Computational Algorithm for Determining the Generic Mobility of Floating Planar and Spherical Linkages

Offer Shai
Mechanical Engineering School,
Tel-Aviv University
Tel-Aviv, Israel

Andreas Müller
Institute of Mechatronics, Chemnitz, Germany,
andreas.mueller@ifm.chemnitz.de

ABSTRACT

It is well-known that structural mobility criteria, such as the Chebychev-Kutzbach-Grübler (CKG) formula, fail to correctly determine the mobility of mechanisms with special geometries. Even more, any known structural mobility criteria also fail to determine the generic (i.e. topological) mobility since they are prone to topological redundancies

A computational algorithm is proposed in this paper, which always finds the correct generic mobility of any planar and spherical mechanism. Its foundation is a novel representation of constraints by means of a constraint graph. The algorithm builds on the 'pebble game', originally developed within combinatorial rigidity theory for checking the rigidity of graphs. An extension of Laman's theorem is introduced that enables application of the algorithm to any planar or spherical mechanism with higher and lower holonomic kinematic pairs and multiple joints.

The novel algorithm further yields the redundantly constrained sub-linkages of a mechanism. In addition this algorithm naturally leads to a decomposition of a mechanism into Assur graphs, however this is beyond the scope of this paper.

Keywords: Mobility, topological redundancy, pebble game, Assur graphs

1. Introduction

The mobility being the essential property of a mechanism has been a major matter of interest in mechanism theory. The approaches can be broadly classified as those that deal with the mobility of a given mechanism, with a particular geometry, and those that aim on the generic mobility of a class of mechanisms with certain topology [1]. Methods of the first class attempt an explicit solution of the constraint equations or the approximation of the solution variety [13, 14, 15], possibly using tools from numerical algebraic geometry [16, 17]. Instead of considering a particular geometry, the second class approaches the problem from a structural point of view. These attempts have a long tradition and only need topological information about the existence of links and joints. The CKG formula is a well-known topological mobility criterion. It is assumed that they generally yield the generic mobility [6], i.e. the mobility of almost all realizations of a particular topology. Although they are independent of any geometric data all such methods are sensitive to topological redundancy since these criteria only take into account the existence of joints and links but not their particular arrangement.

The identification of topological redundancies requires graph-theoretic considerations of the constraints and appropriate algorithms. Such an algorithm is presented in this paper. The basis for this algorithm is a graph representation of the constraints inherited from rigidity theory. This differs from the topological graph [3] often used in that it does not merely represent the arrangement of links and joints, but rather the system of constraints imposed to the links. This is presented in section 2, where the two established types (body-bar, bar-joint) are recalled and are mentioned briefly in the paper, and a novel type of constraint graph is introduced. The mathematical theorem underlying the proposed computational algorithm is given in section 3, and the actual computational algorithm is introduced in section 4. The algorithm is proved to converge to the unique generic mobility [9]. In order to motivate the application of this algorithm, an engineering interpretation of the steps and output of the algorithm is given. The application of the method is shown in section 5 for a simple example, and further interpretations of the output are discussed. The paper concludes with a brief outline of future work in section 6.

The algorithm used in this paper, called pebble game, was developed in 1997 [2] for checking whether a set of points subject to geometric constraints form a rigid structure. The use of this algorithm was also extended to check whether a graph consisting of rigid bodies is rigid or mobile as reported in [10]. In engineering, pebble game was applied to check the mobility of planar mechanisms consisting of only binary links and limited to lower kinematic pairs [8]. It was proved that pebble game can decompose any mechanism with only binary links to Assur graphs in 2d and 3d [7]. The algorithm reported in this paper overcomes this limitation and is applicable to any type of planar mechanisms with holonomic higher and lower kinematic pairs and multiple joints.

2. Constraint Graphs

The kinematic functionality of a mechanism is dictated by the geometric and topological constraints imposed on its bodies. The topological graph already relates bodies and joints but it does not explicitly represent the imposed constraints. To this end a *constraint graph* G is introduced. In the following δ denotes the generic mobility, $G(E,V)$ the constraint graph (undirected or directed), $e(G)=|E|$ the number, and $v(G)=|V|$ the number of vertices of G.

The idea behind constraint graphs is to represent a mechanism as an abstract relation of ‘objects’ representing certain degrees of freedom (DOFs). These objects constitute vertices of the constraint graph, and are chosen so as to uniquely represent the mechanism’s configuration. They can stand for rigid bodies or points. The constraints between them are represented by edges. In this sense the graph represents a system of abstract constraint relations that possibly have different physical meanings (e.g. rotation or translation constraints).

There are several types of constraint graphs, such as Bar-Joint graph and Body-Bar graph, but the most general constraint graph, developed by the authors, applies to any type of planar mechanism is the mixed constraint graph below.

In this paper we introduce a new type of graph, termed mixed constraint graph $G = (V_B \cup V_J, E)$. In this graph a vertex v can represent a rigid body, $v \in V_B$, as well as points, $v \in V_J$. That is, for a planar mechanism, each vertex of the mixed constraint graph embodies an object that can move in the plane, and its physical meaning follows from that of the body-bar and bar joint-graph. If a vertex represents a body then it possesses three DOFs. If it represents a point (i.e. the location of a joint) then it has two DOFs. Note, this type of constraint graph can also deal with multiple joints, a revolute joint connecting m bodies thus stands for $m - 1$ revolute joints with collinear axes. For example, in Figure 1.a, joint J_1 is a multiple revolute joint while the other two joints, J_2 and J_3, are binary joints, i.e., connect between two bodies/links.

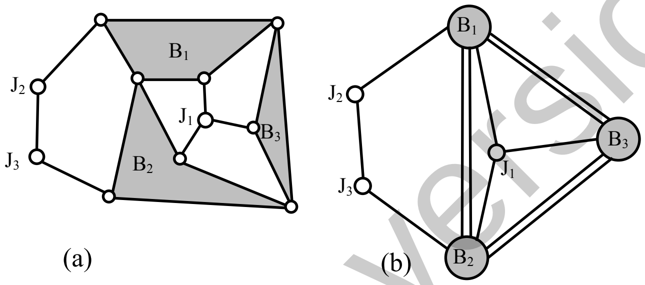

Figure 1. A linkage (a) whose mixed constraint graph (b) does not satisfy the mixed Laman theorem.

3. Rigidity and Mobility of Mixed Graphs

One of the main problems in checking the correct generic mobility of a mechanical system is to identify whether there is no sub-system having over-determinacy, redundant elements. A mathematical criterion for checking such non-existence of over-determinacy was established and proved in 1970 [4] for bar-joint graph, while in 1991 a mathematical criterion for body-bar graphs was reported [12]. These theorems give rise to the following theorem for mixed constraint graphs:

Planar Mixed Laman theorem (Shai and Müller, 2013): A floating planar mixed constraint graph $G = (V_B \cup V_J, E)$ with $e(G) = 3v_B(G) + 2v_J(G) - 3$ is determined if and only if $e(G') \leq 3v_B(G') + 2v_J(G') - 3$ for every subgraph G' of G, where $v_B(G)$ and $v_J(G) = |V_J|$ is the number of vertices corresponding to bodies and points/joints, respectively.

Corollary: A floating planar mixed constraint graph $G = (V_B \cup V_J, E)$ is non-redundant if and only if $e(G') \leq 3v_B(G') + 2v_J(G') - 3$ for every subgraph G'. If this condition is satisfied, the linkage has generic mobility $\delta(G) = 3v_B(G) + 2v_J(G) - e(G) \geq 3$.

For example, the floating system in Figure 1.a is not determined since the corresponding mixed graph in Figure 1.b does not satisfy the Mixed Laman theorem. To prove that, let us choose the sub-graph spanned by the vertices: $\mathrm{V}' = \{\mathrm{B}_1, \mathrm{B}_2, \mathrm{B}_3, \mathrm{J}_1\}$ having 9 edges which is greater than $3 \cdot 3 + 2 \cdot 1 - 3 = 8$, thus mixed Laman's theorem is not satisfied.

4. Pebble Game - A Computational Algorithm

Pebble game is a very efficient algorithm to check if a graph satisfies the mixed Laman theorem and thus to check if there exists an overdetermined sub-graph. The pebble game is of polynomial order in the number of vertices, $O(|V|)^2$ and the required memory also grows quadratically, i.e. with $O(|V|)^2$ [5].

The main concept of the algorithm is to assign 'pebbles' to any physical object in the kinematic model (bodies, points) representing certain DOFs, and to remove them in course of the algorithm. The number of pebbles remaining after running the pebble game is equal to the generic mobility of the linkage. Aiming on the generic, i.e. topological, mobility the method operates exclusively upon the constraint graph, i.e. the topology, and a generic rather than a specific geometry is assumed. Redundancies due to special geometries are thus excluded.

The pebble game starts with an unconstrained system, in the sense that the number of pebbles assigned to a vertex is equal to the DOF as if its members were not subject to any constraint. Denote with $k(v)$ the DOF of the object represented by vertex v. For planar constraints graphs $k(v) = 2$ represents a point and $k(v) = 3$ a body. The algorithm is initialized by assigning $k(v) = 2{,}3$ pebbles to each vertex v. That is, initially there are no constraints between the elements of a linkage, i.e. each element has $k(v)$ DOFs to move in the plane.

Each edge of G represents one constraint. Initially all constraints are inactive, i.e. all objects/vertices are unconstrained. An inactive constraint is represented by an undirected edge (constraint graph G is initially undirected). During the pebble game the constraints are successively activated by directing the edges. This indicates that the DOFs of one vertex are depending on the DOFs of other vertices. In the algorithm this is achieved by removing a pebble from one of its end-vertices. An undirected edge is termed *admissible* if the total of free pebbles next to its end vertices is at least four. Only admissible edges can be directed and can thus become active constraints.

Input to the Pebble Game algorithm:

The algorithm starts from the topological graph, i.e. an undirected graph as described in section 2. Each vertex v represents a physical object that has $k(v)$ DOF.

The Pebble Game algorithm:

1. INITIALIZATION: Assign $k(v)$ pebbles to each vertex v of the undirected graph, thus all edges are admissible. This is equivalent to regarding all mechanical objects, corresponding to the vertices, as unconstrained, i.e. each having $k(v)$ DOFs.

2. WHILE there exist admissible edges DO the following **Orientation Move (Vertex - Edge move)**:

   Let (u, v) be an admissible edge, i.e., the total sum of pebbles next to the two end vertices is at least 4. Remove one pebble from one of its end vertices, let it be vertex u, and replace the edge by a directed edge $\langle u, v\rangle$, i.e., u becomes the tail and v the head vertex of $\langle u, v\rangle$.

   END WHILE

   After this loop there are no admissible edges left. This move corresponds to activating the constraint corresponding to the pebble removed from the tail vertex. The direction of the edge introduces a causality in the sense that one DOF of the tail vertex u is assumed to be dependent on one DOF of the head vertex v. Note that this is an abstract assignment, i.e. it is not said that a certain DOF of u is made dependent on a certain DOF of v.
3. WHILE there are free pebbles left DO the following **Reorientation move (Vertex - Vertex Move):**

   Choose an undirected edge (u, v) and make it admissible by bringing free pebbles to its end vertices by applying the following steps ($\mathrm{peb}(v)$ denotes the number of pebbles at vertex v):

   Suppose $\mathrm{peb}(v) < 2$, if v stands for a point, or $\mathrm{peb}(v) < 3$, if v stands for a body. Then search for a vertex, say z, with free pebbles, i.e., $\mathrm{peb}(z) > 0$, for which there is a non-directed path from v to z. Then redirect all edges within the path from v to z so to form a directed path, and move one pebble from vertex z to v. Finally set $\mathrm{peb}(z) \coloneqq \mathrm{peb}(z) - 1$ and $\mathrm{peb}(v) \coloneqq \mathrm{peb}(v) + 1$.

   END WHILE

5. Example of Applying Mixed Pebble Game

In Figure 2 we apply the mixed pebble game to the mixed graph representing the linkage in Figure 1.a.

Initially, all the bodies and joints have three and two pebbles, respectively, as shown in Figure 2.a. The orientation move is first applied and all the admissible edges are directed. For example, the two edges (B_1, B_3) and (B_1, J_1) are admissible, thus can be oriented, since there are 6 and 5 pebbles respectively next to the two end vertices. Figure 2.b shows all edges that could be directed by applying the orientation move. Since there are no more admissible edges the reorientation move is being applied next. For example, in Figure 2.c edge (J_2,B_1) becomes admissible by moving one pebble from vertex B_3 and one from B_2 thus it can be oriented as shown in Figure 2.d.

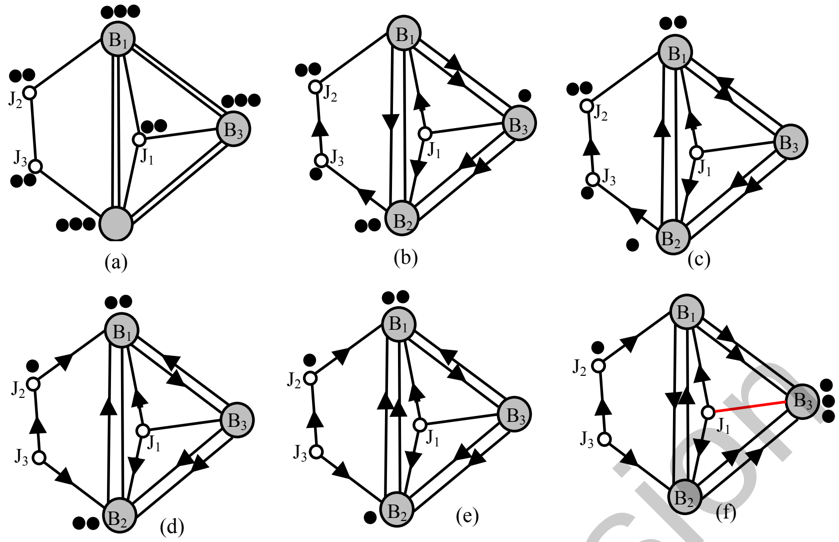

Figure 2. Example of applying mixed Pebble game on mixed constraint graph.

Applying reorientation move on edge $\langle B_2, J_3 \rangle$ brings a free pebble to vertex B_2 thus edge $\langle B_2, B_1 \rangle$ is now directed as shown in Figure 2.e.

Now we are left with four free pebbles and one edge, (J_1, B_3) unoriented. It is possible to move 3 pebbles next to any two end vertices, thus we move them to the end vertices of edge (J_1, B_3). For the sake of consistency, we move them to vertex B_3 as shown in Figure 2.f.

In Figure 2.f there are no edges that can be made admissible by applying the reorientation move and the algorithm terminates. The output of the algorithm allows for the following interpretations:

Result 1: The most obvious result is the generic mobility of the associated linkage. Since the algorithm terminates with 4 free pebbles the planar linkage generically possesses 4 DOFs.

Result 2: Besides the generic mobility the particular location of the pebbles indicates which links can be moved independently, hence can be used as control inputs. As we deal with floating planar linkages there are always 3 DOFs that correspond to the relocation of the linkage as a whole. In this example there are 4 free pebbles. Each of the pebbles represents one DOF that can be independently controlled. The specific allocation of pebbles in figure 2.f, together with the original mechanism in figure 1.a, allows for an apparent interpretation: the 3 DOFs of B_3 describe the location and orientation of the linkage in the plane, and the one pebble at J_2 is a translation DOF of the location point of J_2 that controls the internal shape.

Notice that

1. The pebble at J_2 is not the joint angle but one component of the location vector.
2. There is no specific assignment of coordinates to the DOFs so that ANY generalized coordinate can be used to represent the DOF of J_2. The pebble game algorithm operates on an abstract level and does not need specific selection of coordinates.
3. The particular allocation of pebbles is not unique and can be controlled in course of the algorithm. Also the algorithm's result can be changed by application of the reorientation move (which does not change the number of free pebbles). For instance, in figure 2.f a free pebble is now assigned to vertex J_2. With a reorientation of $\langle J_2, J_3 \rangle$, this pebble can be moved to J_3. Now the one independent DOF is assigned to J_3.

6. CONCLUSIONS AND OUTLOOK

The paper introduces an efficient computational algorithm for determining the correct generic/topological mobility for any planar or spherical mechanism with higher and lower kinematic pairs, including multiple joints. The paper introduces a mixed constraint graph, which is a more general constraint graph than other graphs introduced in the literature, such as body-bar and bar-joint graphs. One of the salient conclusions of this paper is that, by using the mixed constraint graph, it is possible to represent any planar mechanism, and consequently to invoke the corresponding mixed pebble game algorithm. The latter is the main contribution of the paper: it determines the correct generic mobility of the mechanism modeled by a mixed constraint graph. The planar mixed Laman theorem, which is an extension of the well-known Laman theorem for bar-joint graphs, is given as a mathematical foundation of the algorithm. As mentioned in the paper, the novel mixed pebble game always converges to the correct generic mobility. Moreover it is discussed that this computational algorithm allows for decomposing any mechanism into its building blocks, namely Assur graphs.

The reported algorithm applies to floating linkages, i.e. linkages that are not fixed to a ground. In a forthcoming publication, the mixed pebble game will be amended to include mechanisms (grounded mixed constraint graphs), which requires another type of constraint graphs. To this end the algorithm needs to be qualified so as to be able to treat immobile ground vertices.

REFERENCES

[1] G. Gogu: Mobility of mechanism: a critical review, Mech. Mach. Theory 40 (2005) 1068–1097.

[2] D. J. Jacobs, B. Hendrickson: An algorithm for two-dimensional rigidity percolation: the pebble game. Journal of Computational Physics, 137 (1997) 346 – 365

[3] A. Jain: Graph theoretic foundations of multibody dynamics, Part I: structural properties, Multibody. Syst. Dyn. 26 (2011) 307–333.

[4] G. Laman: On graphs and rigidity of plane skeletal structures. J. Engineering Mathematics, 4 (1970) 331–340.

[5] A. Lee, I. Streinu: Pebble game algorithms and sparse graphs, Discrete Mathematics, 308: 8 (2008) 1425-1437.

[6] A. Müller: Generic Mobility of Rigid Body Mechanisms, Mech. Machine Theory, Vol. 44, no. 6, (2009) 1240-1255.

[7] O. Shai: Rigidity of 2d and 3d Pinned Frameworks and the Pebble Game, London. Mathematical Society Workshop: Rigidity of Frameworks and Applications, July 12-15, Lancaster University, England, 2010.

[8] O. Shai, A. Sljoka, W. Whiteley: Directed Graphs, Decompositions, and Spatial Rigidity, Discrete Applied Mathematics, accepted 2012.

[9] O. Shai, A. Müller: A novel combinatorial algorithm for determining the generic/topological mobility of planar and spherical mechanisms", ASME Design Engineering Technical Conferences, August 12-15, 2013, Portland, OR.

[10] A. Sljoka: Counting for Rigidity, Flexibility and extensions via the Pebble Game Algorithm, Master's Thesis, York University 2006.

[11] A. Sljoka, O Shai, W. Whiteley: Checking mobility and decomposition of linkages via Pebble Game Algorithm, ASME Design Engineering Technical Conferences, August 28-31, (2011), Washington, USA.

[12] T.S. Tay: Henneberg's method for bar and body frameworks, Structural Topology, vol. 17, (1991) 53-58.

[13] K.J. Waldron: The constraint analysis of mechanisms, J. Mech. 1 (1966) 101–114.

[14] K.J. Waldron: A study of overconstrained linkage geometry by solution of closure equations – Part I. Method of study, Mech. Mach. Theory 8 (1973) 95–104.

[15] K.J. Waldron: A study of overconstrained linkage geometry by solution of closure equations – part II. Four-bar linkages with lower pairs other than screw joints, Mech. Mach. Theory 8 (1973) 233–24.

[16] C. Wampler, B.T. Larson, A.G. Erdman: A new mobility formula for spatial mechanisms, in: Proc. ASME International Design Engineering Technical Conferences (IDETC), September 4–7, (2007), Las Vegas.

[17] C.W. Wampler, J.D. Hauenstein, A.J. Sommese: Mechanism mobility and a local dimension test, Mech. Mach. Theory 46 (2011) 1193–1206.