Ingeniería y Ciencia, ISSN 1794-9165
Volumen 2, número 3, páginas 103-137, marzo de 2006

Geometric constraint subsets and subgraphs in the analysis of assemblies and mechanisms

Oscar E. Ruiz[1] and Placid M. Ferreira[2]

Recepción: 26 de marzo de 2006 — Aceptación: 17 de mayo de 2006
Se aceptan comentarios y/o discusiones al artículo

Resumen

La habilidad del Razonamiento Geométrico es central a muchas aplicaciones de CAD/CAM/CAPP (Computer Aided Design, Manufacturing and Process Planning). Existe una demanda creciente de sistemas de Razonamiento Geométrico que evalúen la factibilidad de escenas virtuales, especificados por relaciones geométricas. Por lo tanto, el problema de Satisfacción de Restricciones Geométricas o de Factibilidad de Escena (GCS/SF) consta de un escenario básico conteniendo entidades geométricas, cuyo contexto es usado para proponer relaciones de restricción entre entidades aún indefinidas. Si la especificación de las restricciones es consistente, la respuesta al problema es uno del finito o infinito número de escenarios solución que satisfacen las restricciones propuestas. De otra forma, un diagnóstico de inconsistencia es esperado. Las tres principales estrategias usadas para este problema son: numérica, procedimental y matemática. Las soluciones numérica y procedimental resuelven sólo parte del problema, y no son completas en el sentido de que una ausencia de respuesta no significa la ausencia de ella. La aproximación matemática previamente presentada por los autores describe el problema usando una serie de ecuaciones polinómicas. Las raíces comunes a este conjunto de polinomios

[1] Ph.D. Mechanical Engineering, oruiz@eafit.edu.co, researcher, Mechanical & Industrial Eng. Dept., University of Illinois at Urbana-Champaign.
[2] Ph.D. Mechanical Engineering, pferreir@uiuc.edu, Professor, Mechanical & Industrial Eng. Dept., University of Illinois at Urbana-Champaign.

caracteriza el espacio solución para el problema. Ese trabajo presenta el uso de técnicas con Bases de Groebner para verificar la consistencia de las restricciones. Ella también integra los subgrupos del grupo especial Euclídeo de desplazamientos SE(3) en la formulación del problema para explotar la estructura implicada por las relaciones geométricas. Aunque teóricamente sólidas, estas técnicas requieren grandes cantidades de recursos computacionales. Este trabajo propone técnicas de Dividir y Conquistar aplicadas a sub-problemas GCS/SF locales para identificar conjuntos de entidades geométricas fuertemente restringidas entre sí. La identificación y pre-procesamiento de dichos conjuntos locales, generalmente reduce el esfuerzo requerido para resolver el problema completo. La identificación de dichos sub-problemas locales está relacionada con la identificación de ciclos cortos en el grafo de Restricciones Geométricas del problema GCS/SF. Su pre-procesamiento usa las ya mencionadas técnicas de Geometría Algebraica y Grupos en los problemas locales que corresponden a dichos ciclos. Además de mejorar la eficiencia de la solución, las técnicas de Dividir y Conquistar capturan la esencia física del problema. Esto es ilustrado por medio de su aplicación al análisis de grados de libertad de mecanismos.

Palabras claves: graph cycle, Groebner basis, constraint graph,mechanisms, assemblies.

Abstract

Geometric Reasoning ability is central to many applications in CAD/CAM/ CAPP environments. An increasing demand exists for Geometric Reasoning systems which evaluate the feasibility of virtual scenes specified by geometric relations. Thus, the Geometric Constraint Satisfaction or Scene Feasibility (GCS/SF) problem consists of a basic scenario containing geometric entities, whose context is used to propose constraining relations among still undefined entities. If the constraint specification is consistent, the answer of the problem is one of finitely or infinitely many solution scenarios satisfying the prescribed constraints. Otherwise, a diagnostic of inconsistency is expected. The three main approaches used for this problem are numerical, procedural or operational and mathematical. Numerical and procedural approaches answer only part of the problem, and are not complete in the sense that a failure to provide an answer does not preclude the existence of one. The mathematical approach previously presented by the authors describes the problem using a set of polynomial equations. The common roots to this set of polynomials characterizes the solution space for such a problem. That work presents the use of Groebner basis techniques for verifying the consistency of the constraints. It also integrates subgroups of the Special Euclidean Group of Displacements SE(3) in the problem formulation to exploit the structure implied by geometric relations. Although theoretically sound, these techniques require large amounts of computing resources. This work proposes Divide-and-Conquer techniques applied to local GCS/SF subproblems to identify strongly constrained clusters of geometric entities. The identification and preprocessing of these clusters generally reduces the effort required in solving the overall problem. Cluster identification can be related to identifying short cycles in the Spatial Constraint graph for the GCS/SF problem. Their preprocessing uses the aforementioned Algebraic Geometry and Group theoretical techniques on the local GCS/SF problems that correspond to these cycles. Besides improving the efficiency of the solution approach, the Divide-and-Conquer techniques capture the physical essence of the problem. This is illustrated by applying the discussed techniques to the analysis of the degrees of freedom of mechanisms.

Key words: graph cycle, Groebner basis, constraint graph,mechanisms, assemblies.

1 Introduction

In diverse problems in CAD/CAM/CAPP a set of geometric objects is presented, and a set of geometric relations between them is proposed. The goal is to obtain instances or positions of the objects which respect the proposed relations. In a more formal way, the Geometric Constraint Satisfaction or Scene Feasibility (GCS/SF) problem can be stated as follows: Let a World W be, a closed, homogeneous subset of E^3, with a set of partially or totally defined geometric entities $S = \{e_1, ..., e_n\}$ which are closed, connected subsets of W. A set of spatial relations, $R = \{R_{i,j,k}\}$ is defined/specified over pairs of entities, where $R_{i,j,k}$ is the k^{th} relation between entities i and j. The goal is to obtain either instances of every entity e_i in S, consistent with all specified relations in R, or a diagnostic of inconsistency in the set of specified relations.

The fact that GCS/SF underlies a number of problems in CAD/CAM/CAPP areas motivates this work. In fixturing, the holding of a workpiece during a manufacturing process is basically an assessment of the feasibility/consistency of a number of contact relationships between two bodies. The verification of deterministic positioning [1] of workpiece in the fixture is an analysis of the degrees of freedom of the set of contact constraints. In assembly planning the problem of feasibility of an assembly implies a study of the possible relative positions and motion between its components. In constraint-based design geometrical relations specified between entities can be viewed as one subset of the constraint set. Verification of the geometrical feasibility of the design is a GCS/SF problem. Modifications to dimensions or positions of components in the design must be compatible with the relations specified between them. Conversely, modification of these relations must be accompanied by a verification of their consistency, given the dimensions and positions of the existing objects. In tolerancing and dimensioning, tolerance relations are essentially geometric constraints. Their satisfaction implies issues such as inconsistent and redundant dimensioning, which are intrinsically scene feasibility problems. From these examples, it is evident that a strong theoretical and practical background in satisfaction of geometric constraints is crucial in CAD/CAM/CAPP applications.

Topology and Geometry are two interdependent aspects of the GCS/SF problem, though they have often been treated independently. Topology deals exclusively with the connectivity and nature of the spatial relations between entities. Geometry refers to the distances and directions that parameterize these relationships. Topologically, this work will address contact constraints. As is demonstrated in [2] contact constraints can be expressed as algebraic equalities. In contrast, other types of constraints, for example the non-invasiveness between solids, require the use of inequalities. Geometrically, this work is restricted to zero curvature (points, straight lines and planes) proper subsets of E^3.

1.1 Literature survey

Solving the GCS/SF problem implies the ability to:

1. Instance entities (or produce configurations) which satisfy the given constraints.
2. Identify a redundant constraint.
3. Determine an inconsistent set of constraints.
4. Determine the degrees of freedom between two entities.

In addition to the above capabilities, it is necessary to have reduced computational effort and a clear relation between variables used in the mathematical formulation of the GCS/SF problem and physical degrees of freedom of the entities involved. The GCS/SF problem has been be addressed in various forms, often indirectly, using: (i) numerical methods, (ii) procedural or operational approaches, and (iii) mathematical formalization.

Numerical techniques ([3], [4], [5]) essentially sample points in the solution space of the GCS/SF problem. They produce a particular answer (a set of fully instanced entities) representing a single configuration of the scene, irrespective of the multiplicity or dimension of the solution space. They only provide an incomplete answer to question 1. We emphasize the incomplete nature of such an approach because failure of the numerical method to produce an answer does not imply an empty solution space (inconsistent set of relations in the problem) as it could result from a failure of convergence of the numerical procedure. Numerical techniques, although required for determining particular configurations, do not address the questions 2, 3 and 4.

Procedural or operational techniques ([6], [7], [8]) apply intuitive algorithms to keep an account of the degrees of freedom present in the scene in the face of added constraints. Kramer, in [7], attacks the problem of Geometric Constraint Satisfaction using an algorithmic approach called degree of freedom analysis. This work concentrates on the area of kinematic analysis of mechanisms. This procedural technique sequentially satisfies the imposed constraints, placing the emphasis on the degrees of freedom of the entities. They are classified into rotational and translational, and an inventory of degrees of freedom is kept for each entity in the scene. This inventory is updated whenever a new constraint is added to the system. Although this work partially answers questions 1-4; its limitations are: (i) in many situations the separation between rotational and translational degrees of freedom is not possible; (ii) the approach encounters a large number of exceptions and attempts to deal with them on a case-by-case basis; and (iii) template solutions obtained on the basis of the topology of the constraint network cannot be re-applied to identical constraint networks under different geometrical conditions. This fact, extensively documented in [2], [8], [9], [10], [11], is due to the fact that the existence of solution spaces for the constraint equations depends upon the value of the parameters of the problem, even under identical constraint structures.

Although numerical and procedural techniques have the advantages of simplicity and computing efficiency, their lack of completeness is a serious obstacle in their applicability (especially in automated analysis environments). It is opinion of the authors that more work is needed on the mathematical formalization and solution of the GCS/SF problem before numerical or procedural techniques can be effectively used. The following paragraphs address a review of research efforts in this direction.

Questions 2-4 have not been satisfactorily answered in a systematic manner to the present because the dimension of the solution space for the GCS/SF problem is a function of both topological and geometrical conditions. In other words, manipulation of the topological part of the GCS/SF problem is not sufficient for determining the topology (degrees of freedom) of the solution.

In current literature, the GCS/SF problem has been approached from the areas of group theory [10] and kinematics and mechanisms ([9], [12]). A joint in a rigid bar mechanism is, by definition, a constraint. Therefore, historically, the study of mechanism analysis precedes constraint satisfaction problems. This multiplicity of disciplines studying the same area is manifested in the fact that the terms (trivial) constraint, joint and group are used interchangeably in the discussion.

Investigators ([3], [13]) introduced the necessary formalization for the GCS/SF problem in the form of equations of unknown positioning matrices. They proposed re-writing rules as a solution approach to the resulting system of equations. Since it is often the case that there is no closed form solution for the GCS/SF problem, re-writing rules have limited success. They guarantee a complete solution only for trivial constraint chains, discussed below. Popplestone ([14], [15]) has explored the mathematical formalization of situations involving symmetries such as arrays, hexagonal pieces, mirror arrangements, etc. Finite groups are particularly appealing in the statement of these problems.

In the context of kinematic analysis, Angeles ([9], [12]) expanded on Herve's formalization of kinematic joints in terms of the subgroups of the $SE(3)$ Group. Angeles proposed an algorithm for mobility analysis of kinematic chains whose degrees of freedom can be solely determined by the topology of the participant joints. These chains are classified into trivial and exceptional ([9], [12]). The trivial chain is constituted by sequences of joints (subgroups of $SE(3)$) whose composition is another subgroup of $SE(3)$. The exceptional chains are not, but can be reduced to, trivial ones. The method is based on application of re-writing rules from Herve's look-up tables [10]. They predict the topological structure of the composition and intersection of subgroups. The method is limited in the following aspects: (i) re-writing rules are based only on the type of joints (topology) of the chain. Therefore they ignore a variety of chains (called paradoxical), in which the topology aspect is insufficient to predict their behavior; and (ii) they do not allow the so called complex constraint networks, in which an entity may have more than two constraining relations. In addressing paradoxical chains, Angeles proposes the Jacobian method, which has the advantage of including topological and geometrical information. With this integration paradoxical and complex constraint systems can be analyzed. Based on Herve's formalization, the case of trivial constraints has been studied ([4], [16], [17], [18]) in the context of topological reduction of constraint networks. This reduction may be achieved by the application of re-write rules also used by Ambler [3] or the reduction tables by Herve [10]. Limitations of this work are the topology-only treatment, and the type of constraints (trivial) that it considers. Its contributions are (i) the methodology proposed to state the GCS/SF problem in terms of the $SE(3)$ group in the applications of assembly planning; and (ii) the separation of geometry and topology in the formulation of the problem.

Ruiz & Ferreira ([2], [19]) formulated the GCS/SF problem as one of determining the solution space of a set of polynomials. Beyond the elementary goal of solving a set of polynomials for common roots, Groebner Bases were used to characterize the algebraic set of a polynomial ideal and the properties of Groebner Bases [20] were used as a theoretical framework to respond to questions about consistency, ambiguity and dimension of the solution space. The method allowed the integration of geometric and topological reasoning. The high computational cost of Buchberger's algorithm ([11], [21]) for the Groebner Basis forced the use of a more efficient set of variables, able to express the prescribed constraints with a minimum amount of redundancy, and with a strong physical meaning. Using the group theoretic formulation of Herve for the formulation of the problem and Groebner Basis techniques for its solution, Ruiz & Ferreira were able to integrate individual advantages of Algebraic Geometry and Group Theory, therefore reducing the computational effort ([2], [19]). However, to solve larger problems, increased computational efficiency is required to make the theoretical completeness of the methods useful from the practical point of view. Therefore, the issue of lowering computational expenses is addressed in this investigation. As one moves to more complex scenarios, the structure of the problem plays a larger role in the computational costs of the solution. To exploit the problem structure this investigation uses a Divide-and-Conquer paradigm for solving complex problems. First, the problem of identifying well-constrained sets of "clusters" of entities as subproblems is addressed. Then, the aggregation of the solutions to these subproblems into the overall solution is attempted. This paper therefore represents an extension of the work in [2], [19].

This paper is organized as follows: section (2) explores previous work in which Algebraic Geometry and Group Theory complement each other to make the solution to GCS/SF a theoretically sound and physically meaningful procedure. Section (3) discusses the Spatial Constraint (SC) graph as a means of expressing the GCS/SF problem. It also explains how the partitioning of the SC graph relates to physical situations. Section (4) establishes the applicability of graph theory to the solution of the GCS/SF problem. Section (5) presents a case study in Design of Mechanisms as a GCS/SF problem. It applies the different techniques proposed and compares their performances. Section (6) offers conclusions about this work and draws lines for future research. Appendix A presents the detailed Groebner Basis results obtained in the examples.

2 Background

This section briefly reviews material on Algebraic Geometry (Groebner Basis) and Group Theory which have important consequences on the statement and solution of the GCS/SF problem. For standard properties or notation see [2], [10], [11], [17], [19] and [20].

2.1 Algebraic Geometry and the GCS/SF problem

The GCS/SF problem takes place in a world W, with a set of relations R. If a set of entities $S = \{e_1, ..., e_n\}$ satisfies the constraints, it is said that S is feasible for W and R, and this fact is written as $S = feasible(W, R)$. If the polynomial form of the problem is $F = f_1, f_2, ..., f_n$ with f_i polynomials in variables $x_1, x_2, ..., x_n$, it is said that $F = poly\_form(W, R)$. Since S is a solution for F, it is denoted as $S = solution(F)$.

Given that $F = poly\_form(W, R)$ and $S = feasible(W, R)$, F has an associated ideal $I\langle F\rangle$. For any polynomial set F, the Groebner Basis $GB(F)$ is an alternative set, which generates the same ideal $I\langle F\rangle$, but presents advantages in characterizing its solution space. For the purposes of this paper, the calculation of the Groebner Basis of a set of polynomials F can be regarded as a black box procedure whose result, $GB(F)$, has several important properties. The properties allow to draw the following propositions:

1. $S = solution(F)$ iff $S = solution(GB(F))$. This is a consequence of the fact that F and $GB(F)$ span the same polynomial ideal. In the context of the GCS/SF problem, this implies that $GB(F)$ and F describe the same scene, although $GB(F)$ presents properties useful in the solution process.

2. $1 \in GB(F) \rightarrow S = solution(F) = \phi$. If the field is algebraically closed, finding "1" or a constant polynomial in $GB(F)$ implies the equation "0=1" leading to the fact that F has no solution in that field. However, the converse proposition has to be carefully used: If $1 \in GB(F)$, a solution exists, although it might be complex. Therefore, an additional check to ensure a real solution is needed.

3. If $I\langle F\rangle$ is a Zero-dimensional ideal, then the set F (and $GB(F)$) has a finite number of solutions. Therefore $S = feasible(W, R)$ has a finite number of configurations. The zero-dimensionality of I can be assessed: A variable x is free if it does not appear as $head(p)$ for any polynomial $p \in GB(F)(p = x^d + tail(p), d \in N)$. A zero-dimensional ideal $I\langle F\rangle$ has no free variables in its polynomial basis, $GB(F)$.

4. Let a new constraint be represented by polynomial f. f is redundant to $F \leftrightarrow (1 \in GB(F \cup \{y \cdot f - 1\}))$ for a new variable y. This proposition determines whether an additional constraint is redundant by examining if the satisfaction of the new constraint f is unavoidable when the initial set of constraints is satisfied.

5. $GB(F)$ (based on a lexicographic order) is a triangular set in the sense that $GB(F)$ contains polynomials only in x_1, some others only in x_1, x_2,

and so on, making the numerical solution a process similar to triangular elimination.

These properties and propositions provide a theoretical framework for the solution of the GCS/SF problem. It can be summarized in the following macro-algorithm [2], in which the invariant clause for the loop is the existence of a set of non-redundant, consistent and multi-dimensional set of (constraint-generated) polynomials.In the event of the addition of new constraints to the scene (line 3), the algorithm converts them into polynomial(s) (line 6), and tests their redundancy by using property 4 (line 10), consistency by using property 2 (line 7) and zero-dimensionality of the accumulated set of constraint-based polynomials by property 3 (line 15). If the new constraint is redundant, it is ignored (line 11). In the other two cases the invariant becomes false and the loop breaks. If the ideal has become zero-dimensional a triangular Groebner Basis under some stated lexicographic order is extracted and solved by using property 5 (line 24). Property 1 is the underlying basis of the algorithm, since it establishes that the $GB(F)$ faithfully represents F, with the same roots and ideal set.

0 {Pre: W a fixed scenario}
1 $F = \{\}$
2 $GB_t = \{\}$
3 do new relation R_i
4 {Inv: F is consistent, non–redundant, non–zero–dimensional}
5 $R = R \cup \{R_i\}$
6 $f = poly\_form(W, R_i)$
7 if $(1 \in GB_t(F \cup \{f\}))$ then
8 stop (system is inconsistent)
9 else
10 if $(f \in Radical(F))$ then
11 skip(f is redundant)
12 else
13 $F = F \cup \{f\}$
14 $GB_t = GroebnerBasis(F, <_t)$
15 if $(ZeroDimension(GB_t))$ then
16 break loop
17 else

skip (next relation–constraint)
fi
fi
fi
od
$GB_l = GroebnerBasis(F, <_l)$
$S = triangular\_solution(GB_l)$
{Post: $R = \{R_i\}$ a set of relations; $S = feasible(W, R)$}

In [2] the theoretical completeness of this formulation was demonstrated. However, two problems were detected in the initial approaches to the problem: (i) the set of variables used did not have a direct relation with the degrees of freedom of the entities, therefore impeding the interpretation of the resulting Groebner Bases in terms of scene configuration; and (ii) the large computational complexity [11] of Groebner Basis was compounded by the large number of variables used in the formulations. In order to address the issue of computational expenses and the need for a geometrically meaningful statement for GCS/SF, a Group-theoretical approach was adapted from previous investigations. Next section addresses the results of such efforts.

2.2 Group-theoretic formulation for the GCS/SF problem

This section examines the modeling of the GCS/SF problem by using the canonical form of conjugation classes developed by Herve [10] and the application of his work by several authors ([2], [9], [17]). The set of Euclidean displacements in 3D, $SE(3)$, is a (non commutative) group ([22], [23]) with the composition operation ($\circ$). $SE(3)$ presents subsets which are groups themselves, and which express certain common classes of displacements. They are called subgroups. For example, the subgroup of the rotations about a given axis u in the space, R_u, is a subset of $SE(3)$, and a group itself. Given A, B, **subgroups** of the Euclidean group $SE(3)$, A is a **conjugate** of B ($A \sim B$) iff $\exists\, T \in SE(3)$ such that $A = T^{-1}BT$. The relation $A \sim B$ is an equivalence relation. It is symmetric, reflexive and transitive. It defines equivalence classes called conjugation classes. Conjugation classes have a canonical subgroup which represents any other subgroup in the class by applying a transformation T for a change of basis. A list of the conjugation classes for the subgroups of $SE(3)$ and their canonical representation [10], as well as their degrees of freedom is shown in table (1). In this table, $twix(\theta)$ means a rotation about the X axis by θ, $XTOY$ means a rotation by 90° about the Z axis, and $trans(x, y, z)$ indicates a general spatial translation. The concept of equivalence (conjugation) allows naming certain displacements in $SE(3)$ as "linear translations", "rotations", "planar slidings", etc, therefore making the link between subgroups of $SE(3)$ and kinematic constraints. For example, "rotations" are all transformations of the form

$$R_u(\theta) = B.R_w(\theta).B^{-1} = B.twix(\theta).B^{-1},$$

with $B \in SE(3)$ and $R_w(\theta) = twix(\theta)$ being the canonical representation of the conjugation class of rotations. The displacement B represents the geometric part of a particular constraint, while the canonical part contains the topological information; the number and type of degrees of freedom.

Table 1: Conjugation classes and their canonical forms

dof	Symbol	Conjugation Class	Canonical Subgroup
1	R_u	Rotations about axis u	$\{twix(\theta)\}$
1	T_u	Translations along axis u	$\{trans(x, 0, 0)\}$
1	$H_{u,p}$	Screw movement along axis u, with pitch p	$\{trans(x, 0, 0).twix(p.x)\}$
2	C_u	Cylindrical movement along axis u	$\{trans(x, 0, 0).twix(\theta\}$
2	T_p	Planar translation parallel to plane P	$\{trans(0, y, z)\}$
3	G_p	Planar sliding along plane P	$\{trans(0, y, z).twix(\theta)\}$
3	S_o	Spherical rotation about center $O = (0, 0, 0)$	$\{twix(\psi).XTOY.twix(\phi).$ $XTOY.twix(\theta)\}$
3	T	3D translation	$\{trans(x, y, z)\}$
3	$Y_{v,p}$	Translating Screw axis v, pitch p	$\{trans(x, y, z).twix(p.x)\}$
4	X_v	3D translation followed by rotation about v	$\{trans(x, y, z).twix(\theta)\}$

A constraint between two entities by definition maintains invariant certain relations between the constrained entities. For example (see table 1), a planar sliding, G_p, allows 2 translational and 1 rotational degree of freedom, while still ensuring planar contact between the two parts. A rotational constraint, R_u, preserves axial and radial relative distances, allowing 1 angular degree of freedom between the constrained entities.

Using this methodology, the contact constraints addressed in this investigation are specified as shown in table (2). For example, a P-ON-PLN relation confines a point to be on a plane, therefore configuring a 5-dof constraint. It includes 2 dof related to the position of the point on the plane (Tp), and 3 dof, corresponding to the orientation (S) of the frame attached to the point (points are in the origin of their attached frame; lines coincide with the X axis of their frame and planes coincide with the Y-Z plane of their attached frame). These (matrix) equations allow for the construction of the polynomial form of the GCS/SF problem. The methodology for this modeling is discussed next.

Table 2: Entity relations in the form of kinematic joints

Macro	Joint chain	Kinematic joints in chain	dof
P-ON-P	S	spherical	3
P-ON-LN	$T_v \circ S_o$	linear translation, spherical	4
P-ON-PLN	$T_p \circ S_o$	planar translation, spherical	5
LN-ON-LN	C	cylindrical	2
LN-ON-PLN	$T_p \circ R_v \circ R_w$	planar translation, revolute	4
PLN-ON-PLN	$T_p \circ R_v$	planar translation, revolute	3

The GCS/SF problem is stated as a series of constraints R_i relating F_{i1} with F_{i2} as shown in figure (1) (corresponding to a two body system), where F_{ij} is the i^{th} feature of body B_j. The $R_i()$ constraints are in general composed by translations $T()$ and rotations $Rot()$, as dictated by tables (1) and (2). Body B_1 contains two features, whose frames are F_{11} and F_{21}. The corresponding features in body B_2 are F_{12} and F_{22}. The goal is to find a final position of B_1 (assuming B_2 stationary), such that F_{11} relates to F_{12} and F_{21} relates to F_{22} satisfying the invariance dictated by $R_1()$ and $R_2()$ respectively. The final position of B_1 must be such that feature frames F_{11} and F_{12} differ exactly in the orientation and position changes allowed by constraint $R_1()$. The same should be true for F_{21} and F_{22} with regard to $R_2()$. The equations expressing the facts above are

$$B_1 \cdot F_{11} \cdot R_1() = B_2 \cdot F_{12} \text{ and } B_1 \cdot F_{21} \cdot R_2() = B_2 \cdot F_{22}\,. \tag{1}$$

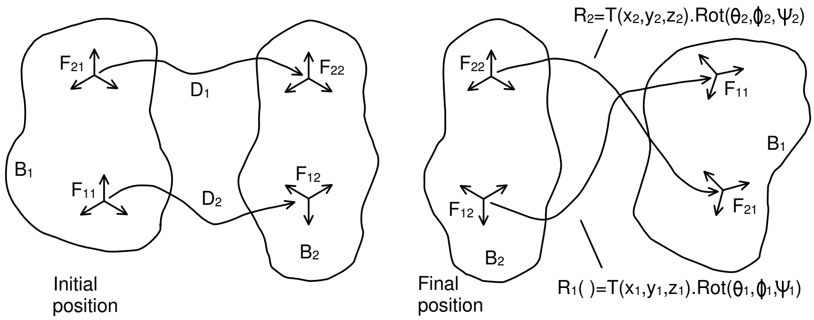

Figure 1: Two body example. Canonical variable modeling of the GCS/SF problem

The above procedure can be generalized to the case in which there are several relations (constraints) $R_i()$ specified among bodies. Once the constraint equations are obtained by this procedure, the construction of the Groebner Basis and its interpretation are carried out in the manner described by the constraint management algorithm discussed in last section. This formulation of the problem produced significant savings in computational effort when compared to a formulation obtained by trying to directly obtain a transformation for each body in a world coordinate frame (see [2] for details). Further information on the group theoretic formulation of such problems appear in [2], [10] and [16].

3 Partitioning of the GCS/SF problem

We have, thus far, outlined a problem formulation based on the underlying group structure of displacements and a general solution procedure based on Groebner Basis construction. In this section, we present a scheme that attempts to exploit structures that might be present in particular instances of a GCS/SF problem by a Divide-and-Conquer Technique [17]. The discussion will be illustrated with an example of a mechanisms; the Cartesian, or $X-Y$ table. The mechanism is expressed in the form of a set of bodies with constraints between them. The goal of the exercise is to determine the degrees of freedom of the design. Other examples of GCS/SF in the area of Mechanism Design and Analysis can be found in [19].

The Cartesian table (see figure 2) is intended to produce two translational degrees of freedom, thereby producing a planar translation between bodies B_4 and B_5. The constraints in the problem are shown in table (3). The features F_{ij} involved in each C_k appear in column 3, while the sequences of compositions of subgroups of SE(3) for each constraint C_k appear in the column 4. Notice that this example includes non-trivial constraints such as C_1, C_2, C_3 and C_4.

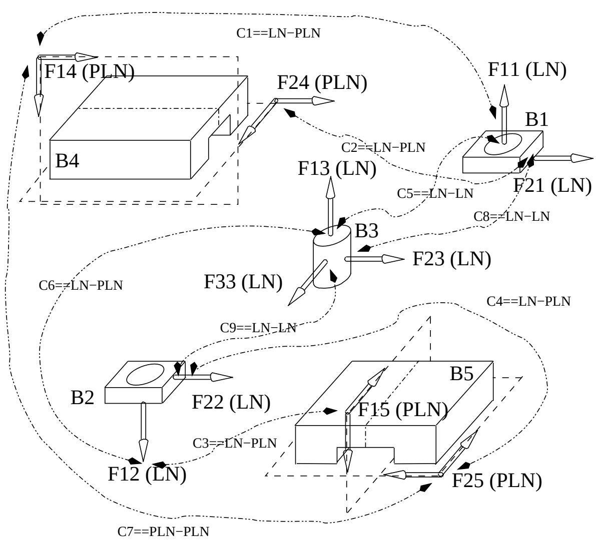

Figure 2: Piece disassembly of cartesian table

With the specified constraints, the bodies B_1, B_2 and B_3 have zero degrees of freedom relative to each other. This fact, together with constraints C_1, C_2, C_3 and C_4, forces the planes F_{15} and F_{14} to remain perpendicular to each other. An additional G_p (planar sliding) constraint forces planes F_{25} and F_{24} to remain in contact, therefore producing the desired $X-Y$ movement.

Table 3: Joint list of the cartesian table

Constraint	Constraint Type	Elements	Canonical Representation
C_1	LN-PLN	$F_{11},\ F_{14}$	$R_u(\theta_1) \circ T_p(y_1, z_1) \circ R_u(\phi_1)$
C_2	LN-PLN	$F_{21},\ F_{24}$	$R_u(\theta_2) \circ T_p(y_2, z_2) \circ R_u(\phi_2)$
C_3	LN-PLN	$F_{12},\ F_{15}$	$R_u(\theta_3) \circ T_p(y_3, z_3) \circ R_u(\phi_3)$
C_4	LN-PLN	$F_{22},\ F_{25}$	$R_u(\theta_4) \circ T_p(y_4, z_4) \circ R_u(\phi_4)$
C_5	LN-LN	$F_{13},\ F_{11}$	$C_u(\theta_5, x_5)$
C_6	LN-LN	$F_{13},\ F_{12}$	$C_u(\theta_6, x_6)$
C_7	PLN-PLN	$F_{14},\ F_{25}$	$G_p(\theta_7, y_7, z_7)$
C_8	LN-LN	$F_{23},\ F_{21}$	$C_u(\theta_8, x_8)$
C_9	LN-LN	$F_{33},\ F_{22}$	$C_u(\theta_9, x_9)$

The SC graph, presented in figure (3), conveys the topological and geometrical information of the GCS/SF problem. This representation allows: (i) a very clear formulation of the problem; (ii) a systematic way, suitable for computer generation of the equations governing the degrees of freedom of the entities involved and; most importantly (iii) the identification of subproblems which help in the solution of the GCS/SF problem, by allowing the application of preprocessing techniques.

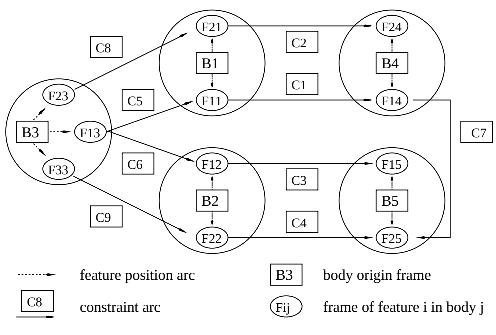

Figure 3: Graph of spatial constraints for cartesian table

Conventions: Since entities are represented by frames, the terms entity and frame are equivalent. In the SC graph the nodes are entity frames (B_j and F_{ij}). The arc between two nodes represents the displacement that relate the corresponding entity frames. There are three types of nodes; nodes B_j, which represent the origin frame of a body in the World Coordinate System, feature nodes F_{ij}, which represent the feature i in body B_j and body nodes that include the origin frame of the body and its features. Conceptually, there are two types of arcs: positioning and constraint arcs. Positioning arcs represent known relative positions of features within bodies. They always join an entity B_i and one of its features F_{ji}. Constraint arcs always connects two feature nodes, which may be joined by more than one arc to admit more than one constraint between them. The constraint arcs are represented by $C_i(x_j, \theta_m, ...)$, with the degrees of freedom $x_j, \theta_m ...$ sometimes being omitted. To simplify the notation, positioning arcs are named F_{ji}, as the features themselves, and the body nodes are named as their origin frame, B_j.

3.1 Partitioning of the Spatial Constraint Graph

Regardless of the methodology used for solving the polynomial form of the GCS/SF problem, the complete set of constraints has to be considered in the solution process. At the same time, given the costly symbolic processing required in the production of a Groebner Basis, redundancy in the constraint-based polynomial set must be avoided. Observing the SC graph of figure (3), it is clear that each cycle in the graph produces a constraint equation for the GCS/SF problem. For example, the cycle involving constraints C_5 and C_8 leads to the following equation

$$F_{23}.C_8.F_{21}^{-1} = F_{13}.C_5.F_{11}^{-1} \, . \tag{2}$$

The cycle-based equation (2) represents the connectivity of a subgraph of the SC graph. Therefore, it is relevant to determine a set of small cycles while still capturing the complete connectivity of the SC graph. A basic set (also called fundamental) of cycles in a graph presents such properties; every other cycle in the graph can be expressed as the ring sum ([24], [25], [26]) of cycles of this set. At the same time, no cycle of the basic set can be expressed in terms of the other cycles of such a set. These two conditions render a complete and non-redundant coverage of the SC graph. Hence, the equations generated by a basic set of cycles of the SC graph are a set of equations that completely and non-redundantly express the topology of the GCS/SF problem.

Well known results ([24], [25], [26]) in graph theory indicate that (i) the set of basic cycles is not unique, and (ii) any such a set contains exactly $|E|-|V|+1$ cycles. Since the set is not unique, it is possible to generate several alternative sets of equations or formulations for the GCS/SF problem. This investigation proposes a partition of the SC graph into cycles that represent easily solvable GCS/SF subproblems. This partition represents the Divide stage of the Divide-and-Conquer strategy presented. The GCS/SF problem decomposition requires the generation of subproblems which are highly constrained since they are associated with ideals of low dimensionality ([2], [20]). Two remarks are relevant at this point: (i) low dimensional ideals are less expensive to calculate since the time complexity of Buchberger's algorithm is doubly exponential in the dimension of the ideal represented by the polynomials in the base [27]; and (ii) in some domains of application, such as assembly planning, low dimensional ideals are associated to self contained subassemblies. Therefore such a partitioning of the GCS/SF problem presents direct applications in CAD/CAM environments. Since high dimensional ideals are usually related to long compositions of constraints, and to expensive computations, a desirable goal is to identify small cycles in the SC graph, with short chains of constraints, which lead to low dimensional ideals and less expensive computations.

Figure (4) illustrates several elementary graph theoretic concepts ([24], [25], [26]) related to the SC graph of figure (3). Figure (4a) presents a simplified version of the graph, in which each node represents the basic body frame and its feature frames. The graph $SC = (V, E)$ therefore presents $|V|$ nodes and $|E|$ arcs. Figure (4b) shows a spanning tree for the graph. Figure (4c) relates the cords (edges not in the spanning tree in 4b) with the cycles shown in figure (4d), which presents a fundamental (or basic) set of cycles for the SC graph. Each cord produces exactly one of such cycles when added to the spanning tree.

The construction of a basic set of cycles for a graph can be achieved by obtaining a spanning tree T and the set of corresponding cords (sometimes called cotree T'). Each cord c_i when added to T, produces one and only one cycle. Since exactly $|E|-|V|+1$ cycles are needed and there exist $|E|-|V|+1$ cords, it follows that the set of cycles obtained in this way serves as a basis for the set of circs (and therefore cycles) of the graph. Obviously, the equations

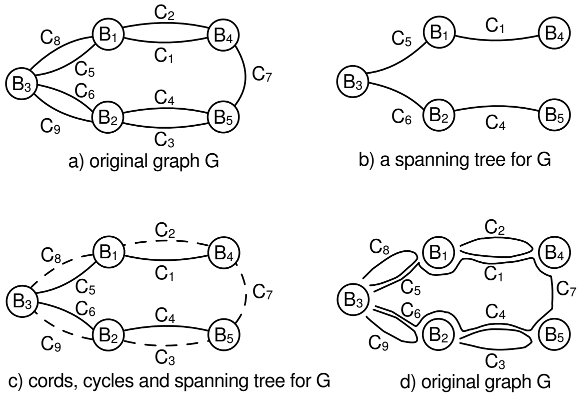

Figure 4: Spanning tree and basic cycles for the constraint graph of the cartesian table

for the GCS/SF problem only need to be written for the cycles which form the basis for the SC graph; any other set of equations can be written as a linear combination of the equations for the set of basic cycles.

The algorithms used for the decomposition of the SC graph are well known in graph theory, and therefore not explicitely included here. The first algorithm determines a spanning tree T from a graph G. In a spanning tree T every cord completes a cycle that, in the worst case, has length $2H+1$, where H is the depth of the tree. Therefore, by using a low-depth spanning tree, the largest cycle length is limited. A heuristic strategy is used to obtain a low-depth tree [17]. The second algorithm uses a given spanning tree T and its cotree T' to obtain the corresponding set of basic cycles. For the SC graph of the Cartesian table this set contains four cycles of length 2, and one cycle of length 5 (see figure 4d). The SC graph presents $|V| = 5$ nodes (entities) and $|E| = 9$ edges (constraints). Since the set of basic cycles must have $|E| - |V| + 1 = 5$ cycles, it follows that, since the set is basic, it constitutes a basis for the set of circs (and cycles) of the graph. For this example, the algorithms [17] partition the GCS/SF problem into subproblems that correspond to the following set of basic cycles

$$SBC = \{\{C_1 - C_2\}, \{C_3 - C_4\}, \{C_6 - C_9\}, \{C_8 - C_5\}, \{C_5 - C_1 - C_7 - C_3 - C_6\}\}\,. \quad (3)$$

The matrix equations describing the constraint chains for each cycle appear in table (4).

Table 4: Constraint graph basic cycles

Cycle name	Cycle equations
$C_1 - C_2$	$F_{11}.C_1.F_{14}^{-1} = F_{21}.C_2.F_{24}^{-1}$
$C_3 - C_4$	$F_{12}.C_3.F_{15}^{-1} = F_{22}.C_4.F_{25}^{-1}$
$C_6 - C_9$	$F_{13}.C_6.F_{12}^{-1} = F_{33}.C_9.F_{22}^{-1}$
$C_8 - C_5$	$F_{23}.C_8.F_{21}^{-1} = F_{13}.C_5.F_{11}^{-1}$
$C_5 - C_1 - C_7 - C_3 - C_6$	$C_5.C_1.C_7 = C_6.C_3.F_{15}^{-1}.F_{25}$

At this point, in the context of the Cartesian table example, a partition of the original GCS/SF problem –using a basic set of cycles for the SC graph– has been determined. The next section will use such a partition in alternative solution procedures for the problem.

4 Problem modeling and solution techniques

This section discusses the method of solution proposed for the GCS/SF problem. Next section applies them to the Cartesian table example.

4.1 Brute force approach

The initial strategy for dealing with the GCS/SF problem, called brute-force here, implies the determination of the set of equations which convey all the connectivity information of the corresponding SC graph. This approach uses the set of basic cycles of the SC graph to merely state a complete and non redundant set of simultaneous equations. The polynomials contributed by all the cycles in the basic set are put together in a set input to a Groebner Basis algorithm (Maple and/or Mathematica were used for this purpose) along with constraints that specify relationships between the parameters used in the canonical representations of the contact constraints (for example, a rotational constraint might produce a sin and cosine of an angle). Although the partition of the SC graph plays a role in the divide-and-conquer techniques, discussed later, it is also a requisite for the statement of the polynomial form of the GCS/SF problem.

4.2 Divide-and-Conquer algorithm

The Divide-and-Conquer algorithms introduced in this investigation assume the existence of a fundamental set of basic cycles for the *SC* graph. For each cycle (or loop) L_i (lines 2, 4) the algorithm extracts the polynomial equations and calculates its Groebner Basis gb_i (line 5). The equations for each cycle have the form of equation (2). In the algorithm they are denoted as $equations(L_i)$. The equations obtained in this way are put together into the set *full_equations* (line 7), whose Groebner Basis is finally calculated. Obviously, if any one of the g_i sets shows any inconsistency ($gb_i = \{1\}$), the process should stop (line 9).

procedure_Divide_and_Conquer(G set of graph)
0 {Pre: $G = \{L_1, L_2, ..L_k\}$ basic cycles in Spatial Constraint graph}
1 $full\_equations = \{\}$;
2 do $not\_empty(G)$
3 {Inv:$full\_equations$ has same roots as $\{L_1, L_2, ..., L_i\}$}
4 $Li = next\_cycle(G)$;
5 $gb_i = GB(equations(L_i), <_l)$;
6 if $(gb_i \neq \{1\}) \rightarrow$
7 $full\_equations = full\_equations \cup gb_i$;
8 else $\rightarrow$
9 exit;
10 fi
11 $G = G - \{L_i\}$;
12 od
13 $full\_GB = GB(full\_equations, <_l)$;
14 {Post: $full\_GB$ is the Groebner Basis for $equations(G)$}

The rational behind the partition technique just discussed lies in several facts; (i) the individual gb_i are (reduced) Groebner Bases for the polynomials representing each basic cycle; therefore they have no internal redundancy; (ii) local inconsistencies are filtered before the full GCS/SF problem is addressed; (iii) local solutions to subproblems can be found and used towards the solution of the full problem, and (iv) the gb_i sets represent an already (triangularly) ordered set of polynomials. Although it is not within the scope of this investigation to examine the details of Groebner Basis calculation, it is possible that in later work the pre-ordering in the individual Groebner Bases could be exploited to speed up the processing of the full set.

4.3 Incremental-instancing algorithm

The Incremental-Instancing (II) method is a variant of the Divide-and-Conquer technique, in which variables that can be given a value by the characteristics of the local constraint scenario are instanced immediately, therefore progressively reducing the size of the variable and polynomial sets.

This algorithm maintains a set named *instanced_variables* which contains the variables that have taken a value at any point in the execution. Subsequently, only variables not contained in this set can be considered for Groebner Basis calculation (lines 8, 9). If a Groebner Basis is successfully calculated for a cycle (line 9), the set of instanced variables is augmented by its contribution (line 10), and the general set of polynomials, *full_equations* is augmented by the partially instanced version of its set of polynomials gb_i (line 11). When the solution of the overall GCS/SF problem is finally attempted, only the free variables and the instanced version of the individual Groebner Bases gb_i are used (lines 17-19).

procedure Incremental_Instancing (G set of graph)
0 Pre: $G = L_1, L_2, ..., L_k$ basic cycles in Spatial Constraint graph
1 $full\_equations = \{\}$;
2 $free\_variables = \{\}$;
3 $instanced\_variables = \{\}$;
4 do $not\_empty(G)$
5 {Inv: $full\_equations$ has same roots as $\{L_1, L_2, ..., L_i\}$}
6 $L_i = next\_cycle(G)$;
7 $V_i = variables(L_i) - instanced\_variables$;
8 $gb_i = GB(equations(L_i), V_i, <_l)$;
9 if $(gb_i \neq 1) \rightarrow$
10 $instanced\_variables = instanced\_variables \cup instanced\_vars(V_i, gb_i)$;
11 $full\_equations = full\_equations \cup instanced\_form(gb_i, instanced\_variables)$;
12 else $\rightarrow$
13 *exit*;
14 fi

```
    G = G − {L_i};
od
free_variables = all_variables(G) − instanced_variables;
full_equations = instanced_form(full_equations, instanced_variables);
full_GB = GB(full_equations, free_variables, <_l);
{Post: full_GB is the Groebner Basis for equations(G)}
```

5 The GCS/SF problem in design and analysis of mechanisms

5.1 Brute-force procedure

The brute-force approach consists of the construction of a polynomial set which contains all the polynomials originating from the cycle-matrix equations in table (4). The set is shown in appendix A, equation (4), together with its lexicographical Groebner Basis, equation (5). No partial or intermediate solutions are used in this case.

By applying the methodology and algorithms developed ([2], [17]) and summarized in previous sections, the following conclusions can be drawn: (i) the Ideal is not zero-dimensional (because the head terms of all the polynomials are not pure powers of some variable and all the variables are not accounted for in the head terms); (ii) the table is restricted to a planar translation, T_p with two degrees of freedom $T_p(y_7, z_7)$ –the two variables missing in the head terms– and (iii) the subassembly $B_1-B_2-B_3$ still keeps one degree of freedom (z_4) when all the other objects in the space are positioned. It can move along the line intersecting planes F_{15} and F_{14}. Although in real machine tool design such a degree of freedom is unrealistic, in this example, it has the capability to demonstrate that the confinement of the subassembly $B_1-B_2-B_3$ onto a plane F_{25} is not a necessary condition for the cartesian movement of the table. In more general terms, this result demonstrates the need for a formal degree of freedom analysis although the problem illustrated may be apparently simple.

5.2 Divide-and-Conquer procedure

This section presents the results of the preprocessing (Divide-and-Conquer) applied to the individual cycles presented in table (4). By observing the figure (2) and considering the constraints in cycles C_1-C_2, C_3-C_4, C_5-C_8 and C_6-C_9 it is evident that the constraint intersections represented by these cycles are indeed reducible, and the resulting constraints are as shown in table (5), where I_4 is the neutral element in the group $SE(3)$, and indicates a null displacement. Their reduction cannot be guaranteed by techniques of group intersection or composition because of the non-triviality of the constraints involved. It will be shown here that the results in table (5) (column 4) can be obtained in a local preprocessing of the constraints by using Groebner Basis, and by the application of the relations, established [2], between the properties of the Groebner Basis, and the solutions for the GCS/SF problem. The application of the Divide-and-Conquer strategy to the Cartesian table problem follows (The lexicographic Groebner bases for each subproblem are given in appendix A):

Table 5: Topological basic cycle reductions

Cycle	Path 1	Path 2	Reduced constraint	Defining geometry
C_1-C_2	$C_1 = F_{11}-ON-F_{14}$	$C_2 = F_{21}-ON-F_{14}$	G_p	F_{14}
C_3-C_4	$C_3 = F_{12}-ON-F_{15}$	$C_4 = F_{22}-ON-F_{15}$	G_p	F_{15}
C_5-C_8	$C_5 = F_{13}-ON-F_{11}$	$C_8 = F_{23}-ON-F_{21}$	I_4	–
C_6-C_9	$C_6 = F_{13}-ON-F_{12}$	$C_9 = F_{33}-ON-F_{22}$	I_4	–

Local preprocessing. Cycle C_1-C_2. The simultaneous enforcement of the two $LN-ON-PLN$ constraints C_1 and C_2 should produce a (trivial) constraint of the type G_p, planar sliding. This can be understood by realizing that non-colinear lines F_{11} and F_{21} of body B_1 have to simultaneously lie on plane F_{14} of body B_4. It is expected that the following procedure will confirm this intuitive conclusion.

By using the cycle equations shown in table (4) for cycle C_1-C_2, and a lexicographic order, the triangular basis is calculated. It can be inferred that y_2,z_2,$C\phi_2$ are free variables since they appear in no polynomial p as the head term, i.e., $head(p)$. Consistently, the result of this preprocessing indicates that angular degrees of freedom θ_1 and θ_2 are lost. The degrees of freedom left represent the planar sliding $G_p(\phi_2, y_2, z_2)$, as predicted in table (5).

Local preprocessing. Cycle $C_3 - C_4$. From table (4) and figure (2) it is apparent that the cycle $C_3 - C_4$ presents a situation identical to cycle $C_1 - C_2$. By using the cycle equations shown in table (4) for cycle $C_3 - C_4$, and a lexicographic order, the corresponding Groebner Basis is calculated. The free variables, z_4, y_4 and $C\phi_4$, are left in the constraint $G_p(z_4, y_4, \phi_4)$. As in the previous case, the cycle would not be reducible by a topology-based re-writing strategy for trivial constraints.

Local preprocessing. Cycle $C_5 - C_8$. The satisfaction of constraints C_5 and C_8 implies that lines F_{13} and F_{23} of body B_3 have to be respectively placed on (perpendicular) lines F_{11} and F_{21} of body B_1. This geometric condition (perpendicularity) suppresses all degrees of freedom of the cycle. As before, this conclusion can be extracted from the Groebner Basis for the polynomials corresponding to this cycle. In this case, no variables are left free; and effectively bodies B_1 and B_3 have their relative movement completely constrained.

Local preprocessing. Cycle $C_6 - C_9$. As in the case of the cycle $C_5 - C_8$ one expects that all movement be restricted between bodies B_3 and B_2. The (triangular) Groebner Basis shows the zero-dimensionality of this ideal; therefore all the variables are instanced, and bodies B_3 and B_2 are rigidly attached.

Local preprocessing. Cycle $C_5 - C_1 - C_7 - C_3 - C_6$. Although this cycle was determined as part of the basic set of cycles in the SC graph, the number of constraints (5) that it involves makes it unattractive for calculation of its Groebner Basis. The reason is that its potential for high dimensionality makes its preprocessing very expensive. The alternative followed was to simply include its original cycle equations in the calculation of the full-graph Groebner Basis, instead of their Groebner Basis. In such a case, the rest of the constraint equations lower the dimensionality of the ideal, making its processing feasible.

Global processing. Full Graph. The (g_i) Groebner Bases already calculated for the individual cycles $\{gb_{1-2}, gb_{3-4}, gb_{5-8}, gb_{6-9}\}$ are used towards the calculation of the Groebner Basis for the whole constraint graph, together with the original cycle equations for cycle $C_5 - C_1 - C_7 - C_3 - C_6$. The same variable order was used as for the brute-force approach. As expected, the Groebner Basis obtained is the same as in equation (5); therefore, it is not presented again.

Table (6) presents the statistics for the application of the Divide-and-Conquer and brute-force techniques. It is found that the Divide-and-Conquer techniques are able to lower the computational expense of the problem, while guaranteeing the correctness of the results.

Table 6: Statistics for the ct example. Divide-and-Conquer strategy

Problem	variables	equations	GB size	time (secs)
Total brute force	40	73	40	107,4
C_1-C_2	12	16	9	1,8
C_3-C_4	12	16	9	2,0
C_5-C_8	6	14	6	0,6
C_6-C_9	6	14	6	1,0
Full graph	40	43	40	54,3
Total D & C				59,9

5.3 Incremental-instancing procedure

According to the incremental-instancing algorithm presented in previous sections, the sequence of cycles considered in the execution is presented in table (7). Cycle C_1–C_2 produces an instancing of variables $C\theta_2$, $S\theta_2$, $S\theta_1$ and $C\theta_1$. This result confirms the fact that two rotational degrees of freedom are lost in this cycle. Cycle C_3–C_4 presents a similar situation for variables $S\theta_3$, $C\theta_3$, $S\theta_4$ and $C\theta_4$, and so on. Notice that, in general, the order in which the cycles are considered is significant if they share variables (line 8 of the incremental-instancing algorithm). In that case, a variable instanced in a processed cycle would become a constant for the later stages of the algorithm. In this particular example the first four cycles considered do not have variables in common among themselves. Therefore they do not influence each other. The last cycle, C_5–C_1–C_7–C_3–C_6, shares variables with the ones previously considered. The comparison between tables (6) and (7) indicates that the advantage of the incremental-instancing technique is present in the manipulation of the full set of equations. This is so because at that stage the set of variables has been reduced by the incremental-instancing.

Table 7: Statistics for incremental-instancing algorithm

Subgraph	Instanced values	# vars	equations	GB size	time (secs)
C_1-C_2	$C\theta_2 \rightarrow 1$	12	16	9	1,8
	$S\theta_2 \rightarrow 0$				
	$S\theta_1 \rightarrow -1$				
	$C\theta_1 \rightarrow 0$				
C_3-C_4	$S\theta_3 \rightarrow 0$	12	16	9	2,2
	$C\theta_3 \rightarrow 1$				
	$S\theta_4 \rightarrow -1$				
	$C\theta_4 \rightarrow 0$				
C_5-C_8	$x_5 \rightarrow 1$	6	14	6	0,7
	$x_8 \rightarrow 2$				
	$S\theta_5 \rightarrow -1$				
	$C\theta_5 \rightarrow 0$				
	$S\theta_8 \rightarrow 0$				
	$C\theta_8 \rightarrow 1$				
C_6-C_9	$x_6 \rightarrow -1$	6	14	6	0,7
	$x_9 \rightarrow 2$				
	$S\theta_6 \rightarrow 1$				
	$C\theta_6 \rightarrow 0$				
	$S\theta_9 \rightarrow 0$				
	$C\theta_9 \rightarrow 1$				
Full graph	$C\phi_1 \rightarrow 0$	20	65	20	10,2
	$S\phi_1 \rightarrow 1$				
	$S\phi_2 \rightarrow 0$				
	$C\phi_2 \rightarrow 1$				
	$S\phi_3 \rightarrow 1$				
	$C\phi 3 \rightarrow 0$				
	$S\phi_4 \rightarrow 0$				
	$C\phi_4 \rightarrow 1$				
	$C\theta_7 \rightarrow 0$				
	$S\theta_7 \rightarrow -1$				
Total time					15,7

6 Conclusions

The ability to produce answers to questions about the feasibility and solution structure of the GCS/SF problem is crucial in automated analysis and planning environments. Previous work by the authors has established an algebraic geometry approach to the problem. As would be expected, the cost of such determinism is the exponential computational effort required. This paper has presented graph-theoretic approaches to formulate and solve the problem using a Divide-and-Conquer approach in the hope of exploiting special structure that might exist in a particular problem. This method (i) identifies the degrees of freedom lost in local subproblems; (ii) detects local geometric or topological inconsistencies and (iii) reduces the size of the GCS/SF problem to the degrees of freedom left by the local instancing processes. The results in tables (6) and (7) evidence the reduced computational effort of these techniques when compared to the results produced by attempting to solve the entire problem at once.

We contemplate the use of such an approach to model and solve instances of GCS/SF problems that present strongly (non-trivially) constrained local sub-problems, in a multi-body multi-constraint problem. We conclude with the following remarks

1. In general, the Groebner Basis, produced by lexicographic or total degree ordering, lends itself very well to computation of the set of common roots of a polynomial set.

2. For larger systems, the Divide-and-Conquer techniques are advisable, since they take advantage of the existence of subsystems strongly constrained internally, and weakly related to the external world. These subsystems correspond to cycles in the Spatial Constraint graph which have instanced some of their degrees of freedom. A directly related situation in Assembly Planning corresponds to the existence of subassemblies within a large assembly. If Divide-and-Conquer techniques are used, the local Groebner Bases are used towards the solution of the general system. These GB_i sets are already ordered (lexicographically or by degree order) and free of redundancy and inconsistencies. Therefore there is a amount of work contributed by these bases towards the final solution.

3. Incremental-instancing presents the advantage of actually eliminating degrees of freedom from the variable set, therefore contributing to lower the computational expenses of the solution. The improvement by this technique acts during the most expensive part of the solution process (Full Graph processing). Therefore, it has the potential of significantly speeding up the computation.

4. Although the pertinent examples are not discussed because limitation in space, it has been found that preprocessing techniques speed up the solution of large-size problems, while for small-sized ones the brute-force approach is more advisable. This result can be attributed to the overhead in setting up the different subproblems, which cannot justified if the full-size problem is not large enough.

5. A partitioning of the GCS/SF problem is required to establish the complete, non redundant set of equations for the problem. If this is done with the Divide-and-Conquer technique in mind, then no additional computation effort is expended in producing a workable set of subproblems. Since the cost corresponding to the partition of the GCS/SF problem is present regardless of the utilization of Divide-and-Conquer techniques, their application simply takes advantage of direct computational costs.

Appendix A. Groebner Bases for cartesian table example

A.1 Brute-force approach

The complete set of group-based matrix equations modeling the constraint structure of the Cartesian table is

$$
\begin{aligned}
&F_{11}.C_1(\theta_1, y_1, z_1, \phi_1) = F_{21}.C_2(\theta_2, y_2, z_2, \phi_2)\\
&F_{12}.C_3(\theta_3, y_3, z_3, \phi_3) = F_{22}.C_4(\theta_4, y_4, z_4, \phi_4)\\
&F_{13}.C_6(\theta_6, x_6).F_{12}^{-1} = F_{33}.C_9(\theta_9, x_9).F_{22}^{-1}\\
&F_{23}.C_8(\theta_8, x_8).F_{21}^{-1} = F_{13}.C_5(\theta_5, x_5).F_{11}^{-1}\\
&C_5(\theta_5, x_5).C_1(\theta_1, y_1, z_1, \phi_1).C_7(\theta_7, y_7, z_7) =\\
&C_6(\theta_6, x_6).C_3(\theta_3, y_3, z_3, \phi_3).F_{15}^{-1}.F_{25}\,.
\end{aligned}
\tag{4}
$$

A lexicographically ordered Groebner Basis is calculated for this model, using the order $S\phi_1 \succ C\phi_1 \succ y_1 \succ z_1 \succ S\theta_1 \succ C\theta_1 \succ S\phi_2 \succ C\phi_2 \succ y_2 \succ z_2 \succ S\theta_2 \succ C\theta_2 \succ S\phi_3 \succ C\phi_3 \succ y_3 \succ z_3 \succ S\theta_3 \succ C\theta_3 \succ S\phi_4 \succ C\phi_4 \succ y_4 \succ z_4 \succ S\theta_4 \succ C\theta_4 \succ S\theta_5 \succ C\theta_5 \succ x_5 \succ S\theta_6 \succ C\theta_6 \succ x_6 \succ S\theta_7 \succ C\theta_7 \succ y_7 \succ z_7 \succ S\theta_8 \succ C\theta_8 \succ x_8 \succ S\theta_9 \succ C\theta_9 \succ x_9$.

The Groebner Basis is as follows

$$
\begin{aligned}
&S\phi_1 + S\theta_4.C\phi_4 = 0 \\
&C\phi_1 = 0 \\
&y_1 - S\theta_4.z_4 - 3 = 0 \\
&5.z_1 - 10.S\theta_4 + y_4.S\theta_4.y_7 + 5.S\theta_4.y_4 - 2.S\theta_4.y_7 - C\phi_4.z_7.S\theta_4.y_7 - \\
&5.C\phi_4.z_7.S\theta_4 = 0 \\
&5.S\theta_1 - C\phi_4.z_7 + y_4 - 2 = 0 \\
&C\theta_1 = 0 \\
&S\phi_2 = 0 \\
&C\phi_2 - S\theta_4.S\theta_7 = 0 \\
&y_2 + S\theta_4.y_7 + 5.S\theta_4 - 2 = 0 \\
&5.z_2 - S\theta_4.z_4.C\phi_4.z_7 - 2.C\phi_4.z_7 + S\theta_4.z_4.y_4 + 2.y_4 - 2.S\theta_4.z_4 - 4 = 0 \\
&S\theta_2 = 0 \\
&5.C\theta_2 + C\phi_4.z_7 - y_4 + 2 = 0 \\
&S\phi_3 + S\theta_4.C\phi_4 = 0 \\
&C\phi_3 = 0 \\
&y_3 - 1 - S\theta_4.z_4 = 0 \\
&z_3 - 2.S\theta_4 + S\theta_4.y_4 = 0 \\
&S\theta_3 = 0 \\
&C\theta_3 + S\theta_4 = 0 \\
&S\phi_4 = 0 \\
&C\phi_4^2 - 1 = 0 \\
&C\phi_4.y_4 - 2.C\phi_4 + 5.S\theta_7 - z_7 = 0 \\
&5.C\phi_4.S\theta_7 + y_4 - C\phi_4.z_7 - 2 = 0 \\
&C\phi_4.z_7^2 + 2.z_7 - 25.C\phi_4 - z_7.y_4 - 5.S\theta_7.y_4 + 10.S\theta_7 = 0
\end{aligned}
\tag{5}
$$

$$
\begin{aligned}
& y_4^2 - 4.y_4 - 21 + 10.S\theta_7.z_7 - z_7^2 = 0 \\
& S\theta_4^2 - 1 = 0 \\
& C\theta_4 = 0 \\
& S\theta_5 + 1 = 0 \\
& C\theta_5 = 0 \\
& x_5 - 1 = 0 \\
& S\theta_6 - 1 = 0 \\
& C\theta_6 = 0 \\
& x_6 + 1 = 0 \\
& S\theta_7^2 - 1 = 0 \\
& C\theta 7 = 0 \\
& S\theta_8 = 0 \\
& C\theta_8 - 1 = 0 \\
& x_8 - 2 = 0 \\
& S\theta_9 = 0 \\
& C\theta_9 - 1 = 0 \\
& x_9 - 2 = 0\,.
\end{aligned}
$$

Which presents y_7 and z_7 as free variables.

A.2 Divide-and-Conquer approach

Local Preprocessing. Cycle $C_1 - C_2$. For this cycle the constraint equation is

$$F_{11}.C_1(\theta_1, y_1, z_1, \phi_1) = F_{21}.C_2(\theta_2, y_2, z_2, \phi_2)\,. \tag{6}$$

Given the order $S\phi_1 \succ C\phi_1 \succ y_1 \succ z_1 \succ S\theta_1 \succ C\theta_1 \succ S\phi_2 \succ C\phi_2 \succ y_2 \succ z_2 \succ S\theta_2 \succ C\theta_2$ the lexicographic Groebner Basis resulted in

$$\begin{aligned}
&S\phi_1 - C\theta_2.C\phi_2 = 0\\
&C\phi_1 + C\theta_2.S\phi_2 = 0\\
&y_1 - 1 + C\theta_2.z_2 = 0\\
&z_1 + 2.C\theta_2 - C\theta_2.y_2 = 0\\
&S\theta_1 + C\theta_2 = 0\\
&C\theta_1 = 0\\
&S\phi_2^2 + C\phi_2^2 - 1 = 0\\
&S\theta_2 = 0\\
&C\theta_2^2 - 1 = 0\,.
\end{aligned} \tag{7}$$

Which presents y_2, z_2, $C\phi_2$ as free variables.

Local Preprocessing. Cycle $C_3 - C_4$. The constraint equation for this cycle is

$$F_{12}.C_3(\theta_3, y_3, z_3, \phi_3) = F_{22}.C_4(\theta_4, y_4, z_4, \phi_4)\,. \tag{8}$$

For the order $S\phi_3 \succ C\phi_3 \succ y_3 \succ z_3 \succ S\theta_3 \succ C\theta_3 \succ S\phi_4 \succ C\phi_4 \succ y_4 \succ z_4 \succ S\theta_4 \succ C\theta_4$, the following lexicographic Groebner Basis is calculated

$$\begin{aligned}
&S\phi_3 + S\theta_4.C\phi_4 = 0\\
&C\phi_3 - S\theta_4.S\phi_4 = 0\\
&y_3 - 1 - S\theta_4.z_4 = 0\\
&z_3 - 2.S\theta_4 + S\theta_4.y_4 = 0\\
&S\theta_3 = 0\\
&C\theta_3 + S\theta_4 = 0\\
&S\phi_4^2 + C\phi_4^2 - 1 = 0\\
&S\theta_4^2 - 1 = 0\\
&C\theta_4\,.
\end{aligned} \tag{9}$$

Which presents free variables z_4, y_4 and $C\phi_4$.

Local Preprocessing. Cycle $C_5 - C_8$. The constraint structure of this loop is as follows

$$F_{23}.C_8(\theta_8, x_8).F_{21}^{-1} = F_{13}.C_5(\theta_5, x_5).F_{11}^{-1}\,. \tag{10}$$

The ordering $x_5 \succ S\theta_5 \succ C\theta_5 \succ x_8 \succ S\theta_8 \succ C\theta_8$ leads to a (lexicographic) Groebner Basis

$$\begin{aligned} &x_5 - 1 = 0\\ &S\theta_5 + 1 = 0\\ &C\theta_5 = 0\\ &x_8 - 2 = 0\\ &S\theta_8 = 0\\ &C\theta_8 - 1 = 0\,. \end{aligned} \tag{11}$$

Which represents a zero-dimensional ideal.

Local Preprocessing. Cycle $C_6 - C_9$. The constraint matrix equation for this loop is

$$F_{13}.C_6(\theta_6, x_6).F_{12}^{-1} = F_{33}.C_9(\theta_9, x_9).F_{22}^{-1}\,. \tag{12}$$

The ordering $x_6 \succ S\theta_6 \succ C\theta_6 \succ x_9 \succ S\theta_9 \succ C\theta_9$ produces this (lexicographic) Groebner Basis, which represents a zero-dimensional ideal

$$\begin{aligned} &x_6 + 1 = 0\\ &S\theta_6 - 1 = 0\\ &C\theta_6 = 0\\ &x_9 - 2 = 0\\ &S\theta_9 = 0\\ &C\theta_9 - 1 = 0\,. \end{aligned} \tag{13}$$

Acknowledgment

This investigation was partially supported by the Manufacturing Research Center of the University of Illinois, and by the National Science Foundation PYI Award (DDM - 9157191).

References

[1] H. Asada and A. By. *Kinematics of workpart fixturing*, IEEE Conference on Robotics and Automation, 1985.

[2] O. Ruiz and P. Ferreira. *Algebraic geometry and group theory in geometric constraint satisfaction*, Proceedings, International Symposium on Symbolic and Algebraic Computation, University of Oxford, 224-233 (1994).

[3] A. Ambler and R. Popplestone. *Inferring the positions of bodies from specified spatial relationships*, Artificial Intelligence, **6**, 1975.

[4] E. Celaya and C. Torras. *Finding object configurations that satisfy spatial relationships*, Proceedings, European Conference on Artificial Intelligence, Stockholm, 141-146 (1990).

[5] D. Rocheleau and K. Lee. *System for interactive assembly modelling*, Computer Aided Design, **19**(2), 1987.

[6] Z. Fu and A. DePennington. *Geometric reasoning based on grammar parsing*, Journal of Mechanical Design, **116**(3), 763-769 (1994).

[7] G. Kramer. *Solving geometric constraint systems*, MIT Press, 1992.

[8] J. Turner et al. *Constraint representation and reduction in assembly modeling and analysis*, IEEE Journal of Robotics and Automation, **8**, 741-750 (1992).

[9] J. Angeles. *Rational kinematics*, Springer-Verlag, 1988.

[10] J. Herve. *Analyse structurelle des mechanisms par groupe des deplacements*, Mechanism and Machine Theory, **13**, 437-450 (1978).

[11] C. Hoffmann. *Geometric and solid modeling*, Morgan-Kaufmann Publishers Co., 1989.

[12] J. Angeles. *Spatial kinematic chains*, Springer-Verlag, 1982.

[13] R. Popplestone et al. *An interpreter for a language for describing assemblies*, Artificial Intelligence, **14**, 79-106 (1980).

[14] R. Popplestone. *Group theory and robotics*, Robotics Research, The First Intl. Symposium, MIT Press, M. Brady and R. Paul, editors, 1984.

[15] R. Popplestone et al. *A group theoretic approach to assembly planning*, Artificial Intelligence Magazine, 1990.

[16] F. Thomas and C. Torras. *Inferring feasible assemblies from spatial constraints*, Technical Report IC-DT-1989.03, Institute of Cybernetics, Univ. Polytecnic of Catalonia, 1989.

[17] F. Thomas. *Graphs of kinematic constraints*, Computer Aided Mechanical Assembly Planning, L. Homem de Mello and S. Lee, editors, Kluwer Academic Publishers, 81–109 (1991).

[18] E. Celaya and C. Torras. *Solving multiloop linkages with limited-range joints*, Mechanism and Machine Theory, **29**(3), 1994.

[19] O. Ruiz. *Geometric reasoning in computer aided design, manufacturing and process planning*, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1995.

[20] D. Kapur and Y. Lakshman. *Elimination methods: an introduction*, Symbolic and Numerical Computation for Artificial Intelligence, Academic Press, B. Donald, D. Kapur and J. Mundy, editors, 45-88 (1992).

[21] B. Buchberger. *Applications of Groebner basis in non-linear computational geometry*, Geometric Reasoning, MIT Press, D. Kapur and J. Mundy editors, 413–446 (1989).

[22] W. Ledermann. *Introduction to the theory of finite groups*, Oliver and Boyd, 1953.

[23] W. Ledermann. *Introduction to group theory*, Barnes and Noble, 1973.

[24] J. R. Johnson and D. Johnson. *Graph theory with engineering applications*, New York: Ronald Press Company, 1972.

[25] N. Deo. *Graph theory with applications to engineering and computer science*, Englewood Cliffs, N.J.: Prentice-Hall Inc., 1974.

[26] M. Swamy and K. Thulasiraman. *Graphs, networks and algorithms*, John Wiley & Sons, 1974.

[27] T. Becker. *Groebner bases: a computational approach to commutative algebra*, New York: Springer-Verlag, 1993.