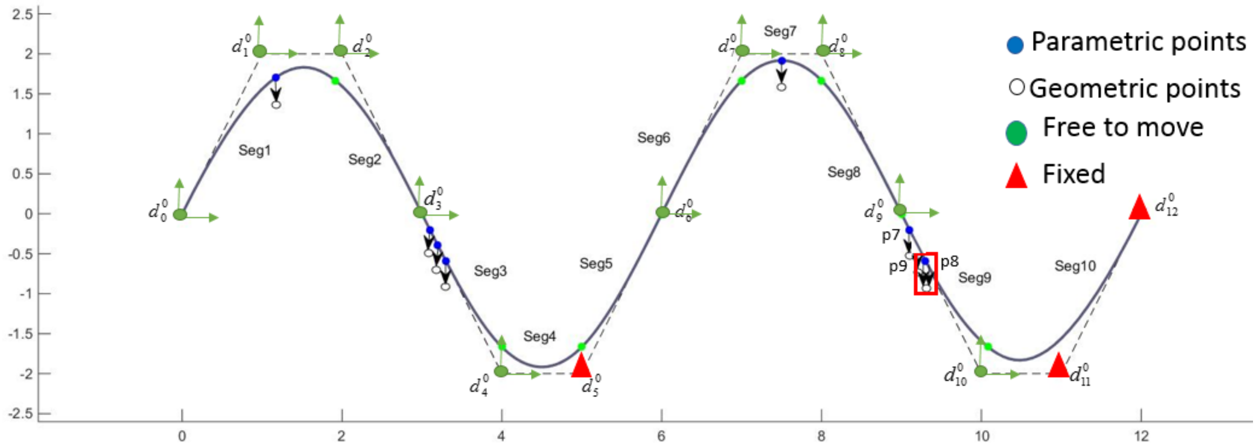


Figure 15: Configuration eight- a b-spline with 8 constraints

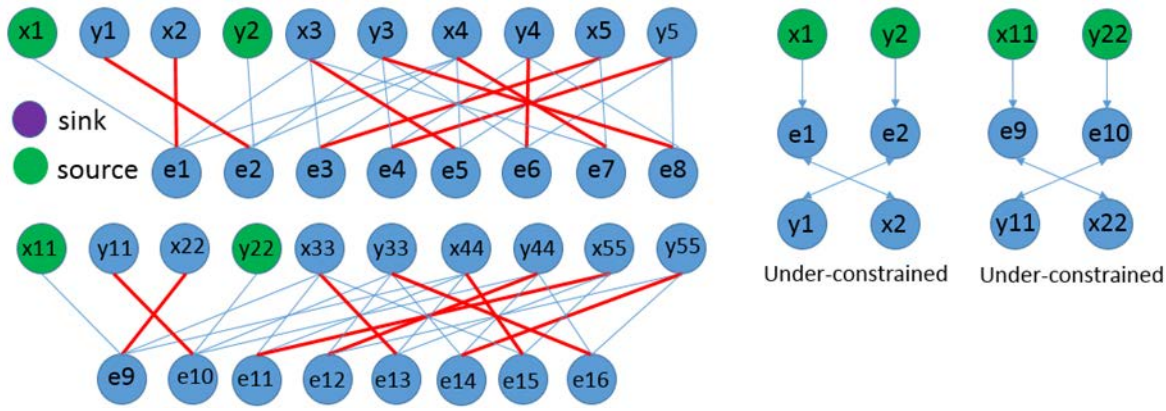


Figure 16: Sub-graphs after D-M decomposition

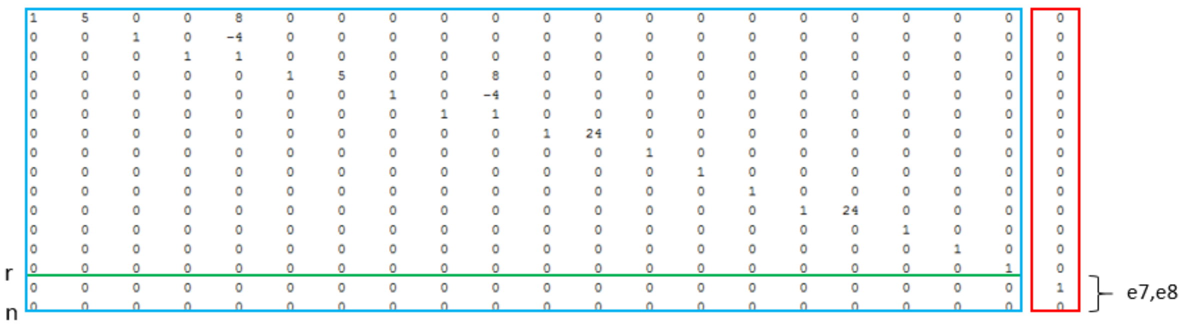


Figure 17: Analysis result of configuration seven

From Fig.16, we can know that even if we have modified several constraints in configuration seven and eight, results after D-M decomposition are the same: Seg 3 in configuration seven and Seg 9 in configuration eight are still well-constrained! This is contradictory to our modification in both configurations. In order to accurately find all the overconstraints, we resort to the algebraic methods.

5. DETECTION USING JACOBIAN MATRIX ANALYSIS

To further detect over-constraints in these two configurations (including both conflicting and redundant constraints), we apply Gauss elimination on their corresponding equation systems. As illustrated in Fig.17 and Fig.18, conflicting

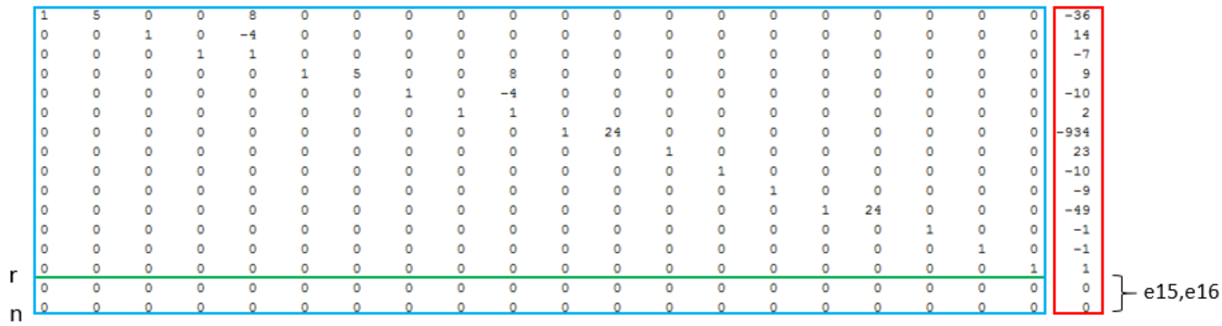


Figure 18: Analysis result of configuration eight

constraints (e7 and e8) and redundant constraints (e15 and e16) are detected.

6. CONCLUSION AND FUTURE WORK

In this paper we have tried to evaluate different methods for detecting over-constrained subparts of free-form based geometric configurations. First, considering the local support property of free-form curves, we have applied a DOF analysis and shown that it is not sufficient to detect all inconsistencies. Then, we have applied D-M decomposition to the corresponding equations systems in order to automatically detect structural over-constraints. Since these methods ignore the numerical information, inconsistent constraints may still be undetected. Therefore, we have shown that a Jacobian Matrix analysis can be used to fully discriminate conflicting and redundant constraint. However, the time complexity of the numerical analysis restricts its use to small cases when facing non-linear equations.

In the geometric constraint solving domain, a 2-step approach is often used to circumvent this, where a graph-based method is first realized to decompose the system into subparts on which algebraic solving methods are applied. In future work, we plan to adapt these approaches to the detection of inconsistent and redundant constraints.

Graph methods such as MFA are an efficient way of decomposing into irreducible subsystems[3]. However, in terms of constraint solving on free-form curves, as far as we know, there is no graph-reduction methods that are able to structurally decompose a system while considering its geometric information. A perspective could be to develop a graph-based decomposition that takes into account the local support property. Combined with an algebraic method such as witness configuration[24, 25], this could provide a general and reliable approach for the identification of overconstraints.

REFERENCES

[1] J. Graver, B. Servatius, H. Servatius, Combinatorial Rigidity, Vol. 2 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, USA, 1993.

[2] S. Foufou, D. Michelucci, J.-P. Jurzak, Numerical decomposition of geometric constraints, in: ACM Symp. on Solid and Physical Modelling, MIT, Boston, USA,ACM Press, 2005, pp. 143–151.

[3] Hoffmann C M, Lomonosov A, Sitharam M. Finding solvable subsets of constraint graphs. In: LNCS, No. 1330; 1997: 163~197.

[4] A. L. Dulmage and N. S. Mendelsohn. On the inversion of sparse matrices. Mathematics of Computation, 16(80):494–496, 1962.

[5] Christoph M. Hoffmann, Meera Sitharam, Bo Yuan Making constraint solvers more usable: overconstraint problem ,Computer-Aided Design Volume 36, Issue 4, Pages 377-399 (February 2004).

[6] Latham R S, Middleditch A E. Connectively analysis: a tool for processing geometric constraints. Computer-Aided Design, 1996(28): 917~928.

[7] D. Serrano. Constraint Management in Conceptual Design. PhD thesis, MIT, 1987.

[8] David A. Cox, John B. Little, Don O' Shea. Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra (Third Edition). Springer, 2007.

[9] Podgorelec D, Salix B, Domiter V. Dealing with redundancy and inconsistency in constructive geometric constraint solving. Advances in Engineering Software, 2008, 39(9): 770~786.

[10] Y. J. Ahn and C. M. Hoffmann. Constraint-based LN-curves. In Proc. ACM. Symp. Appl. Computing, Sierre, Switzerland, 2010.

[11] Iddo Hanniel, Kirk Haller: Solving global geometric constraints on free-form curves. Symposium on Solid and Physical Modeling 2009: 307-312.

[12] Jean-philippe Pernot, Fully Free Form Deformation Features for Aesthetic and Engineering Designs , chap 5, pp.122-152.

[13] Samy Ait-Aoudia, Roland Jegou, Dominique Michelucci Reduction of Constraint Systems ,2014.

[14] Light, R. A. And Gossard, Variational geometry: A new method for modifying part geometry for finite element analysis, Computer & Structures, Vol. 17, No. 5-6, 1983.

[15] D. Michelucci,Foufou S: Geometric constraint solving:the witness configuration method. Computer Aided Design 38(4),284-299(2006).

[16] C. Jerman, G. Trombettoni, B. Neveu, P. Mathis, Decomposition of geometric constraint systems: A survey, Internat J Comput Geom Appl, 16 (2006) 379-414.

[17] J. Owen. Algebraic solution for geometry from dimensional constraints. In Proc. Of Solid Modeling and CAD/CAM Applications, pages 397-407, 1991.

[18] Hoffmann C M, Lomonosov A, Sitharam M. Decomposition plans for geometric constraint problems, Part II. New Algorithms. Journal of Symbolic Computation, 2001,31(4): 367~408.

[19] G. Trombettoni and M. Wilczkowiak. GPDOF: a fast algorithm to decompose under-constrained geometric constraint systems: Application to 3D model reconstruction. Int. Journal of Computational Geometry and Applications (IJCGA), 16, 2006.

[20] Michelucci D, Foufou S. Interrogating witness for geometric constraint solving. In: Proceeding of SIAM/ACM Joint Conference on Geometric and Physical Modeling, 2009.

[21] Hoffmann, C. M., Lomonosov, A., Sitharam, M. (1998). Geometric constraint decomposition. In Bruderlin, Roller eds, Geometric Constraint Solving. Berlin, Springer-Verlag.

[22] Pothen, Alex and Chin-Ju Fan "Computing the Block Triangular Form of a Sparse Matrix" ACM Transactions on Mathematical Software Vol 16, No. 4 Dec. 1990, pp. 303-324.

[23] Les Piegl and Wayne Tiller,1997, The NURBS book,chapter 2,page 55, Springer-Verlag, New York.

[24] S. Thierry , P. Schreck , D. Michelucci , C. Fünfzig , J. Génevaux Computer-Aided Design , Volume 43 , Number 10 , page 1234--1249 – 2011.

[25] Moinet Mireille, Guillaume Mandil, Philippe Serré. Defining tools to address over-constrained geometric problems in Computer Aided Design. Computer-Aided Design 48(2014), 42-52.

[26] Serrano, D Automatic dimensioning in design for manufacturing Proc. Symp. Solid ModeIling Foundations and CAD/CAM Applications (1991) pp 379-386.