Meccanica (2017) 52:441–455
DOI 10.1007/s11012-016-0403-5

Graph-based structural analysis of planar mechanisms

Sebastián Durango · **Jorge Correa** ·
Oscar E. Ruiz

Received: 17 April 2014 / Accepted: 23 February 2016 / Published online: 3 March 2016

Abstract Kinematic structure of planar mechanisms addresses the study of attributes determined exclusively by the joining pattern among the links forming a mechanism. The system group classification is central to the kinematic structure and consists of determining a sequence of kinematically and statically independent-simple chains which represent a modular basis for the kinematics and force analysis of the mechanism. This article presents a novel graph-based algorithm for structural analysis of planar mechanisms with closed-loop kinematic structure which determines a sequence of modules (Assur groups) representing the topology of the mechanism. The computational complexity analysis and proof of correctness of the implemented algorithm are provided. A case study is presented to illustrate the results of the devised method.

Keywords Graph · Kinematic structure · Assur group · Structural analysis · System group classification

S. Durango (✉)
Grupo Diseño Mecánico y Desarrollo Industrial, Universidad Autónoma de Manizales, antigua estación del ferrocarril, Manizales, Colombia
e-mail: sebastiandi@autonoma.edu.co

J. Correa · O. E. Ruiz
Laboratorio de CAD CAM CAE, Universidad EAFIT, Medellín, Colombia

1 Introduction

Modular kinematics and force analysis of mechanisms are reported to be computationally efficient and adaptable since they allow to define in advance a library of modules that are combined and reused to model wide families of mechanisms [10, 17]. Although modular analysis has been present in the literature for several decades, interest on these kinematics [7–9, 12–14, 25, 26] and dynamic analysis methods [6, 10, 23, 24] remains. Recently, modular kinematic analyses have been extended to the calculation of Jacobian matrices and quality indexes [4, 5]. Similar to the analysis, synthesis of mechanisms which addresses topological variations often implements modular approaches [8, 16, 17] that allow to rebuild immediately the kinematic structure of the model, therefore, ensuring the continuity of the design process [18, 24]. In this sense, modules (e.g. Assur groups) are building blocks or structural genes of complex systems (e.g. mechanisms) [19]. Modular analysis and synthesis are not limited to the planar case, e.g. references [18, 22] extend the modules to the spatial case, specifically for the synthesis of spatial mechanisms with a prescribed degree of freedom.

The *system group classification* is central to structural analysis and synthesis methods [17, 18, 20, 21, 24]. It consists of the determination of the modules that form a planar mechanism and the sequence in which those modules are connected. The categories of modules considered for this work correspond to the classical concept [2, 7, 20]: any mechanism can be disaggregated into independent chains: a set of kinematic chains, called primary elements group, whose number of Degrees Of Freedom (DOF) equals the DOF of the whole mechanism, and another set of kinematic chains whose DOF is null, known as Assur groups. In this way, any mechanism can be considered as formed by a primary (driving) elements group and one or several Assur groups. In the following, only planar groups with 1 DOF pairs will be considered.

Definition 1 *Driving-elements group* is formed by the fixed and the input links, and it represents the generalized coordinates of the system.

Definition 2 *Assur group*, which is defined by [1, 2, 4, 7] as a planar kinematic chain composed of $n = 2k$ links and $j = 3k$ joints, satisfying these axioms:

(a) each joint has mobility equal to one,
(b) there are r external joints that connect the group to an external kinematic chain with mobility M,
(c) the mobility of the expanded kinematic chain (external kinematic chain plus Assur group) remains M, and
(d) it is not possible to disaggregate the kinematic chain into simpler chains so that each chain fulfills the same requirements.

The kinematic structure is determined by the topology of the mechanism and its input-motion scheme. Determination of the system group classification of a mechanism lies in the kinematic (topological), and in particular, in the structural analysis domain. Traditionally, the system group classification has been determined by inspection. However, a computer algorithmic approach is required, specially when complex mechanisms are analyzed. References [3, 20, 21] present graph-based algorithms. References [3, 20] use kinematic principles to map mechanisms to graphs, while [21] is based on rigidity theory.

This article presents a graph-based kinematic formulation of the system group classification (Assur groups disaggregation) of planar mechanisms and its computational implementation. Methodology is presented in Sect. 3, developing: a mechanism-to-graph mapping (Sect. 3.1), structural analysis of planar mechanisms (Sect. 3.2), goal of disaggregation of planar mechanisms (Sect. 3.3), a graph-based algorithm for the disaggregation of planar mechanisms into Assur groups (Sect. 3.4), and a proof of correctness of the algorithm (Sect. 3.5). Section 4 presents the results through a test example (Sect. 4.1) and describes the implementation of the algorithm (Sect. 4.2). Finally, conclusions and future work are presented (Sect. 5).

2 Literature review

Kinematic-structure analysis of mechanisms includes mobility analysis, structural synthesis, isomorphism detection, structural analysis, and application to the creative design of products and systems [15].

Mruthyunjaya [15] presents a comprehensive study of research on kinematic structure of mechanisms, including analysis and synthesis. Structural analysis deals with the enumeration of diverse mechanisms derived from a kinematic chain and the characterization of their freedom type depending on the input and output motion scheme.

A category of structural analysis uses the classical concept of kinematic structure developed by Leonid Assur in 1914. An Assur group is a minimal kinematic chain with zero mobility from which it is not possible to obtain simpler kinematic chains with the same mobility each (Sect. 1, Definition 2). Usually, a mechanism can be designed as the successive joining of a driving-elements group (formed by the fixed and input links) and several Assur groups. The Assur groups that form a mechanism and their sequencing determine the kinematic structure of the mechanism. Assur groups are statically determinate. Thus, they represent a modular basis for the kinematics and force analysis, and for the synthesis of mechanisms [23].

Although research on kinematic structure of mechanisms remains of interest, there is little literature available on strategies (algorithms) for the system group classification of mechanisms based on Assur-group disaggregation.

Buśkiewics [3] presents a graph-based combinatorial algorithm, in which the Assur groups forming a planar mechanism are systematically identified. The algorithm is based on the fact that the combinatorial set of the internal loops of the graph representing a mechanism includes the sequence of Assur groups that form the mechanism. The algorithm is computationally efficient since practical mechanisms usually conduct to a small combinatorial set of internal loops. However, their algorithm works under the condition that the set of independent loops of the representing graph is given, which implies a combinatorial computation itself. The complexity of the computation could be reduced to polynomial time by using algorithms to find minimum spanning trees (e.g. Kruskal).

Saura et al. [20] present an algorithm for the structural analysis of planar mechanisms which includes lower and higher kinematic pairs. The algorithm is based on the simple structural group concept by Kolovsky et al. [11]. Structural groups are kinematic chains whose number of independent inputs equals its mobility. If a structural group cannot be dissagregated into several structural groups, then it is called simple. Assur groups are instances of simple structural groups which have zero mobility. The algorithm is inspired by a graph-analytic disaggregation procedure which advances from the fixed link to the distal simple structural group in the structure. Each step requires a combinatorial search of candidate chains which are evaluated to fulfill the simple structural group condition. The paper does not present a formal proof of correctness nor a computational complexity analysis.

Shai et al. [21] study Assur groups as graphs having special properties from rigidity theory. The article establishes the duality between planar mechanisms and unstable isostatic frameworks. The authors devise rigidity mathematical tools and apply them on kinematics and dynamics of mechanisms. They also develop a combinatorial algorithm to disaggregate a mechanism into Assur groups. The authors propose an unusual mechanism-to-graph mapping, which represents kinematic pairs and links by vertices and edges, respectively. This mapping allows direct application of rigidity theory on the graph-based mechanism analysis. However, the mapping only considers binary links. Multi-joint links must be represented as a set of connected binary links, even though no relative motion is developed.

Our work presents an alternative to the determination of the system group classification of mechanisms. A novel graph-based algorithm is designed in which mechanisms are represented by graphs in a conventional manner, resulting in the direct mapping of the kinematic-structural properties to the graph representation. For this purpose, kinematic properties of planar mechanisms and Assur groups are rigorously defined in both domains: mechanisms and graphs. The algorithm is based on the structural definition of Assur groups. The disaggregation procedure advances from the distal group to the fixed link in the structure. Different from [20], assessment of the structural characteristics of the disaggregated (candidate) chain and the remaining chain is done in each step. A proof of correctness and computational complexity analysis are also provided. The algorithm is tailored for kinematic chains formed exclusively by lower pairs. However, in the case of mechanisms which include higher pairs, it is possible to find an equivalent formed by lower pairs exclusively.

3 Methodology

3.1 Mechanism-to-graph mapping

This section describes two representations (structural and graph-based) of the kinematic structure of planar mechanisms. For the sake of simplicity, the following assumptions are made for both methods of representation:

(a) Redundant constraints will be ignored.
(b) All joints are assumed to be binary. Multiple joints will be replaced by a set of equivalent binary joints.
(c) Two mechanical elements with no relative motion between them will be represented as one link.
(d) Each joint will be assumed to have one DOF (lower pair). If a mechanism includes higher pairs, then it is possible to find a kinematically equivalent mechanism formed exclusively by lower pairs as described in “Appendix 1”.

3.1.1 Structural representation of mechanisms

In a structural representation, each link of a mechanism will be illustrated by a polygon whose vertices represent the kinematic pairs. In this manner, a *binary* link is represented by a line with two end vertices standing for their kinematic pairs, a *ternary* link is represented by a triangle with three vertices, a *quaternary* link is represented by a quadrilateral with four vertices (Fig. 1), and so on. Links will be numbered, with fixed link receiving the number 0. Figure 2 shows the structural representation of a four-bar mechanism.

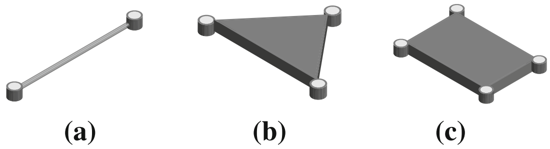


Fig. 1 Structural representation of links. **a** Binary link, **b** ternary link, **c** quaternary link

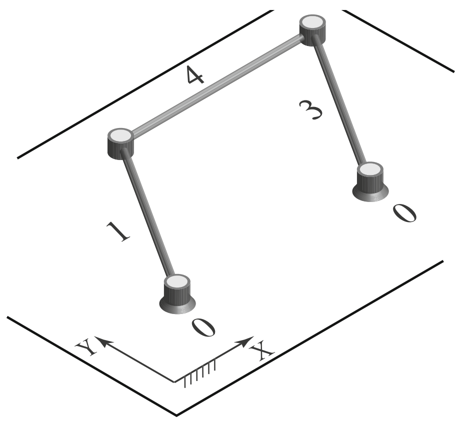


Fig. 2 Structural representation of a four-bar mechanism

3.1.2 Mapping function

Graph representation of mechanisms usually simplifies the structural analysis through graph theory, enabling systematic computer applications. For this purpose, we define the mapping function g in (1) such that it takes a kinematic chain K and returns a graph G in the following manner:

$$\begin{aligned} &g : K \longrightarrow G, \\ &g(N, J) = (V, E) = (V(N), E(J)), \end{aligned} \tag{1}$$

where N is the set of links of the mechanism, J is the set of joints of the mechanism, V is the set of vertices of the graph, and E is the set of edges of the graph.

In graph representation, the vertices of the graph denote the links of the mechanism and the edges denote the joints. Particularly, a joint connecting links u and v is represented by the edge $e = (u, v)$. In such a case, u and v are called the end points of e. The correspondence between the graph vertices and their respective links is straight forward.

3.1.3 Graph representation of mechanisms

The edge connecting two vertices in graph representation corresponds to the pair that connects two links. The degree of a vertex (number of edges incident with that vertex) represents the classification of its corresponding link. In this manner, we call a vertex of degree two a *binary* vertex (representing a *binary* link), a vertex of degree three a *ternary* vertex, and so on. We also call the graph representing a mechanism $G = (V, E)$. The joints connecting the driving links with the fixed link or other driving-elements are represented by thicker edges. Figure 3 shows a mechanism (Fig. 3a) and its graph representation (Fig. 3b).

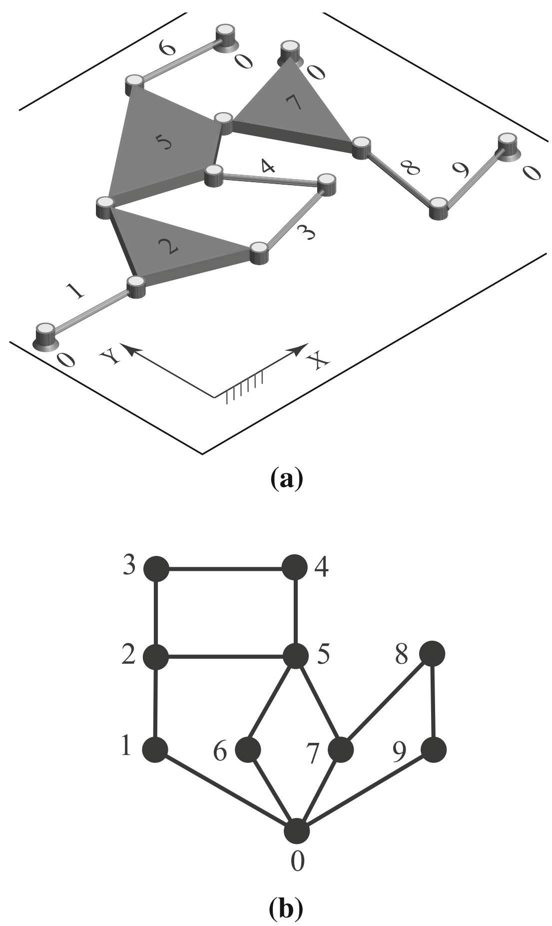


Fig. 3 Mechanism consisting of 9 links and 13 joints together with its graph representation. **a** Mechanism. **b** Graph representation of (**a**)

3.2 Structural analysis of planar mechanism

3.2.1 Domain and range of mapping

The herein developed structural analysis concerns those mechanisms that have a closed-loop kinematic structure and their input links connected either to the fixed or to another input link. According to this assumption, a graph obtained after mapping such type of mechanism, by the use of (1), presents the following characteristics:

(a) It is a connected graph. This means that every vertex in the graph is connected to every other vertex by at least one path.
(b) It is a block. This is, the graph is connected and has no cut points or bridges. A cut point is a vertex whose removal results in an increase in the number of components. In addition, a bridge is an edge whose removal results in an increase in the number of components. Bridges and edges often arise when representing a mechanism with excessive constraints in graph form.
(c) It has no parallel edges or slings as only binary joints are accounted for and two links with no relative motion between them are represented as one link.

3.2.2 Degrees of freedom and independent loops of a mechanism

The first concern in the study of a mechanism's kinematic structure is the DOF. For a graph $G = (E, V)$ which represents a mechanism M that meets conditions stated in Sect. 3.1.2, the DOF W of a mechanism can be written as (2):

$$W = 3(|V| - 1) - 2|E|, \tag{2}$$

where W is the DOF of the mechanism, $|V|$ is the number of vertices, equivalent to the number of links of the mechanism, and $|E|$ is the number of edges, equivalent to the kinematic pairs of the mechanism.

The term $(|V| - 1)$ results from taking vertex zero out the DOF count. In addition, it is possible to determine the number of independent loops L in the graph of a mechanism by considering the equation of Euler (3):

$$L = |E| - |V| + 1. \tag{3}$$

3.2.3 Structure analysis and Assur groups disaggregation of planar mechanisms

Several methods focused on the analysis of planar mechanisms rely on the study of their kinematic structure, particularly, on their division into simple parts or groups of elements called *driving-elements group* and *Assur groups*.

The DOF of a mechanism can be written as:

$$W = W + 0 + 0 + \cdots + 0. \tag{4}$$

According to (4), a mechanism can be divided into separate parts or kinematic chains. The simple kinematic chain, whose DOF W is equal to the DOF of the whole mechanism is denoted as the driving-elements group. Those kinematic chains whose DOF is equal to zero and which cannot be disaggregated into simpler kinematic chains with the same property are denoted as Assur groups, see Sect. 1, Definitions 1 and 2. In this sense, a planar mechanism can be considered as consisting of a driving-elements group and a number, one or more, of Assur groups. We call a graph representing an Assur group $G_A = (V_A, E_A)$, where $G_A \subset G$. The Assur groups forming a mechanism and the order in which those groups are connected between them determine a system group classification.

3.2.4 Driving-elements group

The driving-elements group consists of one fixed element and one or more free elements called *primary elements*. The primary elements are connected either with the fixed element or with another primary element and the DOF W of the mechanism is equal to its number of primary elements. The graph that represents a driving-elements group is called $G_D = (V_D, E_D)$, where $G_D \subset G$, and its characteristic is that every vertex is connected with vertex zero by one path. Table 1 shows different types of driving-elements groups consisting of one and two primary elements together with their graph representation.

3.2.5 Assur groups

The graph representing an Assur group $G_A = (V_A, E_A)$ is a connected graph and its DOF can be determined based on (2). For this purpose, it is necessary to consider that every element of the group is movable as

Table 1 Primary-elements groups and their graph representation

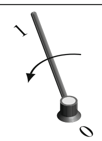
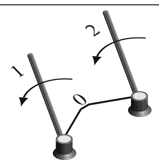
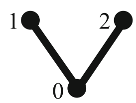
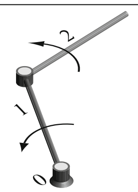
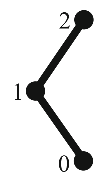
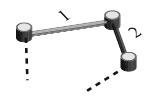
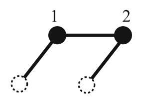
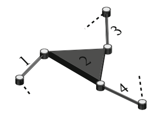
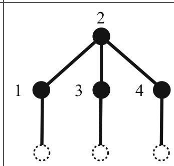
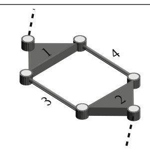
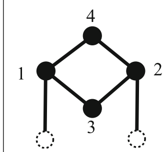
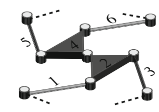
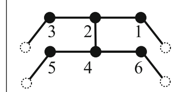
Primary-elements number	Structural representation	Graph representation
1		
2		
2		

Table 2 Different types of Assur groups

k	e	v	r	L	L_c	Structural representation	Graph representation
1	3	2	2	2	0		
2	6	4	3	3	0		
2	6	4	2	3	1		
3	9	6	4	4	0		

a result of incorporation of the fixed element into the driving-elements group. The relation between the edges and vertices of a graph representing an Assur group is given by (5) since the DOF of the Assur group is zero with respect to the links that join to this group:

$$3|V_A| - 2|E_A| = 0. \tag{5}$$

Both numbers $|V_A|$ and $|E_A|$ must be natural, thus, the product $2|E_A|$ will always be an even number. As a result, the product $3|V_A|$ must also be an even number and, $|V_A|$ must be a multiple of 2 that can be represented as $|V_A| = 2k$. We call k the group class. Substituting the value of $|V_A|$ into (5) the following relations (6) are obtained:

$$|V_A| = 2k, \quad |E_A| = 3k, \tag{6}$$

where k is a natural number. In this manner, different Assur groups are obtained by assigning different values to k. Reference [16] presents a specific algorithm for the synthesis of Assur groups. The simpler Assur group (see Table 2, for $k = 1$) consists of two elements or vertices and three kinematic pairs or edges. The edge connecting vertices 1 and 2 is called *inner*. The edges that connect the group with other Assur groups or the driving-elements group are called *external*.

The class of an Assur group can be defined according to the number k. In this manner, the Assur group with two vertices ($k = 1$) will be called a group of the first class; the group of four vertices ($k = 2$) will be called a group of the second class; the group of six vertices ($k = 3$) will be called a group of the third class and so on. It is also common to define the order r of an Assur group as the number of external edges of its representing graph. It should be noted that two or more Assur groups sharing the same class k may have a different order r as it can be seen in Table 2. The dashed vertices of the Assur groups graph representation do not belong to the Assur groups themselves, but to those groups or driving-elements group they might connect with.

If the class k of an Assur group and its order r are known, then the following information is obtained from (3), (5) and (6):

$\lvert V_A \rvert = 2k$,	the number of vertices of the group,
$\lvert E_A \rvert = 3k$,	the number of edges of the group,
r,	the number of external edges of the group,
$L = k + 1$,	the number of independent loops, and
$L_c = k + 1 - r$,	the number of closed loops.

3.3 Goal of disaggregation

The goal of disaggregation of the presented algorithm is formally stated in this section. For this purpose, the required graph operations are first defined and implemented into a mathematical definition of the Assur groups disaggregation of mechanisms.

3.3.1 Graph operations

3.3.1.1 Merge vertices A couple of vertices v_i and v_j of a graph G are said to be merged, if the vertices are replaced with a new one, such that every edge incident with v_i or v_j, or with both is incident with the new vertex.

3.3.1.2 Graph union Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two sub-graphs of a graph $G = (V, E)$. The union of the sub-graphs G_1 and G_2, $G_1 \cup G_2$, is another sub-graph $G_3 = (V_3, E_3)$ such that $V_3 = V_1 \cup V_2$ and $E_3 = E_1 \cup E_2$. The following properties are applied to the graph union operator $\cup$.

(a) Commutative property. This is, $G_1 \cup G_2 = G_2 \cup G_1$.
(b) Left-associative property. This is, $G_1 \cup G_2 \cup G_3 = ((G_1 \cup G_2) \cup G_3)$.

3.3.1.3 Graph subtraction Let $G_A = (V_A, E_A)$ be a sub-graph of a graph $G = (V, E)$. The subtraction of the graphs $G \ominus G_A$ is another sub-graph $G_B = (V_B, E_B)$ such that $V_B = V - V_A$ is the set resulting from deleting all the vertices V_A from V and $E_B = E - E_A$ is the set resulting from deleting all the edges E_A from E. The Left-associative property is applied to the graph subtraction operator $\ominus$. This is, $G_1 \ominus G_2 \ominus G_3 = ((G_1 \ominus G_2) \ominus G_3)$.

3.3.2 Assur groups disaggregation of planar mechanisms

Let $G = (V, E)$ be the graph representing a planar mechanism and $G_D = (V_D, E_D)$ be the graph representing the driving-elements group. There exists a sequence of Assur groups $[G_{A1}, G_{A2}, \ldots, G_{An}]$ such that the result of joining the driving-elements group $G_D = (V_D, E_D)$ with the sequence of its Assur groups is G. This sequence represents the kinematic structure of the mechanism and it is the disaggregation into Assur groups. It is expressed through the graph union operation as shown in (7):

$$G = G_D \cup \left\{ \bigcup_{i=1}^{n} G_{Ai} \right\}, \tag{7}$$

where n is the number of Assur groups present in the mechanism. Note that the disaggregation into Assur groups of a mechanism depends on the selection of its driving-elements group. Furthermore, there are cases in which the sequence is partial since two or more groups share the same hierarchy in the system group classification.

3.4 Algorithm for graph-based disaggregation of mechanisms into Assur groups

The result of application of the Algorithm 1 to a graph $G = (E, V)$ meeting conditions described in Sect. 3.2.1 is a sequence of kinematic chains that satisfies its Assur groups disaggregation as it is described in Sect. 3.3.2.

Algorithm 1 Disaggregation of planar mechanisms into Assur groups

Require: $G = (V, E)$ Graph representing the planar mechanism.
$G_D = (V_D, E_D)$ Sub-graph of G representing the primary-elements groups of G

Ensure: $Assur\_disaggregation = [G_{A1}, G_{A2}, \ldots, G_{An}]$ Assur disaggregation is a sequence such that the graphs G_{Ai} are sub-graphs of G and $G = G_D \cup \{\bigcup_{i=1}^{n} G_{Ai}\}$

```
1: % Merge each of the driving-elements group vertices (V_D) of G and vertex zero
2: G ← MergeVertices(G, G_D)
3: Assur_disaggregation ← [ ]
4: % Let Φ be the graph having only vertex zero and no edges
5: while G ≠ Φ do
6:    % Search starts with Assur class 1
7:    k ← 1
8:    Assur_gr_found ← FALSE
9:    while k ≤ (|V|−1)/2 and Not Assur_gr_found do
10:      % Generate a set of sub-graphs of G candidates for being Assur groups of the class k
11:      S ← CandidatesForClass(G, k)
12:      while S and Not Assur_gr_found do
13:         G_A ← First(S)
14:         if AssurCondition(G, G_A) then
15:            Assur_gr_found ← TRUE
16:            % Update the disaggregation
17:            Assur_disaggregation ← [Assur_disaggregation, G_A]
18:            % Update graph G, the symbol ⊖ stands for graph subtraction
19:            G ← G ⊖ G_A
20:         else
21:            % Remove G_A from the set S
22:            S ← S − G_A
23:         end if
24:      end while
25:      k ← k + 1
26:   end while
27: end while
```

The statement $S = \mathrm{CandidatesForClass(G, k)}$ (Algorithm 1, line 11) is a combinatorial function that generates a set S containing all the sub-graphs of G that have a number of vertices which is equal to $2k$. Efficient algorithms to find all the paths with a prescribed length (e.g. $2k$) are available. However, it is unusual to find mechanisms with an elevated number of links (vertices), then an algorithm with combinatorial complexity results in an acceptable alternative. Computational complexity of this function is estimated on Sect. 3.4.1. The statement $Condition = \mathrm{AssurCondition(G, G_A)}$ (Algorithm 1, line 14) is a boolean function that checks whether the graph G_A represents an Assur group or not. Two conditions are verified:

(a) If G_A is connected and its number of edges $|E_A|$ equals to $3k$.
(b) If $W(G \ominus G_A) = 0$.

If conditions a and b are satisfied, then (4) and (5) are verified and G_A represents a sub-graph (Assur group) of the system group classification.

3.4.1 Computational complexity of the algorithm

The number $|V|$ of vertices of the graph G is a measure of input size. Each of the inner loop statements 12–24, including the conditional and looping statements, requires a constant amount of time $O(1)$. By the sum rule, the combined computational complexity of this group of statements is $O(\max(\mathrm{O}_{12}, \ldots, \mathrm{O}_{24})) = \mathrm{O}(1)$.

In the next steps, we assume the worst-case scenario complexities. The number of iterations of the loop of lines 12–24 is assumed to be $|S|$. By the product rule, the time spent in this loop is $O(|S| \times 1)$ which is $O(|S|)$. S is the set of sub-graphs of G having a number of vertices equal to $2k$ and its size $|S|$ is given by (8):

$$|S| = \frac{(|V| - 1)!}{(2k)!(|V| - 2k - 1)!}, \tag{8}$$

where the term $(|V| - 1)$ corresponds to the number of vertices of G when vertex zero is not included.

Regarding the loop of lines 9–26, statement 11 generates the set of sub-graphs of G having a number of vertices equal to $2k$ and therefore, it requires an amount of time $O(|S|)$. Statement 25 requires an amount of time equal to $O(1)$. Finally, statements 12–24, corresponding to the inner loop, require an amount of time $O(|S|)$, as stated before. By the sum rule, the combined computational complexity of this group of statements is $O(\max(O_9, \ldots, O_{26})) = O(|S|)$, which is the total amount of time the loop takes for each iteration. Hence, the number of operations of the loop is given by (9):

$$\sum_{k=1}^{\frac{|V|-1}{2}} |S| = \sum_{k=1}^{\frac{|V|-1}{2}} \frac{(|V|-1)!}{(2k)!(|V|-2k-1)!} = \sum_{k=1}^{\frac{|V|-1}{2}} \binom{|V|-1}{2k}. \tag{9}$$

The computational complexity of the loop can be bounded above by (10):

$$\sum_{k=0}^{|V|} \binom{|V|}{k}. \tag{10}$$

By the implementation of the binomial theorem of Newton, it can be proved that the total amount of time of the loop (lines 9–26), takes a time proportional to the power of the number of vertices ($|V|$) of the graph G [see (11)]:

$$\sum_{k=0}^{|V|} \binom{|V|}{k} = 2^{|V|}. \tag{11}$$

Let us consider the outer loop of lines 5–27. Each of the statements 7, 8 takes an amount of time $O(1)$, while statements 9 - 26 take an amount of time $O(2^{|V|})$ as stated in (11). By the sum rule, the combined computational complexity of this group of statements is $O(\max(O_5, \ldots, O_{27})) = O(2^{|V|})$.

The outer loop is executed a number of times equal to the number of Assur groups present in the disaggregation of G. Since we are looking for the worst-case computational complexity, we assumed a maximal number of Assur groups forming G. This is, that all groups are assumed to have class $k = 1$ and therefore, each having 2 vertices. According to this, for a total number of free vertices equal to $(|V| - 1)$, the total number of Assur groups of the first class is $(|V| - 1)/2$. By the product rule, the time spend in the outermost loop is $O(2^{|V|} \times (|V| - 1)/2)$, so the total operations number of the program is bounded by $O(|V|2^{|V|})$.

3.5 Proof of correctness

For the outermost loop between lines 5–27, the invariant *Inv* that describes the state of the program when iteration j starts is (12):

$$\{Inv : G_j = (V_j, E_j) \ \wedge \ 3(|V_j| - 1) = 2|E_j|\}, \tag{12}$$

Equation (12) means that the mechanism at the start of iteration j is described by the graph G_j and it has zero DOF, or $3(|V_j| - 1) = 2|E_j|$. The term $(|V| - 1)$ results from taking vertex zero out of the count of the DOF of G. The algorithm starts with a graph G representing the mechanism in iteration 1 ($G = G_1$). In each iteration, G_j ($j = 1, \ldots$) becomes smaller because an Assur group graph G_{Aj} is subtracted from it until eventually G_j equals Φ, the empty graph. The algorithm stops in a finite number of iterations. In each iteration, G_j satisfies the invariant *Inv* in (12). The disaggregation $[G_{A1}, G_{A2}, \ldots]$ of the original G has been calculated.

Let $G_{Aj} = (V_{Aj}, E_{Aj})$ be the Assur group graph identified in the j-th iteration of the inner loop (Algorithm 1, line 12). Being G_{Aj} an Assur group, it satisfies that $3|V_{Aj}| = 2|E_{Aj}|$. The algorithm then subtracts G_{Aj} from G_j, resulting in the new graph $G_{j+1} = G_j \ominus G_{Aj}$ (Algorithm 1, line 12). To prove that the loop correctly maintains the invariant *Inv*, we must show that the mechanism remaining after the subtraction, G_{j+1}, still has zero DOF (i.e. it satisfies $3(|V_{j+1}| - 1) = 2(|E_{j+1}|)$. The proof is in the Eqs. (13) to (15):

$$|V_{j+1}| = |V_j| - |V_{A_j}| \ \Rightarrow \ |V_j| = |V_{j+1}| + |V_{A_j}|, \tag{13}$$

$$|E_{j+1}| = |E_j| - |E_{A_j}| \ \Rightarrow \ |E_j| = |E_{j+1}| + |E_{A_j}|, \tag{14}$$

$$\begin{aligned} &3(|V_j| - 1) = 2|E_j| \ \Rightarrow 3|V_{j+1}| + 3|V_{A_j}| - 3 \\ &= 2|E_{j+1}| + 2|E_{A_j}| \ \Rightarrow 3|V_{A_j}| = 2|E_{A_j}| \\ &\Rightarrow 3(|V_{j+1}| - 1) = 2|E_{j+1}|, \end{aligned} \tag{15}$$

therefore, reaching the invariant for iteration $j + 1$. We have proved that each iteration j preserves the invariant *Inv* of the loop.

Equations (13) and (14) reflect the fact that the size of the graph G_j decreases in each iteration by exactly the size of the Assur group graph G_{Aj} subtracted from

G_j. Equation (15) uses the fact that G_j has zero DOF ($3(|V_j|-1) = 2|E_j|$) and the decrement in size of sets V_j and E_j. Equation (15) also uses the Assur character of G_{Aj} ($3|V_{Aj}| = 2|E_{Aj}|$) to prove that the new graph G_{i+1} has, again, zero DOF. The reader interested in graph subtraction properties may wish to refer to Sect. 3.3.1.

It is clear that both inner loops are correct. In the loop of statements 5–26, the function CandidatesForClass generates a set of Assur candidates with class k. After that, if an Assur group is not found, then the loop increases the value of the class by one. Since k is upper bounded by the number $\left(\frac{|V|-1}{2}\right)$, we guarantee that this loop is executed a finite number of times. As for the loop of statements 12 - 24, the functions First and AssurCondition carry out the extraction and evaluation of each candidate in the set. The loop ends when all candidates have been evaluated or when a candidate graph meets the Assur group condition.

4 Results

4.1 Test example

In this section, we use Algorithm 1 to determine a system group classification (disaggregation into Assur groups) of a 12-bar mechanism with one DOF. The mechanism is also used in [3] as a test example of the structural analysis of mechanisms. Figure 4 shows the structural representation of the aforementioned mechanism, together with its representing graph $G = (V, E)$.

The notation S_i represents the set S in the i-th iteration of the loop in lines 9–26. The first step in Algorithm 1 merges the driving-element group with vertex zero. Figure 5 shows the result of this operation. Now the algorithm starts searching for Assur groups with class k equals to 1. At this point, the function CandidatesForClass generates a set S_1 of candidates G_A for being Assur groups, each having a number of vertices $|V_A|$ equal to 2. Table 3 summarizes the set S_1.

The number of candidates of S_1 is given by $\binom{|V|-1}{2k}$ and is equal to 45. The last column of Table 3 corresponds to the output of the function AssurCondition after each candidate G_A is evaluated.

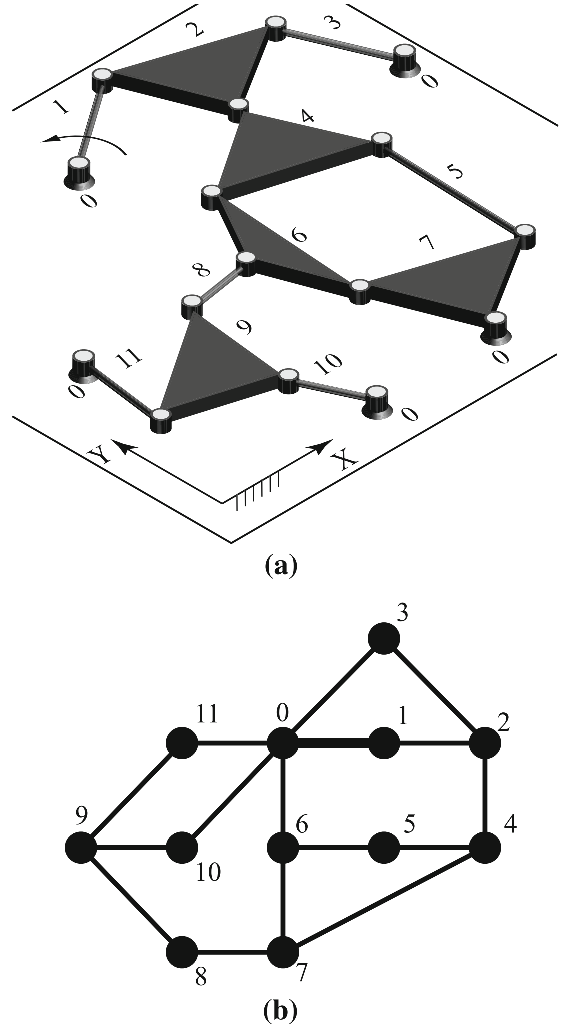


Fig. 4 12-bar planar mechanism presented in [3]. **a** Structural representation. **b** Graph representation of (**a**)

The algorithm increases the value of the Assur class by 1 and starts searching for Assur groups with class $k = 2$, since none of the candidates in S_1 meets the AssurCondition. The new set S_2 of Assur group candidates is generated by the function CandidatesForClass (see Table 4). Each of the candidates in S_2 has a number of vertices equal to four. The last candidate of the set S_2 (Table 4, candidate number 210), consisting of vertices 8, 9, 10 and 11 meets the AssurCondition, and therefore it is an Assur group.

The Assur group found in line 15 is subtracted from the graph G and the algorithm starts again searching for Assur groups with class $k = 1$. The function CandidatesForClass generates a new set S_3 of Assur group candidates. However, none of the candidates of

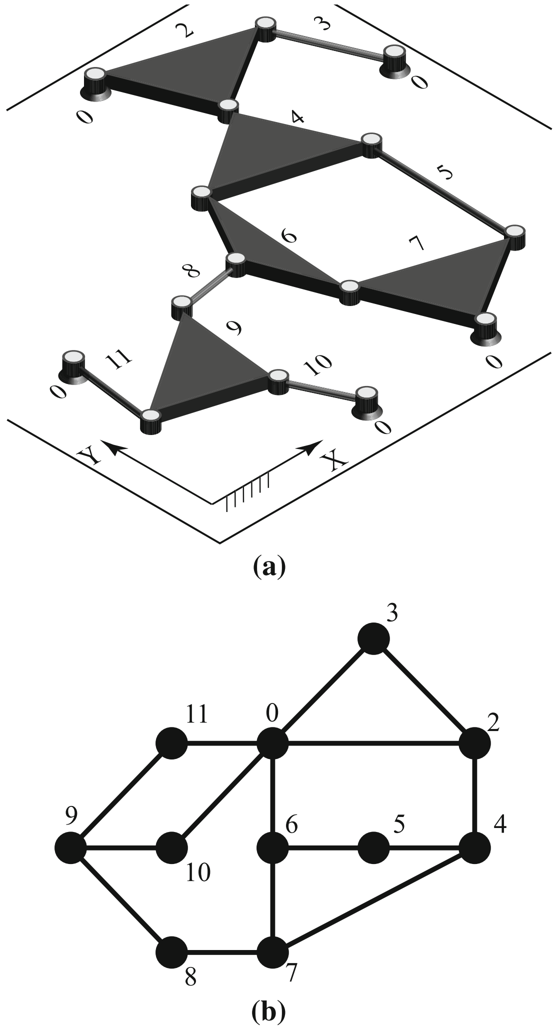


Fig. 5 12-bar planar mechanism: Merge of the driving-elements group with vertex zero. **a** Structural representation. **b** Graph representation of (**a**)

S_3 meets the AssurCondition and the Assur class value increases by 1. Therefore, the algorithm searches Assur groups with class $k = 2$.

A set S_4 of Assur group candidates with class two is generated by the function CandidatesForClass. In this case, candidate number 15 presented in Table 5, consisting of vertices 4, 5, 6 and 7, meets the AssurCondition and it is subtracted from the graph G.

The procedure is iterated until all the Assur groups are subtracted from G, which eventually becomes the trivial graph Φ. Table 6 illustrates the progress of the algorithm towards termination.

The output of the algorithm is a sequence consisting of three Assur groups. The first two elements have class $k = 2$ meanwhile the last one has class $k = 1$. The disaggregation into Assur groups of this mechanism is obtained through the graph union of such groups starting from the last founded group and ending with the first one.

4.2 Implementation

Implementation of Algorithm 1 is briefly described in Sects. 4.2.1 to 4.2.3.

4.2.1 Inputs

The graph representing a mechanism is portrayed by its adjacency matrix. For the sake of simplicity, the input links are removed and the released links are paired up with the fixed link before constructing the matrix, e.g., the matrix (16) represents the mechanism in Fig. 4 after the removal of the input link (Fig. 5).

4.2.2 Algorithm implementation

Adjacency matrix is treated by converting each binary-pattern row to a decimal integer representation. Candidate Assur group to be evaluated is also represented by a binary pattern. Specifically, generation of candidate Assur group is performed by

Table 3 Assur groups disaggregation of a 12-bar mechanism: Set S_1 of Assur candidates with $k = 1$

G_A	11	10	9	8	7	6	5	4	3	2	$\|V_A\|$	$\|E_A\|$	AC[a]
1	0	0	0	0	0	0	0	0	1	1	2	4	F
2	0	0	0	0	0	0	0	1	0	1	2	5	F
3	0	0	0	0	0	0	0	1	1	0	2	5	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
45	1	1	0	0	0	0	0	0	0	0	2	4	F

[a] AssurCondition (AC)

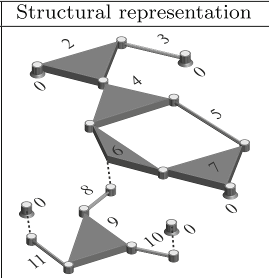
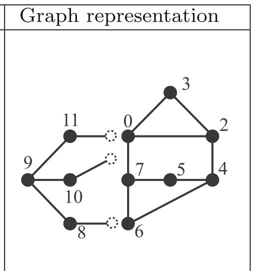
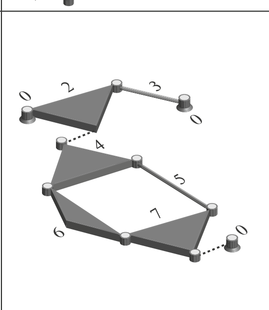
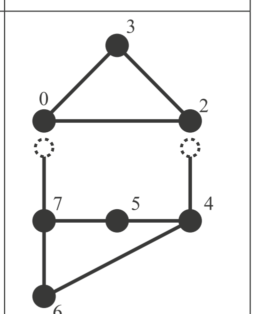
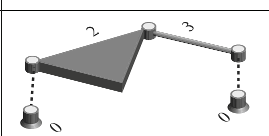
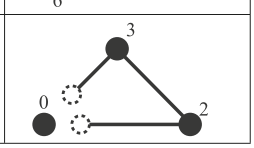
Table 4 Assur groups disaggregation of a 12-bar mechanism: Set S_2 of Assur candidates with $k = 2$

G_A	11	10	9	8	7	6	5	4	3	2	$\|V_A\|$	$\|E_A\|$	AC
1	0	0	0	0	0	0	1	1	1	1	4	7	F
2	0	0	0	0	0	1	0	1	1	1	4	9	F
3	0	0	0	0	0	1	1	0	1	1	4	8	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	0	0	0	0	1	1	1	1	0	0	4	7	F
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
210	1	1	1	1	0	0	0	0	0	0	4	6	T

Table 5 Assur groups disaggregation of a 12-bar mechanism: Set S_4 of Assur candidates with $k = 2$

G_A	7	6	5	4	3	2	$\|V_A\|$	$\|E_A\|$	AC
1	0	0	1	1	1	1	4	7	FALSE
2	0	1	0	1	1	1	4	9	FALSE
3	0	1	1	0	1	1	4	8	FALSE
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
15	1	1	1	1	0	0	4	6	TRUE

Table 6 Assur groups disaggregation of a 12-bar mechanism: summary of the analysis by means of Algorithm 1

i	Structural representation	Graph representation
3		
2		
1		

implementing Algorithm 1, lines 5–27, beginning with the Assur group class 1 ($k = 1$), evaluating the combinatorial of links by bit shift operation on the binary pattern, advancing sequentially to upper classes, and bounded by the size of the adjacency matrix. Once a candidate is corroborated to be an Assur group, then the adjacency matrix is updated and the candidate is stacked first in an Assur group registry. The numbers of links and joints for a particular class are automatically determined by (6). This implementation requires neither databases, nor tables of characteristic data for the candidate Assur group. Our algorithm relies on the mathematical relation of number of edges and vertices representing a kinematic chain, using graph characteristics exclusively. Graph operations described in Sect. 3.3.1 are implemented by bitwise operations (and, or, not, shift, complement).

4.2.3 Outputs

The output of the implementation is a stack (matrix) which contains the binary registry of the Assur groups forming the mechanism. 1 is used to represent a link. The sequence of the rows corresponds to a sequence in which the groups are added to, theoretically, form the mechanism. First row corresponds to the first Assur group in the sequence and so on. Columns in the stack correspond to columns in the adjacency matrix after the removal of input links, e.g., matrix (17) represents the output of the implementation using matrix (16) as the input. A sequence of 3 Assur groups is obtained: the first Assur group is class 1 formed by links 2 and 3, the second is class 2 formed by links 4 - 6 - 7 and 8, and the third is class 2 formed by links 8 - 9 - 10 and 11; which correspond to the results described in the test example 4.1.

links. Since in actual mechanisms the number of links is rarely numerous, then the execution stands for a realistic solution.

Applying the algorithm to a graph not only provides information about the Assur groups within its representing mechanism, but also about the sequence in which those groups are connected. This is useful when attempting to perform a modular approach for the kinematic and force analysis of the mechanism.

Our implementation currently addresses joints allowing 1 DOF. However, in the kinematics sense, it is possible to express higher pairs in terms of a set of lower pairs. Therefore, it is possible to use our procedure to analyze multi-DOF joint planar mechanisms.

Future work will address the application of the proposed algorithm to planar mechanisms with floating input links.

$$G = \begin{array}{c} \text{Link} \\ 0 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array} \tag{16}$$

$$A = \begin{array}{c} \frac{\text{Link}}{\text{Sequence}} \\ 1^{st} \\ 2^{nd} \\ 3^{rd} \end{array} \begin{array}{c} \begin{array}{ccccccccccc} 0 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array} \tag{17}$$

5 Conclusions and future work

A new graph-based algorithm for the disaggregation of planar mechanisms into Assur groups has been implemented. Unlike other procedures, neither previous data manipulation nor visual inspection is required. In this manner, the algorithm can be systematically applied to complex mechanisms. The number of operations of the algorithm execution is bounded by $O(|V|2^{|V|})$, where $|V|$ is the number of

Acknowledgments The authors wish to acknowledge the technical support by Ing. Ricardo Serrano Salazar during the algorithm's implementation. Financial support for this work was provided by the Colombian Administrative Department of Science, Technology and Innovation (COLCIENCIAS), and the Colombian National Service of Learning (SENA), Grant 1216-479-22001. The authors gratefully acknowledge this support.

Appendix: Kinematic structure of planar mechanisms with higher pairs

Kinematic and structural analysis of planar mechanisms can be done by replacing higher pairs with lower pairs. Reference [1] establishes the following condition that must be accomplished by the replacement: the equivalent mechanism preserves: a. the degree of freedom of the original one, and, b. the relative movement of all links in the particular position, e.g., the mechanism of Fig. 6a includes a higher pair formed by 2 curves *hh* and *kk*. The equivalent mechanism is constructed by tracing a normal *nn* at the contact point *C* and determining the curvature centers *H* and *K* of the corresponding curves. The equivalent has four links, where the higher pair was replaced by link 3, which forms lower pairs in *H* and *K*, Fig. 6b. The degree of freedom is preserved and the mechanism is exclusively formed by lower pairs.

Once the equivalent mechanism is determined, then it is possible to analyze the kinematic structure, for example, representing the mechanism by a graph and using Algorithm 1.

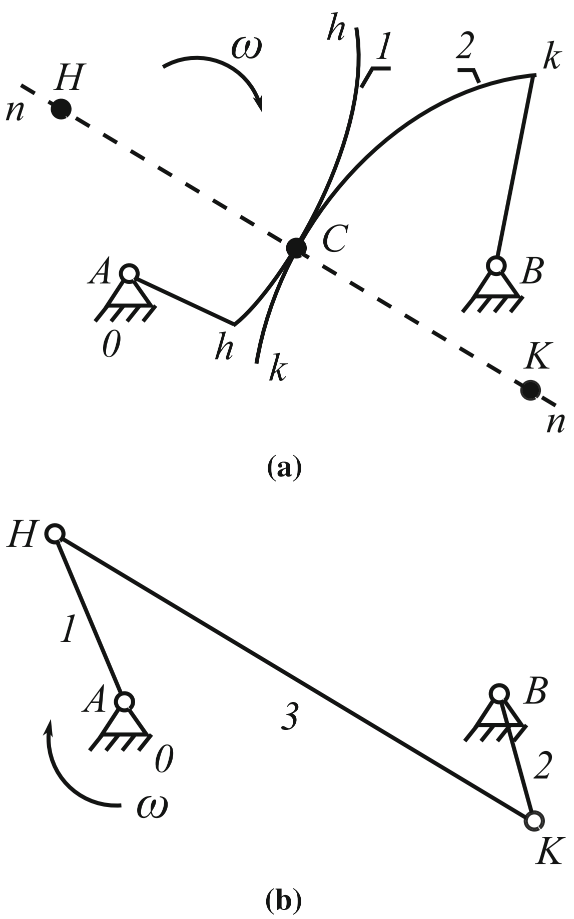


Fig. 6 Equivalence of a mechanisms with higher pair. **a** 1 DOF mechanism with higher pair. **b** Instantly equivalent mechanism of (**a**)

References

1. Artobolevski I (1988) Teoría de mecanismos y máquinas. Nauka, Moscú (in Spainsh)
2. Baránov G (1979) Curso de la teoría de mecanismos y máquinas. MIR, Moscú (in Spanish)
3. Buśkiewicz J (2006) A method of optimization of solving a kinematic problem with the use of structural analysis algorithm (SAM). Mech Mach Theory 41(7):823–837. doi:10.1016/j.mechmachtheory.2005.10.003
4. Campos A, Guenther R, Martins D (2009) Differential kinematics of parallel manipulators using Assur virtual chains. P I Mech Eng C-J Mech 223(7):1697–1711. doi:10.1243/09544062JMES1156
5. Chen G, Wang H, Zhao K, Lin Z (2009) Modular calculation of the Jacobian matrix and its application to the performance analyses of a forging robot. Adv Robot 23(10):1261–1279. doi:10.1163/156855309X462574
6. Durango S, Calle G, Ruiz O (2010) Analytical method for the kinetostatic analysis of the second-class RRR Assur group allowing for friction in the kinematic pairs. J Braz Soc Mech Sci Eng 32(3):200–207. doi:10.1590/S1678-58782010000300002
7. Galletti C (1979) On the position analysis of Assur's groups of high class. Meccanica 14(1):6–10. doi:10.1007/BF02134963
8. Galletti C (1986) A note on modular approaches to planar linkage kinematic analysis. Mech Mach Theory 21(5):385–391. doi:10.1016/0094-114X(86)90086-8
9. Han L, Liao Q, Liang C (2000) Closed-form displacement analysis for a nine-link Barranov truss or a eight-link Assur group. Mech Mach Theory 35(3):379–390. doi:10.1016/S0094-114X(99)00016-6
10. Hansen M (1996) A general method for analysis of planar mechanisms using a modular approach. Mech Mach Theory 31(8):1155–1166. doi:10.1016/0094-114X(96)84606-4
11. Kolovsky M, Evgrafov A, Semenov Y, Slousch A (2000) Advanced theory of mechanisms and machines. Springer, Berlin
12. Mitsi S, Bouzakis K, Mansour G, Popescu I (2003) Position analysis in polynomial form of planar mechanisms with

Assur groups of class 3 including revolute and prismatic joints. Mech Mach Theory 38(12):1325–1344. doi:10.1016/S0094-114X(03)00090-9
13. Mitsi S, Bouzakis KD, Mansour G (2004) Position analysis in polynomial form of planar mechanism with an Assur group of class 4 including one prismatic joint. Mech Mach Theory 39(3):237–245. doi:10.1016/S0094-114X(03)00115-0
14. Mlynarski T (1996) Position analysis of planar linkages using the method of modification of kinematic units. Mech Mach Theory 31(6):831–838. doi:10.1016/0094-114X(95)00120-N
15. Mruthyunjaya T (2003) Kinematic structure of mechanisms revisited. Mech Mach Theory 38(4):279–320. doi:10.1016/S0094-114X(02)00120-9
16. Peisakh E (2007) An algorithmic description of the structural synthesis of planar Assur groups. J Mach Manuf Reliab 36(6):505–514. doi:10.3103/S1052618807060015
17. Popescu I, Marghitu D (2008) Structural design of planar mechanisms with dyads. Multibody Syst Dyn 19(4):407–425. doi:10.1007/s11044-007-9099-6
18. Popescu I, Marghitu D, Stoenescu E (2006) Kinematic chains with independent loops and spatial system groups. Arch Appl Mech 75(10):739–754. doi:10.1007/s00419-006-0064-2
19. Reich Y, Shai O (2012) The interdisciplinary engineering knowledge genome. Res Eng Des 23(3):251–264. doi:10.1007/s00163-012-0129-x
20. Saura M, Celdran A, Dopico D, Cuadrado J (2014) Computational structural analysis of planar multibody systems with lower and higher kinematic pairs. Mech Mach Theory 71:79–92. doi:10.1016/j.mechmachtheory.2013.09.003
21. Servatius B, Shai O, Whiteley W (2010) Combinatorial characterization of the Assur graphs from engineering. Eur J Comb 31(4):1091–1104. doi:10.1016/j.ejc.2009.11.019
22. Tischler C, Samuel A, Hunt K (1995) Kinematic chains for robot hands-II. kinematic constraints, classification, connectivity, and actuation. Mech Mach Theory 30(8):1217–1239. doi:10.1016/0094-114X(95)00044-Y
23. Wang H, Lin Z, Lai X (2008) Composite modeling method in dynamics of planar mechanical system. Sci China Ser E 51:576–590. doi:10.1007/s11431-008-0035-7
24. Wang H, Eberhard P, Lin Z (2010) Modeling and simulation of closed loop multibody systems with bodies-joints composite modules. Multibody Syst Dyn 24:389–411. doi:10.1007/s11044-010-9208-9
25. Wohlhart K (2010) Position analysis of normal quadrilateral Assur groups. Mech Mach Theory 45(9):1367–1384. doi:10.1016/j.mechmachtheory.2010.03.002
26. Zhang Q, Zou H, Guo W (2006) Position analysis of higher-class Assur groups by virtual variable searching and its application in a multifunction domestic sewing machine. Int J Adv Manuf Technol 28:602–609. doi:10.1007/s00170-004-2379-x