The 2nd Joint International Conference on Multibody System Dynamics
May 29–June 1, 2012, Stuttgart, Germany

Computational Analysis of the Structure of Planar Multibody Systems with Lower and Higher Pairs

M. Saura*, **A. Celdrán***, **D. Dopico**†, **J. Cuadrado**†

* Department of Mechanical Engineering
Universidad Politécnica de Cartagena
Doctor Fleming, s/n, 30202 Cartagena, Spain
msaura.sanchez@upct.es

Laboratorio de Ingeniería Mecánica
University of La Coruña
EPS, Mendizábal s/n, 15403 Ferrol, Spain
javicuad@cdf.udc.es; ddopico@udc.es

ABSTRACT

Kinematic and dynamic modelling of multibody systems requires an initial stage of topological recognition or structural analysis, in which the analyst must identify the coordinates needed to define the model and to set the constraint equations which relate them and which will allow the determination of its response. Both global and recursive formulations require, at the present moment, a high level of training for carrying out this stage successfully. This initial phase could be solved quickly, safely and automatically, by determining the kinematic structure of the multibody system; i.e. identifying the basic kinematic chains (Structural Groups) of which it is composed. On the basis of known graph-analytical methods for structural analysis, an implemented and evaluated computational method that determines the kinematic structure of a multibody system by dividing it into a set of structural groups, simply from its adjacency matrix, is developed in this work. Modelling a multibody system from its kinematic structure allows to choose of any type of coordinates (relative, natural or reference point) and to employ kinematic and dynamic formulations more appropriate for the selected coordinates and the kind of problem at hand. The developed algorithm has been applied to a large number of mechanical systems of different complexity, offering the same kinematic structure as the obtained through the application of graph-analytical methods.

1 INTRODUCTION

To address the simulation of multibody systems, an initial stage devoted to topological and/or structural analysis is always required. The topological analysis identifies the number and type of the kinematic pairs that connect the system bodies. The structural analysis starts from the known topology and, by means of equations, determines the mechanism kinematic structure. Currently, the structural analysis is used to identify either rigid kinematic chains and isomorfisms in structural synthesis problems [7, 8], as closed loops and desmodromic chains in kinematic and dynamic analysis problems [1, 5].

In global formulations, a simple topological analysis is enough to identify the degrees of freedom that are constrained by each type of kinematic pair. The dependent coordinates used to define the model (i.e. $[x_B\ y_B\ x_C\ y_C\ \theta_1]$ in Fig. 1.a are then related through the constraint equations due to rigid-body and kinematic-pair conditions. In recursive formulations, besides the topological analysis, a structural analysis must be carried out aimed to identify the mechanism closed loops. Once identified, they are opened so as to yield a tree-like structure of the mechanism (Fig. 1.b). Then, the kinematic relations among bodies due to the joints connecting them can be defined, along with the loop-closure equations which relate the system dependent ($\theta_2\ \theta_3$) and independent (θ_1) coordinates.

The topological recognition carried out for this basic quadrilateral linkage is quite simple, but it may turn into a very complicated task in more complex systems, planar and spatial. This process relies on the analyst's ability to select the appropriate number and type of coordinates to describe the problem, and to identify and formulate the needed constraint equations to solve it. In this work we propose to apply the structural analysis methods to automate this initial phase. The computational algorithm divides a multibody system into simple kinematic chains called Structural Groups (SG-I y SG-II, Fig. 1.c). Once the division into SG is obtained, the variables and constraint equations that each SG introduces into the system are determined. In Fig. 1.c, SG-I introduces ($\theta_1\ x_B\ y_B$); if θ_1 is known, then ($x_B\ y_B$) can be calculated. Now in SG-II ($x_B\ y_B\ x_D\ y_D$) are known and either ($x_C\ y_C$) or ($\theta_2\ \theta_3$) can be determined. Both, the variables and the corresponding constraint equations introduced by each SG can be included in formal kinematic

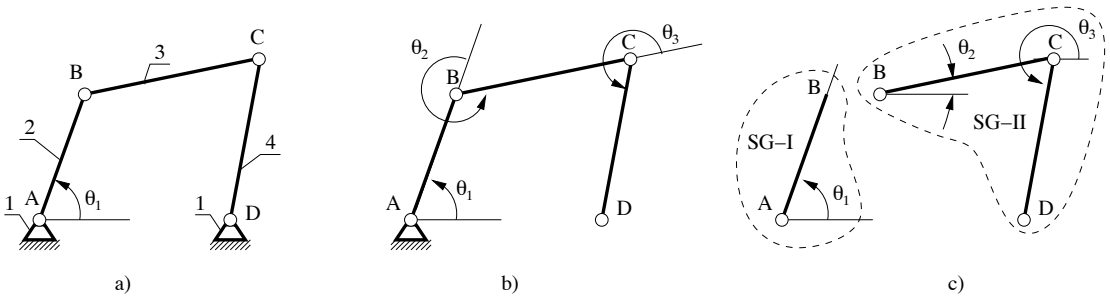

Figure 1. Four bar linkage with given input θ_1. b) Tree-like structure and variables. c) Split into structural groups: SG-I and SG-II.

and dynamic formulations to obtain the response of the whole system. This last step is currently under development.

In spite of its utility, the use of methods based on kinematic structure has not been deeply explored yet. In the literature search carried out, there have been found few structural analysis methods applied to structural synthesis [1, 6], very few to kinematic analysis [1], and none to the dynamic analysis of mechanisms. Furthermore, none of the reviewed computational methods, either for synthesis or for analysis, considers higher pairs in the mechanism except gear trains.

To introduce the computational method developed in this work, we first review, in section 2, the graph-analytic methods for structural analysis that obtain the kinematic structure of planar multibody systems. In section 3, we present in detail the steps and algorithms needed to implement the proposed method. Section 4, shows the results obtained when applying the graph-analytic and the computational method to a complex mechanism. Finally, in section 5 the main conclusions of this study are established.

2 STRUCTURAL ANALYSIS

The structural analysis is the study of the nature of connection among the members of a mechanisms and its mobility [4]. It is concerned with the fundamental relationships between the degrees of freedom, the number and type of joints, and the number of links of a mechanism. These relationships can be employed to identify kinematic chains in a mechanical system under different topological criteria. Other authors refer to structural analysis as the division of a mechanisms into basic kinematic chains called Structural Groups. A Structural Group (SG) is a kinematic chain whose number of independent chain inputs n_c coincides with the number of degrees of mobility L_c, ($n_c = L_c$), and that cannot be splitted into SG of smaller number of links [3]. Under this condition and using the Grübler criteria to determine the mobility of a kinematic chain, the analytical condition that a kinematic chain must satisfy to be SG is obtained (Eq. 1). In Eq. 1, S_c indicates the number of degrees of freedom allowed by the P kinematic pairs formed by their N_m mobile links.

$$S_c - n_c = 3 \cdot (P - N_m) \tag{1}$$

Graph-analytical methods for structural analysis use Eq. 1 to obtain the kinematic structure of a mechanism with the help of its structural graph: a graph associated with the mechanism that uses the elements of graph theory to show its topology [3, 4, 6]. In the structural graph, vertices corresponds to links and edges to kinematic pairs. The number of edges connecting two vertices corresponds to the mobility of the kinematic pair and equals the degrees of freedom (DOF) or relative movements allowed between them. A number of these edges equal to n_C, independent movements defined between the links of the kinematic pair, become bold lines, referred to as root edges. The Figures 2.a and 2.b show the four-bar linkage's kinematic graph and its corresponding structural graph. The lower pair $(1-2)$ joins vertices 1 and 2 with only one edge. As

there is an input movement q_1 defined between them, this is a root edge. Among the other kinematic pairs of the mechanism there are not input movements and all pairs are grade I (lower pairs), therefore all of them are joined with a single edge.

The general sequence of the graph-analytical method used in this work to determine the kinematic structure (Fig. 2.b-g) of a mechanism is described by applying it to the four-bar linkage, and is useful as a basis for the computational method proposed in this work. In this method, we call external pairs those in a kinematic chain which can be used to attach one link of the kinematic chain to a link pertaining to another kinematic chain or to the frame. An internal kinematic pair joins two links of the same kinematic chain. In Figure 1.c, links 2 and 3 form a kinematic chain which is a structural group (SG-II). The rotational joint B joins link 3 to the external link 2, so $(2-3)$ is an external pair. The rotation joint C joins links 3 and 4 of the same kinematic chain, so $(3-4)$ is an internal pair.

1. Frame isolation and DOF assignment. (Fig. 2.c). The frame 1 is isolated. The DOF allowed by each pair in which the frame participates are assigned to the links that form kinematic pair with the frame. This assignment of degrees of freedom is represented in the graph by an assignment arrow directed to the link that receives the DOF (directed edge). Here, the chassis assigns DOF to the links 2 and 4. The links that receive DOF from the frame become *candidate* to be a SG.

2. Search for SG from shorter to larger length. Each of the candidates is checked to satisfy Eq. 1. Here, link 2 is selected (Fig. 2.d). The kinematic pairs P are accounted as the internal pairs and, from the external pairs, only those with a directed edge (a DOF has been assigned). Thus, for this body we find:

   - Kinematic pairs: $P = 1$. Only the external pair $(1-2)$,
   - DOF allowed by kinematic pairs: $S_c = 1$; lower pair $(1-2)$,
   - Input movements: $n_c = 1$; q_1,
   - Movable links: $N_m = 1$. Body 2 is only taken into consideration.

   Replacing these values in Eq. 1 results: $1 - 1 = 3 \cdot (1-1)$. The condition is satisfied and this link is SG.

3. Re-assign DOF. If a kinematic chain forms SG, the DOF of its external pairs are assigned to the corresponding external links. In the example, body 2 is SG, and assigns the DOF $(2-3)$ to the body 3, which is now a new *candidate* (Fig. 2.e). There are no more assignments.

4. Turn to Step 2. Links 3 and 4 are *candidates*. Starting from one candidate, i.e. link 3, the parameters of this kinematic chain are: $S_c = 1$; $n_c = 0$; $N_m = 1$; $P = 1$. After susbtituting in Eq. 1, link 3 shows not to be SG. Link 4 has the same parameters than link 3 so it is not SG either.

   As it is not possible to form a SG with a single link, larger chains have to be considered. Starting from a candidate, i.e. link 3, the chain is expanded by selecting another link that forms kinematic pair with the candidate. The chain $3-4$, whose parameters are: $S_c = 3$; $n_c = 0$; $N_m = 2$; $P = 3$ satisfies equation Eq. 1, and therefore is SG. Finally, the obtained group is marked with a dashed line (Fig. 2.f).

The structure of a mechanism is graphically represented by its *structural diagram* (Fig. 2.g). It is composed by as many circles as SG which have been obtained plus one, corresponding to the frame, which is identified with the number 0. Two parameters are written inside each circle (N_m, n_c) corresponding to the number of movable links and input movements of the SG. A directed arrow joins the circles if any of their links forms a kinematic pair. The arrows are directed in the same way that the DOF have been assigned. This direction indicates in which order the different SG have been obtained, and defines the sequence to be followed to solve, recursively, the kinematics of the complete system.

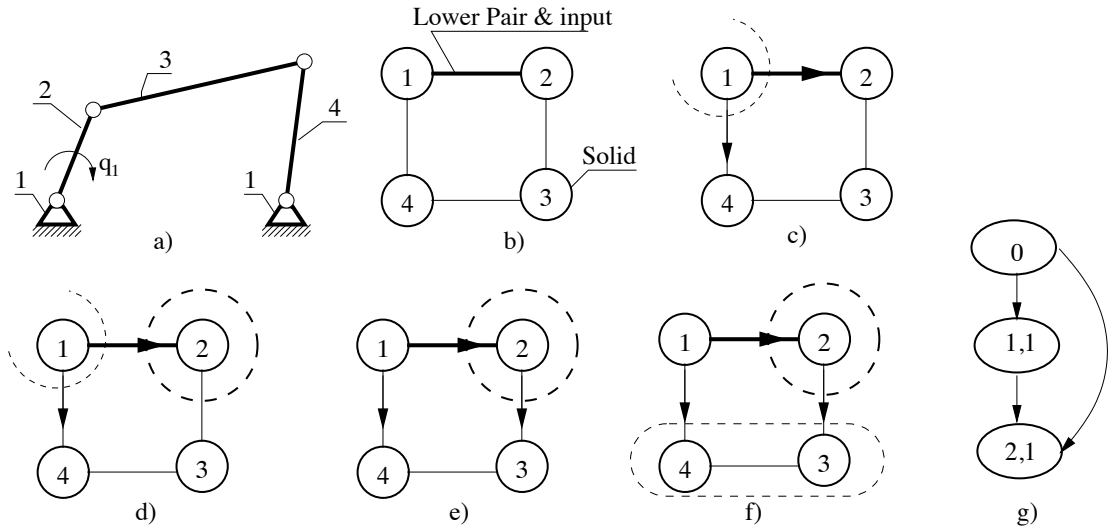

Figure 2. Four-bar linkage. a) Kinematic graph. b) Structural graph. c) to f) steps to perform a structural analysis through its structural graph. g) Structural diagram.

$M(i,j)$	Type of kinematic pair	Number of inputs
0	No pair	-
1	Grade I (lower)	0
2	Grade II (higher)	0
3	Grade I (lower)	1
4	Grade II (higher)	1
5	Grade II (higher)	2

Table 1. Code for topological kinematic pairs and input movements identification.

3 COMPUTATIONAL STRUCTURAL ANALYSIS METHOD AND ALGORITHMS

In the previous section a graph-analytical method has been used to divide a mechanical system into SG and determine the order for their kinematic analysis. This section explains the computational method implemented to obtain the kinematic structure of any planar mechanism. The main program has been divided into seven steps shown in Fig. 3. The description of the steps will be presented later, together with the algorithm's details listed in their corresponding figures.

Step 1: Data provided by the user

The user introduces the adjacency matrix M (symmetric) that collects the topology of the mechanism with N bodies. This matrix, usually used to recognize different closed loops in mechanisms [6, 4], is slightly modified here in order to obtain their structure. Table 1 sets the values to be introduced in $M(i,j)$ for two bodies i, j that form a kinematic pair, and their meaning. This numbering is easy to modify and extend to include new types of pairs, spatial mechanisms or special situations that allow to expand the possibilities of the described algorithm (i.e. pure rolling, multi-revolute joints, etc.), and that are currently under study.

Step 2: Structural transformation

One of the advantages of topological methods in computational dynamic analysis is that they allow the selection of the most appropriate set of independent coordinates to describe the behaviour of the mechanical

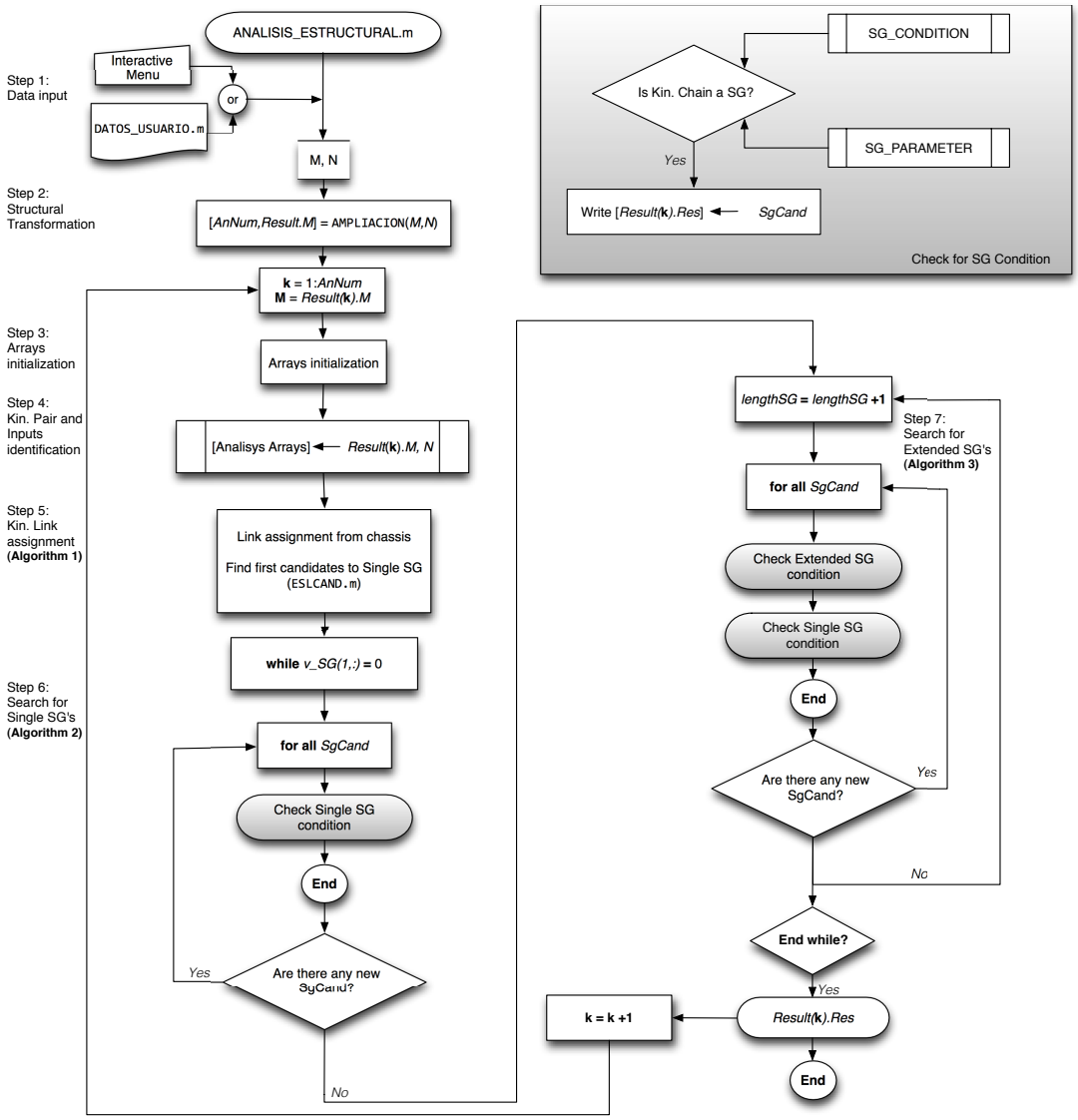

Figure 3. Computational structural analysis algorithm flow chart.

system throughout its cycle of operation [9]. A change in the set of independent coordinates represents a structural transformation and, therefore, the kinematic structure of the system is modified. The proposed method includes the routine `AMPLIACION.m` (optional), which evaluates the $AnNum$ possible structural transformations of the system to analyse, and stores their corresponding adjacency matrixes in the data structure $Result(k).M$. For each adjacency matrix, its structural analysis (steps 3 to 7) and the results are saved in the data structure $Result(k).Res$. Evaluating the different structural transformations obtained, the analyst could select the most advantageous for kinematic and dynamic analyses. For example, the kinematic structure which best allows the parallelization of the calculations, or the one that will avoid singular positions, could be used.

Step 3: Arrays initialization

The variables and arrays needed for computational structural analysis are initialized. The vectors of N dimension (v_{PI}, v_{PS}, v_{LG}) indicate, respectively, the number of lower, higher pairs and input movements that each link participates in. Vector v_{SG} indicates the bodies that belongs to any structural group, and

Algorithm 1: `Assign links and search for candidates`

Result: v_{SgCand}

1.1: **for** $c = 2 : N$ **do**
1.2: | **if** $Result(k).M(1, c) \sim= 0$ **then**
1.3: | | $Result(k).Res(c, 1) = 1$;
1.4: | **end**
1.5: **end**
1.6: `ESLCAND.m`;

Algorithm 2: `Search for Single SG (BSGIND.m)`

2.1: **while** $\exists$ *new* $SgCand$ **do**
2.2: | **forall the** $SgCand$ **do**
2.3: | | **foreach** $j \in (Result.Res(j, SgCand) \sim= 0)$ **do**
| | | `/* identifies link` j `that assigns DOF to` $SgCand$ `*/`
2.4: | | | (N_m, S_c, P, n_c)=`PARSGNIND; /* Calc. param. to evaluate Eq.1 */`
2.5: | | **end**
2.6: | | ($Result.Res$, v_{SgCand})=`CONSGIND; /* Eval Eq.1. Look for new candidates */`
2.7: | **end**
2.8: **end**

v_{SgCand} indicates which bodies are candidates to be SG. Arrays of $(N \times N)$ dimension: a_{PI}, a_{PS}, a_{LG} indicate if its elements (i, j) form a lower pair, higher pair or if they have any input movements. $Results.M$ contains each of the adjacency matrices obtained in Step 2 and $Result.Res$ will store their corresponding kinematic structure.

Step 4: Kinematic pairs and input movements identification

The upper diagonal of the adjacency matrix $Results(k).M$ is analysed. Depending on the value of its elements (first column in Table 1), the arrays a_{PI}, a_{PS} and a_{LG} are generated.

Step 5: Frame isolation and DOF assignment to bodies

The main script performs this function and its operation is shown in Algorithm 1. The first row of the adjacency matrix is analysed. All the bodies joined to the frame receive the corresponding DOF from the later. Then, the function `ESLCAND.m` finds the bodies candidate to be SG, i.e. those which have received any DOF and do not already belong to any structural group v_{SG}. The result is saved in the vector v_{SgCand}. After this DOF assignment, the main script enters in a loop $while - end\ while$ (Fig. 3), which will not stop until all bodies are assigned to some SG. During this loop, the steps 6 and 7 are executed to find the SG of different length.

Step 6: Search for single groups

The vector v_{SgCand} contains the rigid-bodies that are candidates to become single SG. This step runs the loop $for - end$ to determine whether any candidate link is SG. Calling the script `BSGIND.m`, whose detail is shown in Fig. 3, performs the evaluation. In case of finding a SG, the own script will return to assign DOF to the external links and update the vector v_{SgCand}. Before leaving step 6, it checks if there are new

Algorithm 3: `Search for Extended SG (BSGGR.m)`

```
3.1:  lengthSG=1;
3.2:  while ∃ SgCand ∉ LFSG do
3.3:      lengthSG = lengthSG + 1
3.4:      (SGgroup)=BKCH [Result(k).M, Result(k).Res, SgCand]
          /* search for kinematic chains with lengthSG elements.  At least one
          SgCand must be no SG                                                 */
3.5:
3.6:      forall the SGgroup do
              /* identifies and counts the type of DOF between links          */
3.7:          (N_m, S_c, P, n_c)=PSGGR;            /* Calc.  parám.  y evaluate Eq.1 */
3.8:
3.9:          if SGgroup ∈ LFSG then
3.10:             (Result.Res, v_SgCand)=PSGGR;          /* Search for new candidates */
3.11:             Search for single SG;
3.12:         else
3.13:             Clean link assingment between each solids ∈ SGgroup;
3.14:         end
3.15:     end
3.16: end
3.17: Result(k).Res
```

candidates and, if so, this step 6 is still running. The script `BSGIND.m`, that checks if a candidate is SG, calls to two other scripts, `PARSGIND.m` and `CONSGIND.m`, as seen in Algorithm 2.

The script `PARSGIND.m` calculates the parameters of Eq. 1 from arrays a_{PI}, a_{PS}, a_{LG}. It counts the internal and external pairs and input movements between the candidate link i and the external link j. With these values, the script `CONSGIND.m` evaluates if the candidate is SG (Eq. 1). If so, it includes the candidate as SG in $v_{SG}(i) = 1$, writes the results matrix $Result(k).Res$, and updates the counter of the order of formation of SG.

Step 7: Search for extended SG

The last step of the root script `STRUCTURAL_ANALYSIS.m` is to form new SG by progressively increasing the length of the kinematic chains. The script `BSGGR.m` performs this task, calling two other functions, `BKCH.m` and `PSGGR.m`, with a similar structure than those used for single groups, but considerably more complex. Its operation is shown in Algorithm 3.

The script `BKCH.m` searches, for each candidate link, kinematic chains of length $lengthSG$. Then, it calls routine `PSGGR.m` to verify if any of them satisfies Eq. 1. Only the kinematic chains that do not contain any link that already pertains to any SG are verified.

The script `PSGGR.m` checks if any of these chains is SG. To do so, it identifies the type of DOF that exists between each kinematic pair and calculates the group parameters, as shown for single groups. Then, Eq. 1 is checked. If it is verified, the variables v_{SG} and $Result(k).Res$ are modified and the DOF assigned to the links of the external pairs. After this, the search for single SG begins again, always looking for SG of the lowest length. If the chain does not satisfy Eq. 1, another chain stored in $SGgroup$ array is then verified. If no chain of array $SGgroup$ is SG, the control returns to the script `BSGGR.m` to search for SG with one additional link.

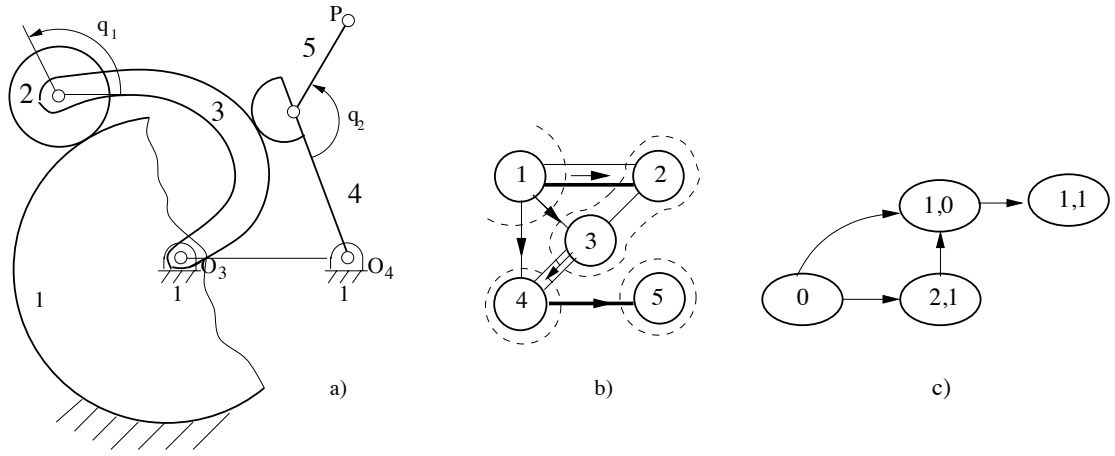

Figure 4. a) System with epicyclic gear train, cam-follower and open kinematic chain. b) Structural graph showing its division into SG and how the DOF have been assigned to the links. c) Structural diagram showing the kinematic structure of the mechanism.

4 RESULTS

The computational method presented in the previous section and all the algorithms described in detail have been implemented in the MATLAB programming environment. This section provides the results of its application to a mechanism composed by lower and higher kinematic pairs. Figure 4.a shows the kinematic graph of the mechanism. Bodies 3 and 4 are articulated to the frame 1 through lower rotation kinematic pairs. The wheel 1 is fixed. Wheel 2 and arm 3 form an epicyclic gear train. Arm 3 forms a cam-follower pair with link 4. Link 5 is articulated to link 4. The set of coordinates $[q_1, q_2]$ are considered as independent DOF and they represent the rotation of the wheel 2 with respect to the fixed frame and the rotation of link 5 relative to link 4. Equation 2 shows the adjacency matrix $Result(1).M$ provided by the user for computational analysis.

Applying the graph-analytical method described in Section 2, we obtain the structural graph (Fig. 4.b) and the structural diagram (Fig. 4.c) of the mechanism. It can be seen that this mechanism is composed by three structural groups: SG-I:$\{2, 3\}$, SG-II:$\{4\}$, SG-III:$\{5\}$. Applying the computational method presented in this work we obtain the same kinematic structure, represented by its matrix of results $Result(1).Res$ (Eq. 2).

$$Result(1).M = \begin{pmatrix} 0 & 4 & 1 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \qquad Result(1).Res = \begin{pmatrix} 0 & 1 & 1 & 2 & 3 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \tag{2}$$

The first row in matrix $Result(1).Res$ shows the SG to which each link (column) belongs to in the order in which the SG have been obtained. The value 1 in elements $(1, 2)$ and $(1, 3)$ means that links 2 and 3 form SG-I. Value 2 in element $(1, 4)$ means that link 4 forms the second SG, and so on.

If we consider than the input movement q_1 is applied to the kinematic pair $\{1,\ 2\}$ instead of being applied to links $\{1,\ 3\}$, then the first structural transformation is obtained. Applying the routine `AMPLIACION.M` the corresponding adjacency matrix is obtained. This adjacency matrix, $Result(2).M$, is shown in Eq. 3. The structural graph and the structural diagram corresponding to this structural transformation are shown in Fig. 5.a and Fig. 5.b.

It can be seen that the kinematic structure of this structural transformation obtained by both graph-analytical

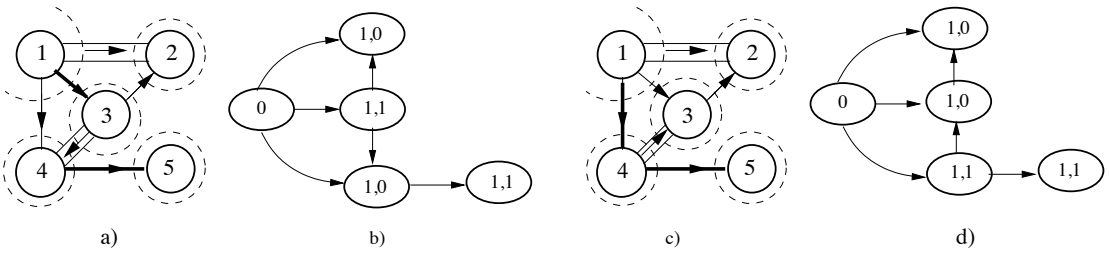

Figure 5. a) and b) Structural graph and diagram of the mechanism with adjacency matrix $Result(2).M$; c) and d) Structural graph and diagram of the mechanism with adjacency matrix $Result(3).M$.

(Fig. 5.b) and computational methods ($Result(2).Res$, in Eq. 3) are exactly the same. This kinematic structure is easier to analyse than the previous one, because as it has a higher number of SG which are necessarily simpler.

$$Result(2).M = \begin{pmatrix} 0 & 2 & 3 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 3 & 1 & 0 & 2 & 0 \\ 1 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \quad Result(2).Res = \begin{pmatrix} 0 & 4 & 1 & 2 & 3 \\ 1 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \tag{3}$$

Finally, another structural transformation can be obtained, considering the input q_1 as the absolute rotation of body 4 with respect to the chassis. The new structural graph and the structural diagram obtained by graph-analytic methods are shown in Fig. 5.c and Fig. 5.d. The adjacency matrix $Result(3).M$ of this structural transformation and the computational result $Result(3).Res$ are shown in Eq. 4. Once again, the kinematic structure obtained by both methods is the same. In addition, this new kinematic structure is very similar to the previous structural transformation and does not introduce any improvement on it.

$$Result(3).M = \begin{pmatrix} 0 & 2 & 1 & 3 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 3 & 0 & 2 & 0 & 3 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \quad Result(3).Res = \begin{pmatrix} 0 & 4 & 3 & 1 & 2 \\ 1 & 0 & 3 & 0 & 0 \\ 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \tag{4}$$

5 CONCLUSIONS

The computational method presented in this work allows to obtain the kinematic structure of planar multibody systems with lower and higher pairs, closed and open kinematic chains, and with one or several degrees of freedom. The only information required from the user is the adjacency matrix of the system to be analized. This matrix is easy to obtain: only the recognition of the types of kinematic pairs and the input movements defined on their links are required. From the adjacency matrix, the method obtains new structural transformations, and for each of them it also determines its kinematic structure automatically. The following developments of this study will identify the variables that each structural group introduces into the system and the constraint equations which relate them, in order to completely automate the kinematic and dynamic analysis of multibody systems from its kinematic structure. This paper introduces the first stage of a new completely autonomous method of kinematic and dynamic analysis based on the kinematic structure of mechanisms. This first stage could provide the following advantages over other methods currently in use:

- Automatization of the initial phase of topological and structural analysis.

- Allows the selection of the structural transformation that best suit the problem at hand (parallelization, avoidance of non-programmed groups or groups suffering from singular positions, etc.)
- Solution of the structural synthesis and the kinematic and dynamic analyses in one single application.
- Automatic selection of the constraint equations for each SG based on the mechanism adjacency matrix, thus limiting the options of making mistakes in this task.
- Open selection of coordinates (absolute, relative, natural o mixed) that best suit the problem to be solved and tthe model definition. The application would make use of the constraint equations programmed for each SG according to the type of selected coordinates.

Although the proposed method is limited to planar systems, it can be extended to spatial mechanisms and can also address other features, still to be explored in structural analysis, like pure rolling, multi-revolute joints and additional degrees of freedom (those found in systems with higher mobility than that corresponding to their topology, due to special geometric configurations).

REFERENCES

[1] Buskiewicz, J.:A method of optimization of solving a kinematic problem with the use of structural analysis algorithm (SAM) Mechanism and Machine Theory, Vol 41 , No 7, pp. 823-837, 2006.

[2] Huafeng, D.; Zhen H.; Dejun, M.:Computer-aided structure decomposition theory of kinematic chains and its applications Mechanism and Machine Theory, Vol 41 , No 43, pp. 1596-1609, 2008.

[3] Kolovsky, MZ.; Evgrafov, AN.; Semenov, Yu.A.; Slousch, A.V.:Advanced theory of mechanism and machines. Springer, Dordrecht, 2000.

[4] Lung-Wen, T.:Enumeration of Kinematic Structures According to Function CRC Press, 2001

[5] Mitsi S.; Bouzakis K.; Mansour G.; Popescu I.:Position analysis in polynomial form of planar mechanisms with Assur groups of class 3 including revolute and prismatic joints Mechanism and Machine Theory, Vol 38, No , pp. 1325-1344 , 2003.

[6] Mruthyunjaya, TS.:Kinematic structure of mechanisms revisited Mechanism and Machine Theory, Vol 38, No 4, pp. 279-320, 2003.

[7] Popescu, J.;Marghitu, D.:Structural design of planar mechanisms with dyads Multibody System Dynamics, Vol , No 19, pp. 407-425, 2008.

[8] Shai, O.: Topological synthesis of all 2D mechanisms through Assur Graphs. in: Proceedeings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Informatin in Engineering Conference, Montreal, Quebec, Canada, 2010, pp. 1–11.

[9] Shi, P; McPhee, J.:Symbolic programming of a graph theoretic approach to flexible multibody dynamics Mech. Struct. Mach. Vol 30, No 1, pp. 123-154, 2002.