Mechanism and Machine Theory 176 (2022) 104985

Contents lists available at ScienceDirect

Mechanism and Machine Theory

journal homepage: www.elsevier.com/locate/mechmt

Computational structural analysis of spatial multibody systems based on mobility criteria

A. Celdran [a], M. Saura [a,*], D. Dopico [b]

[a] Dept of Mechanical Engineering. Universidad Politecnica de Cartagena. Doctor Fleming, s/n, 30202 Cartagena, Spain
[b] Mechanical Engineering Laboratory. University of A Coruña. Mendizabal, s/n, 15403 Ferrol, Spain

ARTICLE INFO

Keywords:
Computational
Structural analysis
Multibody systems
Kinematic substructuring

ABSTRACT

Computational efficiency is a critical aspect of multibody dynamics. Recent developments in linear algebra libraries and hardware architectures allow the multibody community to improve the performance of their kinematic and dynamic formulations. Kinematic substructuring of multibody systems may take great advantage of these resources by dividing the multibody system into modules that can be solved more efficiently than the whole set of equations of the multibody system at once. This work discusses some advantages of the kinematic substructuring based on mobility criteria over methods based on linear graph theory. It also introduces detailed information on the algorithms we have developed for the computational structural analysis of planar and spatial multibody systems. Structural transformations easily generate different kinematic substructures, ordered and classified under various criteria depending on the analyst's needs, such as robustness and efficiency. A graphical user interface may guide the analyst while defining the topology of the multibody system and show detailed information about the obtained results. The algorithms are explained through an example, and the advantages of this method are reinforced as applied to several spatial multibody systems.

1. Introduction

Multibody systems are composed of rigid or flexible bodies connected by different types of mechanical joints, allowing certain relative motions, rotations or translations, between the joined bodies. They are present in a broad range of applications in robotics, biomechanics, industrial machinery, or the automotive and aerospace industry. Structural, kinematic and dynamic analysis of multibody systems are essential tools needed for many tasks, as their design, simulation and control.

These tasks demand improved resources such as theoretical formulations, hardware configuration and software implementation, which have to be efficiently combined. To achieve this efficient combination of resources, up-to-date developments in hardware, as parallel computing capabilities of new processors should be included in software development (i.e. linear algebra solvers with implicit parallelism) and these new capabilities, should be exploited in the theoretical formulations and practical implementation of the analysis of multibody systems (MBS).

Concerning the theoretical formulations, both global and topological approaches are traditionally employed to model and solve the kinematics and dynamics of MBS. Global approaches use a set of coordinates (reference point or natural) to model the multibody system without considering its topology (open or closed loops). These coordinates are related through a set of constraint equations which are simpler and more prone to be automatically obtained than in the topological formulations with joint coordinates. These

* Corresponding author.
E-mail address: msaura.sanchez@upct.es (M. Saura).

https://doi.org/10.1016/j.mechmachtheory.2022.104985
Received 28 March 2022; Received in revised form 7 June 2022; Accepted 7 June 2022
Available online 22 June 2022

advantages make global formulations good candidates for the automatic modelling of multibody systems [1–4]. The approaches in natural coordinates usually lead to smaller size problems than reference point approaches, especially in spatial MBS [5].

The main drawback of the global approaches is that, as the complexity of the problem increases, the number of coordinates, constraint equations and the size of any other matrix involved in their analysis increases accordingly; solving these systems of equations efficiently is a permanent field of research.

On the contrary, in the topological approaches, kinematic substructuring techniques exploit the topology of a multibody system to obtain its kinematic structure; that is, an ordered set of kinematic chains (known as subsystems or modules), improving their efficiency in three ways: first, the obtained subsystems can be modelled with a reduced number of coordinates; second, if the reduced-size subsystems are kinematically determined, as in our approach, the multibody system may be analysed, at each time step, by solving the subsystems in its kinematic structure, rather than solving the set of dependent coordinates of the complete multibody system [6]; and third, this efficiency might be further improved using explicit parallelism techniques as OpenMP or MPI to solve these modules in distributed environments (parallel) [7].

There are several interesting questions related to the kinematic substructuring phase in topological approaches, as: how many kinematic structures can be obtained for a given multibody system and how to obtain them; which one is more appropriate for automatic modelling (simplicity of the constraint equations) or might be more efficiently solved (hierarchical dependency of the subsystems), how the kinematic structure is affected if the topology of the multibody system changes along the simulation time, or how to avoid subsystems singular configurations during analysis. The theory of structural analysis, a field of mechanisms and machines theory, offers different methods for kinematic substructuring of multibody system and answers to some of these questions which will be addressed here, but others have to be further investigated.

Different algorithms which exploits the topology of multibody systems are used for kinematic substructuring; the most frequently used are linear graph algorithms (LGA) *y* and mobility-based algorithms (MBA).

In the linear graph theory algorithms, the kinematic structure of the MBS is formed by a set of independent open and closed loops (subsystems) which, in some cases, can be hierarchically organised, and then, more efficiently solved [8–13]. On the contrary, in the mobility-based algorithms, the kinematic structure of the MBS is formed by an ordered set of kinematically determined chains of the shortest length, known as modules or structural groups, which can be solved hierarchically [14–21]. Algorithms based on linear graph theory or mobility criteria can be applied by visual inspection in basic multibody systems to obtain their kinematic structure. Still, computational implementation is needed when dealing with more complex ones.

An interesting review about how these two techniques have been applied in the last decades to structural analysis and structural synthesis problems can be found in [22]. Structural analysis studies the kinematic structure of a given MBS, whereas structural synthesis aims to design multibody systems under specific topological criteria; a typical example of the latter would be to generate N_{mech} different mechanisms of specific mobility f, composed by N of bodies and which must be connected by a particular type of joints (i.e. revolute joints). Structural synthesis is frequently based on computational algorithms since the number of generated combinations is relatively high (some authors agree that for $N = 10$, $f = 1$ connected only by revolute joints, up to $N_{\text{mech}} = 1834$ different mechanisms can be generated). To identify the various linkages that solve the synthesis problem, first, all the possible combinations with the given topology must be generated. Then, kinematic substructuring techniques are applied to discard both rigid sub-chains and isomorphisms amongst them.

However, the substructuring techniques reviewed in [22] are not aimed at the kinematic or dynamic analysis of multibody systems. The following section covers this gap introducing how kinematic substructuring has been used in the last decades to analyse multibody systems under topological approaches and how these techniques would perform the kinematic substructuring of a given multibody system.

1.1. Linear graph algorithms for kinematic substructuring

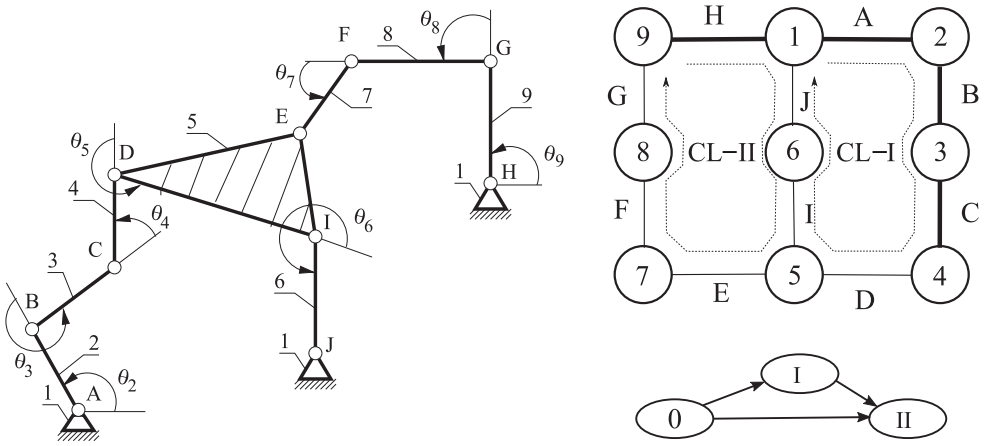
Kinematic substructuring exploits the topology of a multibody system to divide it into a set of independent open and closed-loops. In simple multibody systems, as the one shown in Fig. 1(a), a topological graph as in Fig. 1(b) (up) helps in identifying the closed-loops by visual inspection. The graph is constructed by vertices representing bodies and, if two bodies form a kinematic pair, they are joined by as many lines as the degree of the kinematic pair (DOF allowed by their mechanical joint). Moreover, these DOFs are represented by bold lines if they are driven and by thin lines if they are not (unknown DOFs). The lines (joints) in the graph may be named with the same letter assigned to the joints in the mechanism. As shown in Fig. 1(b), we are considering that the four driven DOFs in this linkage are: $\theta_2, \theta_3, \theta_4, \theta_9$, and then, their values are known at any instant of time.

By visual inspection, a linear graph algorithm would detect three closed loops for this multibody system; for clarity, only two of them have been represented in Fig. 1(b)(up). In the first closed-loop (CL-I), bodies: $\{1, 2, 3, 4, 5, 6\}$ and coordinates: $[\theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ are involved; the second closed-loop (CL-II) is formed by bodies: $\{1, 6, 5, 7, 8, 9\}$ and coordinates: $[\theta_6, \theta_5, \theta_7, \theta_8, \theta_9]$, and the third one (CL-III), by bodies: $\{1, 2, 3, 4, 5, 7, 8, 9\}$ and coordinates: $[\theta_2, \theta_3, \theta_4, \theta_5, \theta_7, \theta_8, \theta_9]$. From these three closed loops, the linear graph algorithm must calculate how many are independent (Eq. (1)) and select which ones offer the most efficient kinematic structure.

$$C = P - N_m \tag{1}$$

where, C is the number of independent closed-loops, P is the number of kinematic pairs and N_m is the number of movable bodies.

In the example, there are only three possible kinematic structures obtained by combining two independent closed-loops: {CL-I, CL-II}; {CL-I, CL-III} or {CL-II, CL-III}. In order to solve the whole linkage efficiently, the kinematic structure with a lower number of coordinates must be selected: {CL-I, CL-II}. Considering that each closed-loop in this linkage imposes two constraint equations, the



(a) Nine-bar linkage with four degrees of freedom with closed loops.

(b) The structural graph associated to this linkage resumes its topology (up). The kinematic graph shows two kinematic-dependent closed loops (down).

Fig. 1. Nine-bar mechanism with the corresponding topological graph and kinematic graph.

ones from CL-I solves the unknown coordinates: $[\theta_5, \theta_6]$ by relating them to the driven DOFs: $[\theta_2, \theta_3, \theta_4]$. Then, two new constraint equations from CL-II relate the unknown coordinates: $[\theta_7, \theta_8]$ to the: $[\theta_5, \theta_6, \theta_9]$ known ones. This kinematic structure is represented by a kinematic graph in Fig. 1(b)(down), and it is similar to the concept of kinematic network in [11]. The first node with a zero in it represents the frame and the other nodes represents the number of the module (closed-loop). Arrows indicate the dependency between modules, giving the sequence in which they must be solved.

The second kinematic structure {CL-I, CL-III} would also offer a hierarchical solution but with a large number of coordinates in CL-III, and the third kinematic structure is less efficient, as it would require to solve the four constraint equations together to determine the values of the four remaining unknown coordinates.

Instead of visual inspection, the adjacency matrix is widely used in the literature for general-purpose linear graph algorithms. In [8], the closed-loops are automatically obtained for complex multibody systems keeping their number of joints to a minimum and the kinematics and dynamics of a hybrid open and closed-loops robotic arm are solved using relative coordinates and homogeneous transformation matrices for the kinematic loops.

The same criteria of keeping the set of loops with the minimum number of joint coordinates is used in the linear graph algorithm by Kecskemethy [9] for the automatic generation of symbolic equations for multiple loop mechanisms whose kinematics is solved in closed-form. They also discuss how to select a suitable set of multibody loops to create the corresponding set of kinematical transformers and how to couple equations between loops employing a solution flow (hierarchy) which allows the solution of the complete multibody system. Later, in 1997, Kecskemethy [11] extends the closed-form solution of the kinematical transformers introduced in [9] for the dynamic analysis of multi-loop mechanisms. He applies this method for the solution of a planar and a spatial mechanism and obtains speed-ups closed to 2.5 in kinematics and 8 in dynamics compared to iterative methods.

Applying a linear graph algorithm with the criteria of a minimum number of joints in each loop would offer the same kinematic graph as in Fig. 1(b)(down).

More recently, in 2015, Wehage [13] uses kinematic substructuring based on graph theory to reorganise the elements of the Jacobian matrix of the MBS into uncoupled sub-matrices according to the obtained subsystems. Then, he uses the coordinate partitioning method based on the Gaussian Elimination with Complete Pivoting (SGCP) to find the optimal set of independent coordinates, which avoid the multibody system to enter singular positions during dynamic analysis. His algorithm applies to multibody systems with closed-loops; however, their definition of a subsystem allows, under certain conditions, to integrate new closed-loops into an existing subsystem, increasing its complexity, which is opposite to the idea of the modular approach based on structural groups or the idea of a minimum number of joint coordinates [11].

In Wehage [13], a topological graph may also be constructed, as shown in Fig. 2(up) for the nine-bar linkage. In this graph, virtual bodies 10 and 11 are included indicating with dashed lines (defined as chords) the cut joints in order to leave the multibody system in a tree-like structure. Defining an array of parents, a list of loop_terminators and a list of parent_joints as in Table 1, the LGA offers a kinematic substructuring formed by two substructures in Table 2.

Note that, for clarity, the body numbers shown in the tables coincides with the ones used in the graph (Fig. 2(up)), but the algorithm proposed by [13] the numbering starts with 0 assigned to the frame. The obtained kinematic substructure coincides with the kinematic structure {CL-I, CL-III}, meaning that we have not obtained the shortest-length loops. The kinematic graph shown in Fig. 2(down) reveals that the order and dependency of the closed loops is not the most efficient, as discussed above.

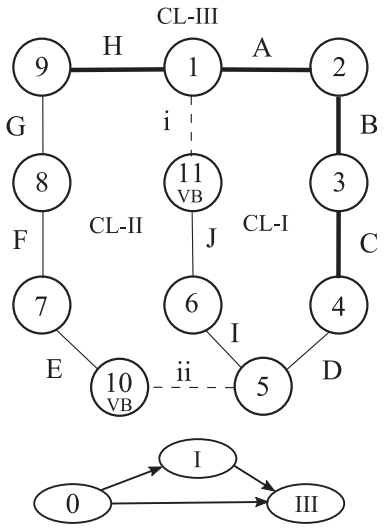


Fig. 2. A topological graph helps identify the information the algorithm needs (up). The kinematic graph (down) represents the kinematic structure obtained for the nine-bar linkage using the LGA from [13].

Table 1
Parameters the user must provide for computational kinematic substructuring using the LGA in [13].

List		Items
P	=	[0, 1, 2, 3, 4, 5, 8, 9, 1, 7, 6]
loop_terminators	=	[[5, 10], [1, 11]]
parent_joints	=	[None, 'A', 'B', 'C', 'D', 'I', 'F', 'G', 'H', 'E', 'J']

Table 2
Substructures for the nine-bar mechanism using computational LGA in [13].

Substructure id	Loops	Loop members	Loop correspondence
1	ii	1, 2, 7, 3, 8, 4, 5, 9, 10	CL-III
2	i	1, 2, 3, 4, 5, 6, 11	CL-I

Different kinematic substructures might be obtained by visual inspection in simple multibody systems, but computational algorithms are needed when dealing with complex ones. However, as shown in the nine-bar example, the analyst must anticipate some work to provide the algorithm with the needed information.

1.2. Mobility based algorithms for kinematic substructuring

The ordered set of subsystems obtained by kinematic substructuring using mobility-based algorithms are known as modules or structural groups (SGs); moreover, kinematic and dynamic formulations based on modules or structural groups are known as modular approaches in the literature.

Leonid Assur introduced the first method for kinematic substructuring based on mobility criteria in the first half of the XIX century [23,24]. Two types of structural groups were defined for planar articulated linkages: primary elements, like one driving crank or slider that moves with respect to the frame (as bodies 2 and 9 in Fig. 4(a)), and Assur groups of class k defined as kinematic chains with $N_m = 2k$ mobile bodies and null mobility. An Assur group of class one (k = I) is named a dyad of type *RRR, RPR, RRP, PPR, PRP, PPP* depending on how their two bodies are connected internal, and externally to other bodies of the multibody system with revolute (R) or prismatic (P) joints. In Fig. 4(a), bodies $\{5,6\}$ form a typical *RRR* dyad.

Exploiting the modularity of mobility based algorithms have been a permanent topic of interest in the multibody community but limited to a reduced set of primary elements and dyads for planar articulated multibody systems. For example, in 1984, Kinzel [14] performs the kinematic analysis of 2D articulated linkages using closed-form solutions for a set of six dyads and one primary element. In the same year, Smith [15] introduces the program SIMPLA for the kinematic analysis of planar linkages composed by a combination of nine different dyads. Hammar [16], in 1987, discusses the advantages of modular programs as the Linkage Analysis Program (LAP) to specific-purpose programs and general-purpose software under licence. The author introduces the concept of subroutine package to enclose the kinematic solution of particular dyads, which may be reused as many times as this module appears in the kinematic structure. In these works, the kinematic substructuring has to be obtained mentally by the analyst.

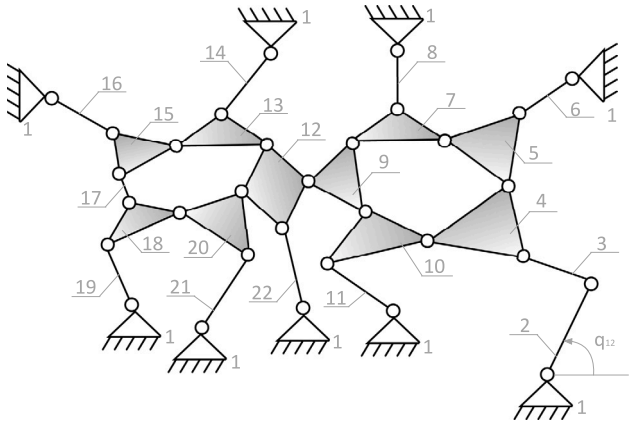


Fig. 3. A 22 bodies, one DOF planar articulated multibody system studied in [19] to convert non-dyad into dyad linkages.

In 1993, [17] Fanghella describes a mobility-based algorithm that detects modules as single-loop determined chains of the shortest length (dyads) which can be solved in closed-form and hierarchically. In 1996, Hansen [18] introduces a modular approach for 2D MBS kinematics and dynamics, combining neutral modules (five different dyads and two Assur groups of second class $k = \text{II}$) and two expansion modules (primary elements). Hansen outlines the advantages of a modular approach in structural synthesis since different mechanisms with a given DOF can be easily generated by combining these modules. His algorithm performs kinematic substructuring by iteratively searching for expansion and neutral modules of a prescribed kind. In the authors' opinion, an extension of the method to spatial modules is not recommended because of practical implementation problems and lower efficiency than a global approach. However, we have not encountered any of these problems in our studies.

Not any planar articulated multibody system may be divided into primary elements and a combination of structural groups of class $k = \text{I}$ or $k = \text{II}$, and the kinematic substructuring with the cited mobility methods would fail in these cases, as in the mechanism shown in Fig. 3 from Ming [19], in 1997. In his work, the author automatically decompose a planar MBS into the corresponding set of SGs from a database similar to [14]. If a complete decomposition is possible, the linkage is named a dyad linkage, and the solution of the MBS is obtained by solving their dyads hierarchically. Moreover, an optimisation method is introduced to transform non-dyad linkages into dyad linkages. The main limitations of this method are the small number of dyads in the database, the planar case, and no higher kinematic pairs are allowed.

Another computational MBA for structural analysis was proposed by Buskiewicz [25] in 2006, valid for one-DOF planar mechanisms with lower kinematic pairs. Once the primary elements have been removed from the MBS, the user must introduce a large amount of information: the number of loops, number of joints, number of moving links, links in the loops and joints in the loops, to help the algorithm to divide it into Assur groups of any class, instead of searching for kinematic chains from a predefined library. The MBA algorithm, called SAM (Structural Analysis of planar Mechanism), uses the obtained kinematic structure to reorganise the global kinematic equations of the system through the Jacobian matrix and, theoretically, solve them more efficiently.

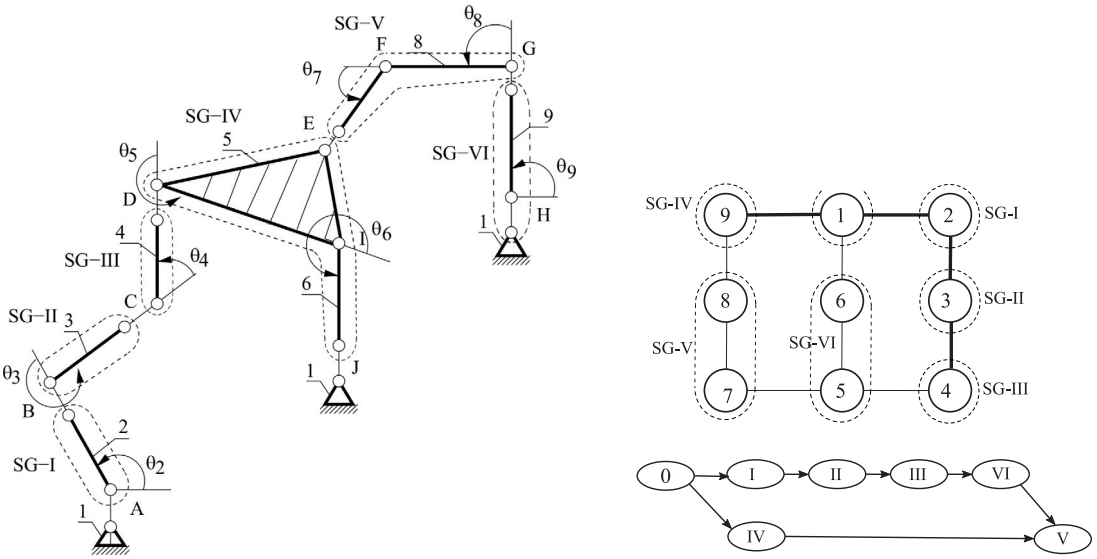
At the same time, Varbanov [26] introduces the software (S&A expert system) for the solution of the kinematic analysis and synthesis of one-DOF planar linkages formed by a combination of one primary element and up to five different types of dyads with lower kinematic pairs. Although kinematic substructuring relies on the analyst, the work illustrates that the same modular approach can be used for analysis and synthesis.

Graph-analytical kinematic substructuring of spatial mechanisms was proposed for the first time by Zeng and Yuefa, in 2012, for the division of serial–parallel hybrid mechanisms used in the robotic field [27]. They introduce a *multiple spatial loops division* based on the *card of displacement of subsets of chains* ($CCDS$). They show some analytic examples and find it interesting to develop a computational method to automate this substructuring algorithm.

In 2014, we introduce a computational MBA for the kinematic substructuring of planar MBS [21]. The main difference regarding previous works is that, instead of dividing the multibody system into a set of primary elements and a set of well-recognised Assur groups of zero mobility, the concept of SG was extended to any kinematically determined chain of the smallest size possible, including both lower and higher kinematic pairs and any combination of revolute, prismatic, or direct contact kinematic pairs (as cams and gears). The user must provide the adjacency matrix and the distribution of the independent DOFs. However, the spatial case was not considered. The methods developed were based on the particular classification among lower or higher kinematic pairs, which affected most of the code, considerably reduced the algorithm's scope and the information generated during the solution.

Most of the reviewed MBA leave kinematic substructuring to the analyst. Others only deal with one-DOF multibody systems, so they cannot obtain the kinematic structure of the nine-bar mechanism in our example. Applying the MBA in [21], the kinematic structure in Fig. 4(a) is obtained.

Contrary to the two independent closed-loops offered by linear graph algorithms, the applied MBA would divide the nine-bar linkage with the selected DOFs: $[\theta_2, \theta_3, \theta_4, \theta_9]$ into the six modules shown in Fig. 4(a). The kinematic graph in Fig. 4(b)(down) defines the hierarchical order in which these modules might be solved in a kinematic or dynamic analysis.



(a) Kinematic structure formed by six structural groups.

(b) Structural graph as defined in [20] (up) and Kinematic diagram (down).

Fig. 4. Kinematic substructuring of a nine-bar linkage using a MBA capable to deal with several DOFs [21].

Compared to Fig. 1(a), the obtained kinematic structure is composed of a higher number of basic subsystems of the same type: {SG-I, SG-II, SG-III, SG-VI} are modules formed by one body with one driving input motion and two dependent coordinates (the Cartesian coordinates of the unknown joints: B, C, D and G respectively), and {SG-IV, SG-V} are dyads with none input motions and two dependent coordinates (the Cartesian coordinates of the inner joint: I and F respectively, or the unknown orientations of the bodies in the corresponding dyad: $[\theta_5, \theta_6]$ and $[\theta_7, \theta_8]$).

These subsystems are reduced in size and more straightforward to analyse than the ones obtained in Fig. 1(b). The kinematics of the whole multibody system can be solved hierarchically following the sequence shown in Fig. 4(b)(down). First, SG-I and SG-VI can be solved simultaneously, as they are not dependent; then, SG-II, SG-III, SG-IV structural groups must be solved sequentially. Finally, once the solution to SG-IV and SG-VI has been obtained, solving the last SG-V will end the analysis of the whole MBS. Moreover, each subsystem type might have its kinematics solved in a specific subroutine that can be optimised and reused to solve the multibody system more efficiently.

The hierarchical sequence described above for a multibody system's kinematic/dynamic solution may not be the most efficient. The optimal sequence for a given combination of software and hardware equipment may be determined only when the structural groups that define the kinematic structure of a multibody system have been obtained, which is the main purpose of this work.

To clarify these ideas, suppose we need to solve a multibody system (Stewart platform) whose kinematic structure is formed by one structural group (SG_T) and six kinematically independent structural groups (SG_MB_1, ..., SG_MB_6). The kinematic diagram of such a system is shown in Fig. 5 and will be analysed as a case study (Fig. 17.b). Each module has its solution in a specific routine: SG_KINEM_6C for SG_T and the same routine SG_KINEM_REC for all SG_MB_X because they share the same topology.

In a global approach, or an LGA that solves all the model constraint equations at once, the kinematic structure consists on one module (MBS_STEWART) and only one possible route, as shown in Fig. 6.

On the contrary, if the kinematic structure in Fig. 5 is considered, we may use our PARCSIM (PARallel Computing SIMulator) software [7] to build the tree of kinematic routes (Fig. 7). Note that the upper route (57) and the lower route (23) correspond to the full sequential and parallel solutions of the structural groups, while the other routes combine parallel and sequential solutions of structural groups. The PARCSIM simulator studies all the routes to determine the most efficient for a given set of algorithmic parameters [7].

Other authors have also applied the concept of modular approach using primary elements and dyads in the dynamics of rigid bodies. Open-loops [28], one single closed loop [29,30], basic MBS with multiple-loops [29,31], one-loop flexible MBS [32,33]. The dynamics of a (3-RRR) planar parallel manipulator with three identical modules (the corresponding legs) and the terminal body being the fourth module [29]. Also, in [34,35] the authors use kinematic substructuring in the dynamic analysis of vehicles. Although no references are made in these works about how to obtain the kinematic substructures of an MBS (except in [33]), it is interesting to note that not only in kinematic analysis but also in dynamics of MBS, kinematic substructuring can be used to construct the equations of motion of the MBS.

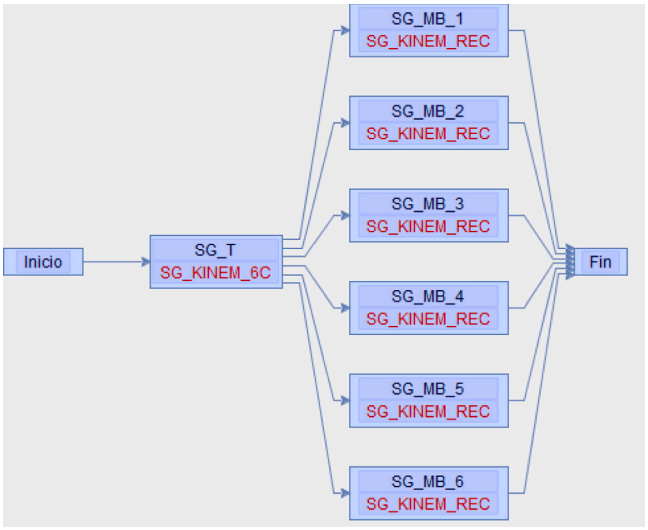


Fig. 5. Structural diagram of a multibody system (Stewart platform) with one structural group followed by six kinematically independent structural groups.

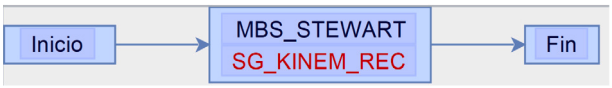


Fig. 6. Kinematic structure of a multibody model (Stewart platform) defined by a global approach or an LGA approach without considering its kinematic structure.

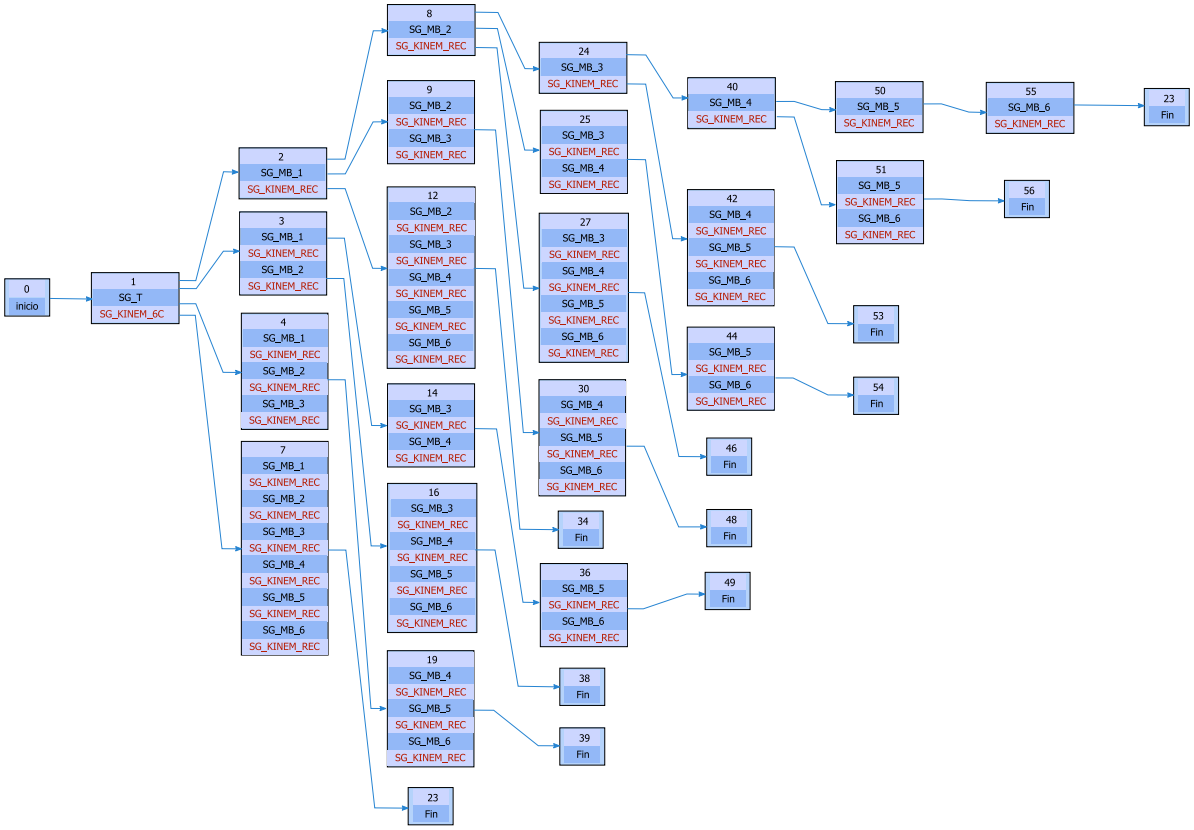


Fig. 7. Tree of kinematic routes showing parallelism capabilities in the analysis of multibody systems, when its kinematic structure based on structural groups have been defined.

1.3. Advantages and drawbacks between linear graph and mobility-based algorithms

In summary, from the literature review, the studied example and our own experience, we introduce some advantages and drawbacks of both approaches to highlight the need to extend the mobility-based algorithm to a computational solution for 3D multibody systems, which is the main contribution of this work:

(a) Both linear graph theory and mobility-based algorithms have evolved in parallel as valuable techniques that exploit the kinematic structure of multibody systems to improve the efficiency of the kinematic and dynamic formulations used in the design, simulation and control of multibody systems.
(b) LGA look for independent open and closed loops regardless of the multibody system is planar or spatial, the type of mechanical joint between bodies and which DOF are driven. Therefore, LGA are easier to develop and have been applied to complex 2D and 3D multibody systems. We find it easy, in LGA, to identify modules but challenging to generate reusable closed-form solutions to those modules.
(c) On the contrary, MBA need the parameters cited in (b) making kinematic substructuring more complex, and applied in the literature only to simple 2D linkages formed by a limited type of Assur groups and primary elements, mainly for one-DOF multibody systems. Therefore, in MBA, we find it easy to obtain and reuse efficient closed-form solutions for these modules but challenging to obtain kinematic substructuring in complex planar multibody systems and nonexistent algorithms for spatial ones.
(d) LGA offer a reduced number of kinematic structures for the modelling and kinematic or dynamic analysis of a given multibody system. Each kinematic structure is formed by all the open loops and a selection of independent closed-loops (Eq. (1)). However, the more kinematic structures available, the best, allowing the possibility of selecting the one that best fits the analyst needs, as cited in (e).
(e) Each kinematic structure offered by LGA comprises a reduced number of modules, increasing their complexity which might further increase depending on the cut-joints selected by the user, and making closed-form solutions impossible or less efficient. On the contrary, MBA searches for minimum size modules, as dyads, with very efficient and well-known closed-form solutions. Moreover, the obtained modules in MBA can be solved using a combination of different coordinates (joint, natural or reference point), solvers (dense or sparse) and distributed strategies that make their solution even more efficient.
(f) The proximity to a singular position of a given multibody system might be more efficiently analysed by checking the rank of the Jacobian matrix of the structural groups that composes the kinematic structure of the MBS rather than the rank of the Jacobian matrix of the whole MBS.
(g) In the design (synthesis) of multibody systems, LGA are used to discard rigid kinematic chains and isomorphisms from many possible designs. However, MBA may be more efficient by simply adding modules of different types according to the analyst's needs, like the number of bodies or the type of joints to consider.
(h) In MBA, there exist techniques that convert planar non-dyad multibody systems into dyad ones, making their solution more efficient [19].
(i) In MBA, there are techniques for the kinematics and dynamics of MBS whose kinematic structure has been obtained using a set of coordinates different from those driven during the simulation time [14]. These techniques allow the analyst to select the most efficient kinematic structure independently of the actual DOFs of the multibody system. This versatility is not available in LGA, as mentioned in (d).
(j) A valid set of independent coordinates must be chosen among all the coordinates to correctly define a kinematic or a dynamic analysis with both topological methods. In this sense, a simple identification of independent open and closed-loops with LGA does not assure a proper selection of independent coordinates. On the contrary, MBA guarantees that the problem is well defined. Otherwise, kinematic substructuring would fail.
(k) Comparing the information that the analyst must provide in linear graph algorithms with respect to mobility-based algorithms is complicated, as it depends on the author's method and programming abilities. In two references where the authors detailed this information, their algorithms needed more information than our MBA. For example, in [25] their MBA needs the number of loops, number of joints, number of moving links, links in the loops and joints in the loops. Also, in [13], their LGA needs: to identify all the bodies, joints and loops, cut the loops and specify loop terminators (as many as independent loops) and, finally, identify the parent of each body and the parent joint of each body. On the other hand, in [21] needs only the adjacency and the independent DOFs distribution matrices.

The considerations above show that, compared to linear graph algorithms, the modular approach based on mobility criteria has many advantages regarding modelling and closed-form solution of computational kinematics, dynamics and design of multibody systems. Furthermore, regarding computational cost, MBA have shown clear advantages with respect to global formulations [6,7]. However, the reviewed mobility-based algorithms completely lack generality in kinematic substructuring of complex 2D multibody systems and have not been developed for the spatial case.

1.4. Scope and contributions

The present work introduces a mobility-based algorithm for the computational structural analysis of spatial multibody systems. An in-depth literature survey has been performed. The code and functions of the algorithms in [21] have been reviewed, modified

and updated to contemplate both planar and spatial cases in the same algorithm. It has been redefined with a modular architecture, dividing the code into five modules. Each one encloses a complete algorithm step resolution similar to the graph analytic method steps, opening the algorithm to future developments. The new code handles complex MBS systems on a user-oriented view. It is not limited to lower and higher pairs identification; a new concept of the extended adjacency matrix is introduced considering the type of joint at the kinematic pairs, including floating bodies. Structural transformations may consider any DOF in the multibody system as a potential input movement, increasing the number of available kinematic structures considerably.

We have compiled these modules into a library named **MBSSA3D**, which stands for MultiBody System Structural Analysis in 3D. An intuitive graphical user interface guides any user without topological analysis or programming skills to obtain the kinematic structure of 3D multibody systems. We classify certain exceptions to the Kutzbach–Gruebler mobility criterion and explain how MBSSA3D must deal with them in the singularities module; however, implementing this module is ongoing.

The original algorithm provided the kinematic structure in a matrix form and no additional information. Now, in the final module, a complete list of information is provided with two objectives: first, to classify the kinematic structures according to different criteria, allowing the analyst to select the best kinematic structure for the kinematic and dynamic analysis of the multibody system on a practical view. The second objective is ongoing and not implemented, should automatically generate a kinematic or dynamic model in a compatible format to be read from already developed libraries for kinematics and dynamics of multibody systems as MBSLIM [36] or MBDSBEG [6].

This work applies the MBSSA3D for the kinematic substructuring of several spatial multibody systems of different complexity, which satisfies the Kutzbach–Gruebler mobility criterion. We compare graph-analytical and computational LGA and MBA results.

1.5. Organisation of the paper

In Section 1 we have introduced a literature survey of kinematic substructuring methods. Using a case study, we have shown the main advantages of mobility-based algorithms to linear graph algorithms and the importance of developing general-purpose algorithms for spatial multibody systems. We have also introduced the main contributions and scope of the present work. Section 2 shows the basics of structural analysis theory based on mobility criteria, and Section 3 introduces how the algorithm works applied to a spatial crank–slider mechanism as a case study. Section 4 applies the MBSSA3D to different complex spatial multibody systems and Section 5 performs a comparative study on robustness and efficiency of MBSSA3D algorithm with respect the 2D version introduced in [21]. Finally, Section 6 gathers the main conclusions obtained from this work and introduces new developments to be accomplished in the future.

2. Structural analysis theory

2.1. Mobility and kinematically determined chains

The mobility L_c of an holonomous kinematic chain defines its number of degrees of freedom (DOF), which can be calculated as:

$$L_c = BN_m - \sum_{k=1}^{B-1} e_k P_k \tag{2}$$

where B is a constant: $B = 3$ or $B = 6$ in planar and spatial multibody systems, respectively. N_m defines the number of bodies in the kinematic chain, k stands for the grade of a kinematic pair (DOF allowed by the joint that connects two bodies), P_k is the number of kinematic pairs of grade k that the N_m bodies form between them (internal) or with other bodies out from the kinematic chain (external), and e_k is the DOF removed by a kinematic pair of grade k ($e_k = B - k$). There are topological and kinematic conditions that make equation Eq. (2) fail; in this work, only holonomic kinematic chains in which Eq. (2) holds will be considered.

Eq. (2) also calculates the mobility f of a MBS; in that case, N_m accounts for the number of movable bodies in the multibody system. The kinematics of such a multibody system, modelled with a set of n dependent coordinates $\mathbf{q} \in \mathbb{R}^n$ can only be solved if: on the one hand, f independent coordinates $\mathbf{q}^i \in \mathbb{R}^f$ are driven; that is, their values are *input motions*, known at any instant of the simulation time, and, on the other hand, if the $\mathbf{q} \in \mathbb{R}^n$ coordinates are related thought $n - f$ independent equations. Such a multibody system is known as kinematically determined.

In the example of Fig. 1(a): $B = 3$, $N_m = 8$, $P_1 = 10$, $P_2 = 0$ the mobility of the MBS is $f = 4$. Since its mobility coincides with the number of driven joint coordinates $\mathbf{q}^i = [\theta_2, \theta_3, \theta_4, \theta_9]$, this multibody system is kinematically determined and then, its kinematics can be solved at each time step.

The concept 'kinematically determined' applies to any kinematic chain with a number of driven DOFs n_c equal to its mobility (L_c in Eq. (2)): $n_c = L_c$. For the given kinematic chain, special attention requires the number of *active* kinematic pairs, P_k, which stands for the sum of their internal ones and, among the external ones, only those in which the external body is the frame or a body from any already obtained structural group.

From now on, the following notation will be used: KP(i-j) as a reference to a kinematic pair formed by bodies i and j, KC{i, j, ..., z}, to a kinematic chain formed by bodies i, j, ..., z and SG-N{i, j, ..., z}, to a numbered structural group where (N) is a roman number and i, j, ..., z, the bodies that form this structural group. Sometimes, like in Fig. 4(a), the simplified notation SG-N will also be used.

Dividing a MBS into an ordered set of kinematically determined chains is the main objective of the structural analysis theory and, as the result of such process, the kinematic structure of a MBS is obtained.

2.2. Kinematic structure of a multibody system

The theory of structural analysis defines a structural group as a kinematically determined chain ($n_c = L_c$). If a structural group (SG) has neither excessive constrains, nor additional DOF due to special geometric considerations, it is defined as a *normal* SG [20]. Furthermore, normal SG which cannot be split into other normal SG of smaller number of bodies are denominated *simple* SG. Then, the kinematic structure of an MBS is defined as an ordered set of simple SG in which a MBS can be divided; this division is not unique and depends on the topology of the MBS and the selected driven coordinates of the MBS.

Multibody systems can be both divided into or built from an ordered set of simple SG. By considering $e_k = B - k$ in Eq. (2) and the condition of simple SG ($n_c = L_c$), we obtain the generation principle of mechanisms Eq. (3), which is the necessary condition for any kinematic chain to be a simple SG.

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

where P is the sum of all active kinematic pairs formed by the N_m movable bodies of the kinematic chain, and n_c is the number of the input motions (out of the complete set of $\mathbf{q}^i$ driven coordinates in the multibody system) that are present in the kinematic chain and S_c is defined as the number of DOF allowed by the P kinematic pairs as shown in Eq. (4):

$$S_c = \sum_{k=1}^{B-1} k P_k \tag{4}$$

The kinematic structure of a MBS can be obtained utilising graph-analytical or computational methods [21]. In the graph-analytical method, Eq. (3) defines the generation principle of mechanisms and must be applied combined with a graph representing the topology of a multibody system. Topological graphs can be represented in a variety of forms (see [11,37,38]). In this work, the structural graph introduced in [21] is used. The structural graph is helpful for efficient structural analysis and a better understanding of the computational algorithm introduced here.

In [21], the graph-analytical method for structural analysis was applied to a planar four-bar linkage. In this work, the same four steps of this method are explained in detail while applied to obtain the kinematic structure of the spatial four-bar linkage shown in Fig. 8(a). This linkage is composed by three movable bodies (2, 3, and 4) and a fixed one, the frame (1); four kinematic pairs: KP(1-2) of type revolute (R, $k = 1$) at joint A, KP (2-3) of type spherical (S, $k = 3$) at joint B, KP(3-4) of type cylindrical (C, $k = 2$) at joint C and KP($1 - 4$) of type prismatic (P, $k = 1$) at joint D. Moreover, this linkage has one degree of freedom $f = 1$ which is represented by the rotation angle (θ) between body 2 and the frame 1.

Once the basic topology of the multibody system has been recognised, the following information can be extracted: $N = 4$, $N_m = 3$, $P = 4$, $f = 1$ and its structural graph can be represented as shown in Fig. 8(b), where: vertices (1 to 4) correspond to the bodies of the multibody system, and two bodies that form a kinematic pair are connected by a number of edges (thin lines) equal to the degree k of the kinematic pair. Moreover, at each kinematic pair, a number of thin lines equal to the number of driven coordinates must be transformed into bold lines known as root edges. In Fig. 8(b) the bold line between vertices 1 and 2 represents that the degree of freedom in KP(1-2) is driven.

By considering Eq. (4) and the fact that each line in the structural graph, thin or bold, identifies one DOF, we rewrite Eq. (3) as:

$$L_{th} = B(P - N_m) \tag{5}$$

where L_{th} is the number of thin lines in the structural graph that corresponds to the P *active* kinematic pairs in the kinematic chain: internal kinematic pairs and, from the external ones, only those with a directed edge.

Then, with the support of the structural graph, to identify which kinematic pairs are active, Eq. (5) must be used so that if the left-hand side equals the right-hand side, the kinematic chain under study is a simple structural group. This procedure can be applied to obtain the kinematic structure of the MBS straightforwardly, as depicted in Figs. 8(c) to 8(f), following four basic steps:

Step 1: Frame (body 1) isolation and DOF assignment (Fig. 8(c)). Since the frame is a fixed body, it cannot belong to any structural group, and it has to be excluded from the analysis (isolated from other bodies). The dashed curve in this figure reflects this frame isolation. Then, all the DOF of the kinematic pairs between any movable body and the frame are assigned to the movable ones; in this example, one driven DOF is assigned to 2 and one DOF to 4. The DOFs assignment is represented by a directed edge in the structural graph, making the corresponding kinematic pairs *active* to the bodies that receive them. In Fig. 8(c), after the frame isolation, the KP(1-2) is active for body 2 while KP(2-3) is not, and for body 4 the KP(1-4) is active while KP(3-4) is not. Finally, any body in this step or in a later stage of the structural analysis which receives one or more DOF both from the frame or from other bodies becomes *candidate* to be an SG.

Step 1: Search for SGs from shorter to larger length. First, each candidate obtained in the previous step is studied (no matter in which order), and any of them which satisfies Eq. (5) is an SG. In Fig. 8(c), the candidate in the KC{4} has ($N_m = 1$, $P = 1$) since it is a single body, KP(1-4) is active and KP(3-4) is not. In Eq. (5) we have, on the left hand side: $L_{th} = 1$ and on the right hand side: $B(P - N_m) = 6(1 - 1) = 0$; this equation is not fulfilled and KC{4} does not form a SG. Next, the candidate in the KC{2} is studied. It participates in only one active external KP(1-2), which is a root edge, then: $N_m = 1$, $P = 1$ and $L_{th} = 0$. Thus, this candidate satisfies Eq. (5) and forms a simple structural group SG-I{2}. This is represented in the structural graph by a dashed line around the corresponding body (or bodies). If none of the kinematic chains of size one (one candidate) forms a structural group, kinematic chains of size two must be analysed with at least one candidate in

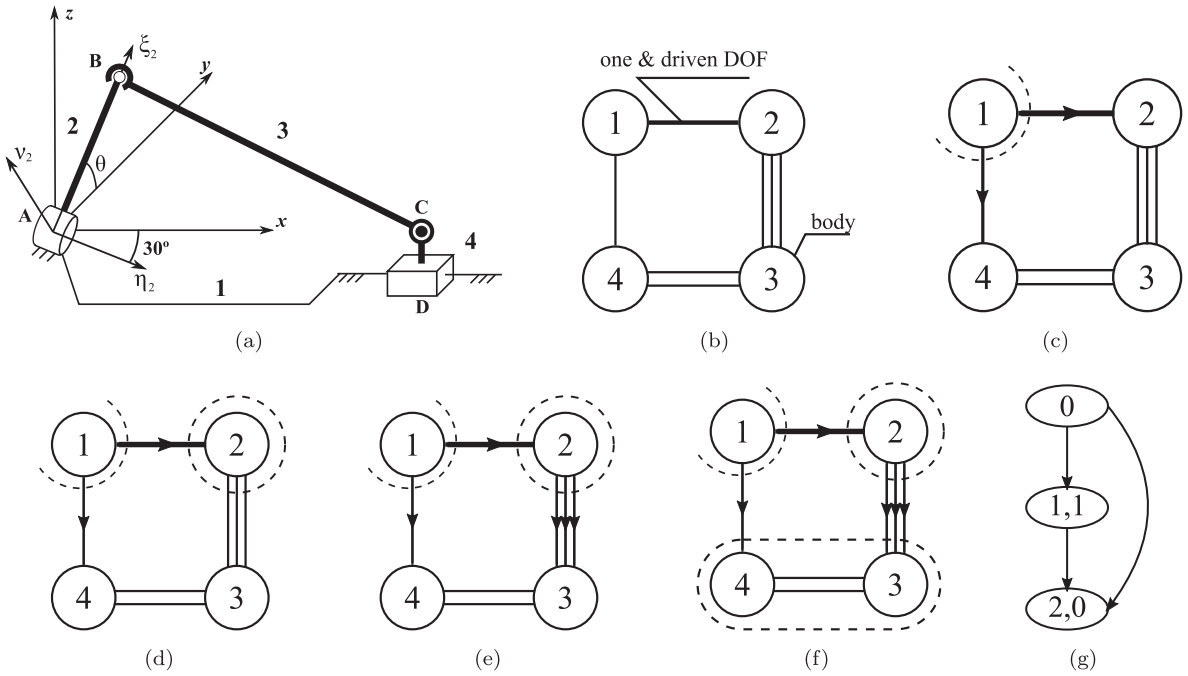


Fig. 8. Spatial four-bar linkage. (a) Kinematic graph. (b) Structural graph. (c) to (f) Steps to perform structural analysis through its structural graph. (g) Structural diagram.

it. This process continues, increasing the length of the kinematic chain in one body at a time until, in the worst case, the whole MBS satisfies Eq. (5) meaning that it cannot be divided into smaller structural groups.

Step 1: Reassign DOF. If a kinematic chain forms an SG in the previous step, the DOF of its non-active external pairs are assigned to the corresponding bodies and, for these bodies, the cited external pairs become active. In the example, as body 2 becomes an SG, it assigns and makes active to body 3 the three DOF of the KP(2-3). As stated in step 1, since body 3 receives DOF from another body, it becomes a new *candidate* (Fig. 8(e)).

Step 1: Turn to Step 2 until the complete kinematic structure of the MBS is obtained. Bodies 3 and 4 are *candidates*. Kinematic chains of only one candidate KC{3} or KC{4} do not satisfy Eq. (5) since $L_{\text{th}} = 1$ and $L_{\text{th}} = 3$, respectively. Then, larger chains must be considered. Starting from one candidate, e.g. 3, the chain is expanded by selecting another body joined to the candidate. In the example, there is only one possibility, KC{3,4}, formed by two bodies and three active kinematic pairs, being: KP(2-3) and KP(1-4) external, and KP(3-4) internal. The parameters for this kinematic chain are: $L_{\text{th}} = 6$; $N_{\text{m}} = 2$; $P = 3$, which satisfies Eq. (5) ($B(P - N_{\text{m}}) = 6(3 - 2)$) and therefore it is a simple SG as show in Fig. 8(f).

Structural diagram. The kinematic structure of a mechanism is graphically represented by its *structural diagram* (Fig. 8(g)), also introduced in [21]. It is composed of one circle representing the frame (0) and as many circles, as obtained SG, with two parameters inside each one: $(N_{\text{m}}, n_{\text{c}})$. Recall that an arrow joins two circles in the same direction as the DOF assigned during the structural analysis, showing the order in which the SG has been obtained and the hierarchical sequence in which their kinematics must be solved. In the example, the frame assigns DOF to bodies 2 and 4, and body 2 assigns DOF to body 3, meaning that the kinematics of SG-I{2} must be solved first and then, the kinematics of SG-II{3,4}.

3. Structural analysis algorithm for spatial multibody systems

This section describes the algorithm **MBSSA3D** (MultiBody Systems Structural Analysis 3D), developed for the computational structural analysis of planar and spatial multibody systems.

This algorithm is divided into five main *blocks* or *modules* which are executed sequentially: gathering topological information, initialisation, singularities recognition, structural analysis, and finally, results and modelling (Fig. 9).

3.1. Module 1. Gathering topological information

In this first module, the algorithm gathers basic topological information of the MBS. The user has two options to provide the necessary information: a developed guided user interface (GUI) or loading this information from a file. The first option is more comfortable, practical to avoid human mistakes and easier to use for a non-experienced user; the second option is faster when

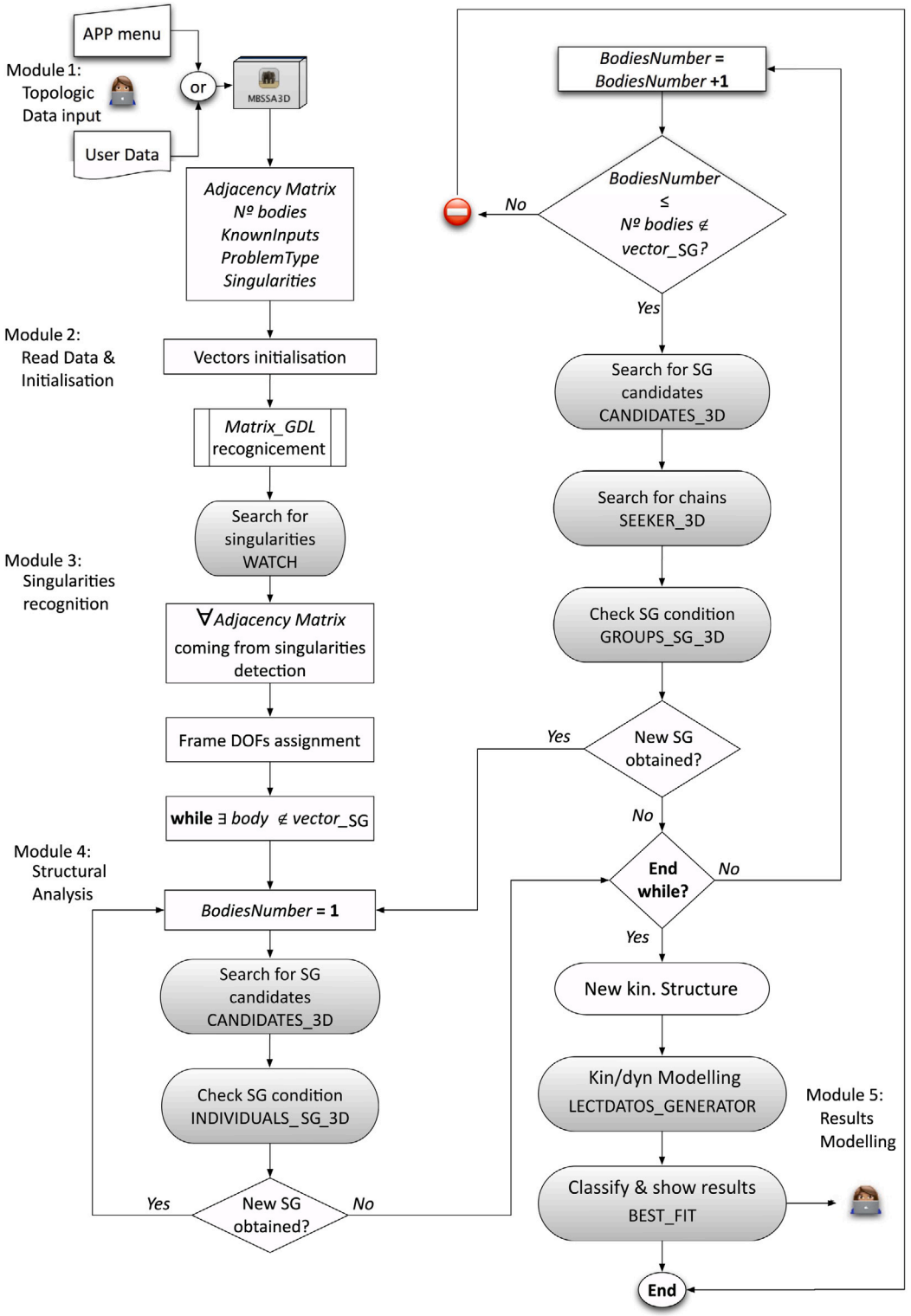


Fig. 9. MBSSA3D algorithm modules and workflow.

modelling complex MBS. Whatever option is selected, this first module organises the topological information into the five following entities:

Table 3
Code of characters used in the extended adjacency matrix to identify the *Type* of kinematic pair formed by two bodies together with its description and allowed degrees of freedom (DOFs), known as grade of the kinematic pair (k).

Type	Kinematic pair type	DOF
–	Not connected	0
R	Revolution	1
P	Prismatic	1
H	Helicoidal	1
CPEP	Penta Punctual Contact	1
C	Cylindrical	2
ER	Slotted Spherical	2
U	Universal	2
L	Cam/point–curve/curve–curve	2
EO	Ordinary Gear	2
EPI	Epicyclic Gear	2
CTEP	Tetra punctual Contact	2
RP	Pure Rolling	2
RD	Sliding Rolling	2
S	Spheric	3
PL	Planar	3
CR	Slotted Cylindrical	3
CTRP	Tri Punctual Contact	3
FP	Free body Plane	3
SC	Sphere–Cylinder	4
PC	Plane–Cylinder	4
CBIP	Bi Punctual Contact	4
SP	Sphere–Plane	5
SPSP	Sphere–Sphere	5
F	Free body Spatial	6

1. ProblemType: sets the B parameter value (Eq. (3)) for the planar o spatial problem.
2. N: the total number of bodies in the multibody system as previously introduced.
3. Extended adjacency matrix: in this work, the concept of *extended adjacency matrix* $M(i,j) = M(j,i)$ is introduced, whose elements are characters if two bodies form a kinematic pair or a dashed line otherwise. The available characters are shown in Table 3 together with the description of the kinematic pair that they represent and their grade (k). Most of these characters are extensively used in the literature, while others are introduced here. An extended adjacency matrix exempts the user from knowing, in advance, the degrees of freedom or additional features that will be automatically translated later by the algorithm for the different types of kinematic pairs.
4. KnownInputs: bold lines between two bodies in the structural graph represent controlled or driven DOFs among them and, if the multibody system is kinematically determined, the total number of these lines must coincide with its mobility f. The number w_p of driven DOFs in KP(i,j) of grade k must be provided by the user and included in the algorithm through the symmetric and integer matrix $KnownInputs(i,j) = KnownInputs(j,i) = w_p$. In the present version of the algorithm, if the total number of driven DOFs differs from the mobility of the MBS or if the selected ones do not allow kinematic substructuring, a pop-up message warns the user about the error.
5. Singularities: in some multibody systems, Eq. (3) might fail to calculate their mobility due to different types of topological or kinematical singularities. The types of singularities and the fundamentals of how these singularities will be treated in kinematic substructuring will be introduced in Section 3.3 although is still under development.

Fig. 10 shows the **MBSSA3D** guided user interface developed to gather the topological information from the user in the first main module. Missing or mismatched inputs are checked by the algorithm and reported to the user by the appropriate warning pop-ups. All the matrices and variables needed for kinematic substructuring are automatically generated as the user introduces the requested information.

For the spatial four-bar linkage shown in Fig. 8(a), the file generated by the algorithm, which might also be written directly by the user, should be similar to the code shown in the script `User_Data.m`. It is straightforward the correspondence between the information represented in Fig. 8(b) by the structural graph, the matrices shown in the cited script, and the information shown in the output window named "Current Adjacency Matrix obtained" and "Inputs" at the right bottom of the GUI.

3.2. Module 2. Read data and initialise

The algorithm reads the topological information to create the matrices needed for kinematic substructuring. In matrices of size $\mathbb{R}^{1\times N}$ each column i stores the information related to the body it represents:

- $vector_{\mathrm{PAIR}} \in \mathbb{R}^{1\times N}$: total number of active kinematic pairs in which this body participates,

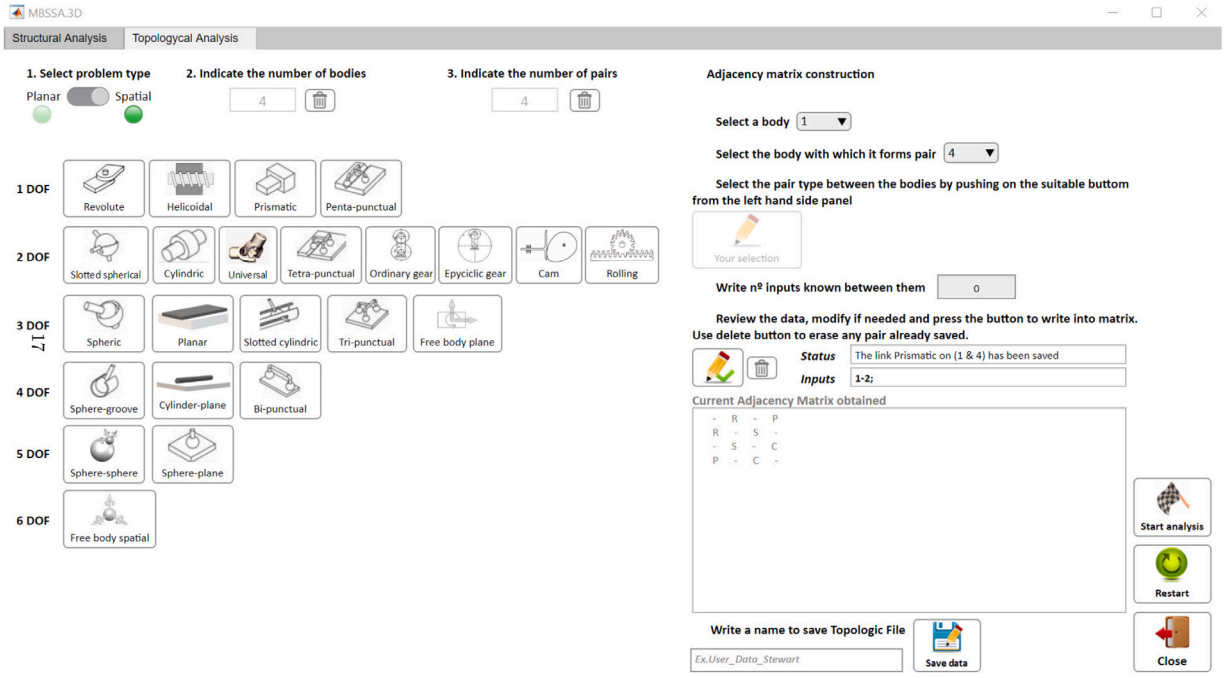


Fig. 10. **MBSSA3D** guided user interface developed to gather topological information from the user.

Script `User_Data.m`

```
1  global ProblemType N M KnownInputs
2   %Topological information
3  ProblemType=6 ; %Select 3 for planar and 6 for spatial mechanism
4  M={ ... ; %Extended adjacency matrix based on Type
5  '-' 'R' '-' 'P';
6  'R' '-' 'S' '-';
7  '-' 'S' '-' 'C';
8  'P' '-' 'C' '-'};
9  N=length(M) ; %Number of bodies including chassis
10 KnownInputs=[ ... ; %Known inputs matrix
11 0 1 0 0;
12 1 0 0 0;
13 0 0 0 0;
14 0 0 0 0];
```

- $vector_{\mathrm{BL}} \in \mathbb{R}^{1\times N}$: number of active, driven DOFs (directed bold lines in the structural graph) received during the assigning phase (step 2),
- $vector_{\mathrm{DOF}} \in \mathbb{R}^{1\times N}$: the number of degrees of freedom regardless they are driven (directed bold lines) or not (directed thin lines), which corresponds to the P active kinematic pairs for body i,
- $vector_{\mathrm{SG}} \in \mathbb{R}^{1\times N}$: bodies assigned to a structural group,
- $candidate_{\mathrm{SG}} \in \mathbb{R}^{1\times N}$: bodies candidates to form structural group,
- $candidate\_chain_{\mathrm{SG}}$: matrix of variable size, stores the kinematic chains with more than one body which are candidates to form a structural group.
- $matrix_{\mathrm{GDL}} \in \mathbb{R}^{N\times N}$: stores the DOF accumulated by KP(i,j).
- $CounterPair \in \mathbb{R}^{N\times N}$: takes into account the active kinematic pairs assigned from body i to other bodies when it becomes a structural group.
- $Results_{\mathrm{SA}} \in \mathbb{R}^{N\times N}$: The first row indicates the roman number N in SG-N of the structural group to which the body of column i participates, and the elements in $Results_{\mathrm{SA}}(y,j) = numAssg$ stores $numAssg$, a counter indicating that a new assignment has

taken place ($numAssg = 1$ after step 1), j are the bodies that assigns its non-active external kinematic pairs (the frame, in step 1) and y are the bodies that receive these DOFs.

- $MBSSG$: row matrix which contains the kinematic structure of the MBS.

After the initialisation of the aforesaid elements, the extended adjacency matrix is analysed to create $matrix_{\mathrm{GDL}}$ as shown in algorithm 1, which is also self-explanatory.

Algorithm 1: Degrees of freedom identification from M

```
for a = 1 : N − 1 do
    for b = a + 1 : N do
        switch Kinematic Pair Type in M(a, b) do
            case 'R', 'P', 'H' or 'CPEP' do
                matrix_GDL(a, b) = matrix_GDL(b, a) = 1;
            case ... other Kinematic pairs ... do
                see Table 3
        end
    end
end
```

Following the diagram shown in Fig. 9, when algorithm 1 is applied to the MBS of Fig. 8, $matrix_{\mathrm{GDL}}$ (Eq. (6)) is obtained and the MBSSA3D algorithm enters in the next main module.

$$M = \begin{pmatrix} '-' & 'R' & '-' & 'P' \\ 'R' & '-' & 'S' & '-' \\ '-' & 'S' & '-' & 'C' \\ 'P' & '-' & 'C' & '-' \end{pmatrix};\ matrix_{\mathrm{GDL}} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 3 & 0 \\ 0 & 3 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix} \tag{6}$$

3.3. Module 3. Singularities recognition

In the literature, many exceptions make the Kutzbach–Grubler Eq. (2) fail when predicting the mobility of a multibody system and their study is the objective of this third module. We cite here which singularities the algorithm should handle but how they will be treated in kinematic substructuring and its implementation are still under development.

(a) Multijoints. In the case of rotational joints, when n_b bodies ($n_b > 2$) are joined at the same joint, Eq. (2) holds, only if $P_1 = n_b - 1$ kinematic pairs of grade $k = 1$ are considered at this joint. In order to find the best kinematic structure in these cases, the user must introduce all the possible kinematic pairs ($nComb = \frac{n_b(n_b-1)}{2}$) through the extended adjacency matrix and perform: $N_{an} = \frac{nComb!}{(n_b-1)!\,(nComb-n_b+1)!}$ structural analysis. If a multibody system shows more than one multiple joint (i.e. A and B), the algorithm will study all the possible combinations of kinematic pairs in B for each possible combination of kinematic pairs in A.

(b) Rolling without slipping. Means that the contact point of the rolling bodies must have the same velocity. When this kinematic constraint at the velocity level is holonomous, it may be integrated, offering well-known constraint equations at the position level, which reduces the grade k of the kinematic pair. Fig. 10 shows three types of rolling without slipping we have in mind: ordinary, epicycloidal and rack-pinion gear trains (or similar) and Table 3 shows their nomenclature and grade.

(c) Redundant DOFs. Double-spheric articulated bodies or structural groups introduce unnecessary degrees of freedom (redundant) for the kinematic or dynamic analysis of the multibody system. Here, the body's rotation along the axis formed by the two spherical joints is fixed to a known value, irrelevant for structural analysis. The algorithm shall check for this condition and introduce the appropriated changes where the DOF consideration is impacted.

(d) Additional DOFs. Certain geometric conditions in multibody systems add additional DOFs that cannot be considered by the parameters in Eq. (2). Some examples are the tangent redundancy in the double parallelogram linkage, or when all the axis of the rotational joints cut in a fixed point (spherical linkages) or at infinite (planar multibody systems studied as 3D ones). Geometric singularities shall be introduced in the topological analysis for the consequent handling by the algorithm in the DOF calculation.

(e) In some kinematic pairs, known as direct-contact pairs, as the cam-follower case, the contact between the two bodies may be assured by geometry or by force elements, as springs. In the latter case, contact might be lost and the kinematic structure might change. This option may be introduced in MBSSA3D to generate the corresponding kinematic structures which will be considered in the generation of the kinematic or dynamic models of the multibody system.

3.4. Module 4. Structural analysis

This module performs the kinematic substructuring of the multibody system following the sequence introduced for the graph-analytic method (Fig. 8). At step 1, the first assignment of DOF goes to the bodies (i) connected to the frame, which is identified from the first row of the extended adjacency matrix out of $matrix_{\mathrm{GDL}}$; then, the following matrices are updated: $vector_{\mathrm{PAIR}}(1, i) = 1$,

$vector_{\mathrm{DOF}}(1,i) = k$, where k is the grade of the KP(1,i) obtained from $matrix_{\mathrm{GDL}}$, and $Results_{\mathrm{SA}}$ as cited above. The driven coordinates are considered from matrix $KnownInputs$ to update $vector_{\mathrm{BL}}(1,i) = n_{\mathrm{c}}$ and assigned as shown in Algorithm 2.

After this first assignment, the MBSSA3D app starts a *while-end* loop until all the elements in the row matrix $vector_{\mathrm{NSG}} \in \mathbb{R}^{1\times N}$ become zero. This matrix identifies with 1 the bodies (columns) not assigned to an SG, and then, this loop ends when the complete kinematic structure has been obtained, or no kinematic structure is available. This task is carried out by Algorithm 3.

In Algorithm 3, two flags ($flag1$ y $flag2$) controls when a new structural group has been created, and each time an SG is obtained, the search for new ones must begin with kinematic chains formed by only one body; if none of the candidates forms a structural group by itself, the sum of the components of vector $vector_{\mathrm{NSG}}$ does not change, $flag1 = flag2$ and the variable $BodiesNumber$ is increased in one unit to search for longer kinematic chains.

Algorithm 3 is divided in two parts; the first one uses the **INDIVIDUALS_SG_3D** function to check Eq. (5) for one-body SG and the second part searches for kinematic chains using **SEEKER_3D** and SG formed by more than one body by means of **GROUPS_SG_3D**. Both parts call the function **CANDIDATES_3D** which provides the bodies candidates to form structural group. These functions are described next.

Algorithm 2: Frame isolation and DOF assignment

```
for c = 2 : N do
    if matrix_GDL(1, c) ~= 0 update:
        Results_SA(c, 1);
        vector_PAIR(1, c);
        vector_DOF(1, c);
    end
    if KnownInputs(1, c) ~= 0 update:
        vector_BL(1, c);
    end
end
```

Algorithm 3: Structural groups decomposition

```
Result: Results_SA
order_SG = 1;
flag1 = ∑ (vector_NSG[2 : N]);
flag2 = 0;
while ∃ vector_NSG[2 : N] ∉ 0 do
    if flag1 ~= flag2 then
        BodiesNumber = 1;
        % search for candidates
        CANDIDATES_3D
        % check if any candidate forms SG
        INDIVIDUALS_SG_3D
        flag2 = flag1;
        flag1 = ∑ (vector_NSG[2 : N]);
    else
        BodiesNumber = BodiesNumber + 1;
        CANDIDATES_3D
        SEEKER_3D
        forall candidate_chain_SG do
            GROUPS_SG_3D
            flag2 = flag1;
            flag1 = ∑ (vector_NSG[2 : N]);
            if flag1 ~= flag2 then
                break;
            end
            break;
            disp error % if not candidates_SG
        end
        break;
        disp error % if not Structural Analysis solution found
    end
end
StructuralAnalysisResults
```

Function **CANDIDATES_3D** identifies any body candidate to form a structural group, which must satisfy two conditions: it does not belong to any structural group already obtained, and the body has, at least, one active external kinematic pair.

Function CANDIDATES_3D

```
Result: candidate_SG
for i = 2 : N do
    for j = 1 : N do
        if Results_SA(i, j) ~= 0 & vector_SG(i) == 0 then
            candidate_SG(i) = i;
            %bodies with assigned DOFs but not included in any SG become candidates
        end
    end
end
```

In the example of the spatial four-bar, applying the function **CANDIDATES_3D** we obtain Eq. (7) showing that bodies 2 and 4 are candidates as, for both of them, the external kinematic pairs KP(1,2) and KP(1,4) are active.

$$candidate_{SG} = \begin{pmatrix} 0 & 2 & 0 & 4 \end{pmatrix}; \tag{7}$$

Function **INDIVIDUALS_SG_3D** is executed after function **CANDIDATES_3D** to check if any of the obtained candidates satisfies Eq. (5), and then, forms a structural group. The sequence is shown in Fig. 9. A pseudo-code of the main steps carried out is used to make this function more readable in **INDIVIDUALS_SG_3D**. Basically, for each candidate e the function reads all elements from columns 2 to N in row e within $Results_{SA}$, searching for the bodies f which are active external kinematic pairs to this candidate e but only if the KP(e,f) was not previously considered. Each kinematic pair found is added to $vector_{PAIR}(e)$ to calculate P of Eq. (5); moreover, from the set of active kinematic pairs it is easy to read from $matrix_{GDL}(f,e)$ the DOF of each of them which are added to $vector_{GDL}(e)$ to evaluate S_c; finally, from the matrix $knownInputs(f,e)$ it is also easy to read number of driven DOF in each active kinematic pair which is added to $vector_{BL}(e)$ to calculate n_c.

Eq. (5) is checked and, if satisfied by candidate e, the following matrices and variables are updated: $vector_{SG}$, $vector_{NSG}$, $candidate_{SG}$, $Results_{SA}$ and $order_{SG}$, and the external KP(e,g) not active for body e will become active for the bodies g which are not yet structural group.

Function INDIVIDUALS_SG_3D

```
Output: Results_SA, order_SG
forall candidate_SG do
    e=candidate_SG;
    for f=2:N do
        % evaluate P, DOF and nc
        if Results_SA(e,f) ~= 0 & CounterPair(e,f) = 0 update:
            vector_PAIR(e);
            vector_GDL(e);
        end
        if KnownInputs(e,f) ~= 0 update:
            vector_BL(e);
        end
        CounterPair(e, f)=1;
    end
    % Check SG condition and update matrices accordingly
    if (vector_GDL(e) − vector_BL(e)) = ProblemType · (vector_PAIR(e) − 1) then
        Results_SA(1, e) = order_SG;
        order_SG = order_SG + 1;
        update vector_SG(e), vector_NSG(e), candidate_SG(e);
        % New DOF assignment
        for g=2:N do
            if matrix_GDL(e,g) ~= 0 & vector_SG(g) ~= 0 then
                Results_SA(g, e) = order_SG;
            end
        end
    end
end
```

After the first execution of **INDIVIDUALS_SG_3D** over the example (Fig. 8.a), matrix $Results_{\mathrm{SA}}$ is resulting as (8):

$$Results_{\mathrm{SA}} = \begin{pmatrix} 0 & \mathbf{1} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & \mathbf{2} & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{8}$$

where the value from the first row, reserved for the kinematic structure of the MBS: $Results_{\mathrm{SA}}(1,2) = 1$ means that body 2 (second column) is the first ($order_{\mathrm{SG}} = 1$) structural group obtained.

As shown in algorithm 3, when **INDIVIDUALS_SG_3D** is executed, both flags are updated and compared. As in this case, a new SG has been obtained, $flag1 \neq flag2$, the functions **CANDIDATES_3D** and **INDIVIDUALS_SG_3D** are again executed and vector $candidate_{\mathrm{SG}}$ is updated:

$$candidate_{\mathrm{SG}} = \begin{pmatrix} 0 & 0 & 3 & 4 \end{pmatrix}; \tag{9}$$

Both candidates, 3 and 4, are analysed by function **INDIVIDUALS_SG_3D**. As the parameters for body 4 have not suffered any change from previous iteration, this body is not SG. Body 3 is also not a structural group; then $flag1 = flag2$, and it becomes necessary to increase in one unit the length of the kinematic chains that can form new structural groups.

Function **SEEKER_3D** is in charge to find kinematic chains candidates to form a new structural group; the conditions they must satisfy are:(1) their length is given by the value of $BodiesNumber$; (2) none of their bodies belongs to a SG already obtained and (3), at least, one of its bodies has one active external kinematic pair (the body must appear in $candidates_{SG}$).

First, the function calculates all chains combination of the $N-1$ bodies in the MBS taken $BodiesNumber$ at a time. The outcome is the matrix $chain$ with $\frac{(N-1)!}{((N-1)-BodiesNumber)!\ BodiesNumber!}$ rows and $BodiesNumber$ columns. The first condition is satisfied. Only the chains whose elements are connected according to the $matrix_{\mathrm{GDL}}$ are consider for the next step.

Over each row of $chain$ is checked the conditions: (2) none of their elements belongs to a SG which can be only satisfied if all these elements in $vector_{\mathrm{SG}}$ are zero, and then, the variable $check1 = 0$; and (3) at least one body has one active external pair is satisfied if one of the elements has already been stored in matrix $candidate_{\mathrm{SG}}$, and then, the variable $check2 \neq 0$.

The filtered kinematic chains by the fulfilment of $check1 = 0$ and $check2 \neq 0$, are extracted out from $chain$ and resulting in the matrix $candidate\_chain_{SG}$ with the SG candidate chains to continue with the structural analysis algorithm 3.

When function **SEEKER_3D** is executed for the spatial MBS of the example under study, the only kinematic chain which is candidate is:

$$candidate\_chain_{SG} = \begin{pmatrix} 3 & 4 \end{pmatrix}; \tag{10}$$

Function SEEKER_3D

Output: $candidates\_chain_{\mathrm{SG}}$

$chain = \binom{N-1}{BodiesNumber}$

forall $chain \cap matrix_{\mathrm{GDL}}$ **do**

  check1=$\sum vector_{\mathrm{SG}}([chain, :])$;

  check2=$\sum candidate_{\mathrm{SG}}([chain, :])$;

  if *check1 = 0 & check2 ~= 0* **then**

    $candidates\_chain_{\mathrm{SG}} = chain$;

  end

end

Function **GROUPS_SG_3D** studies if any of the kinematic chains contained in $candidates\_chain_{\mathrm{SG}}$ satisfies Eq. (5) considering both external (active) and internal kinematic pairs to evaluate P, all the DOFs S_c contained in these kinematic pairs, which ones are driven n_c and which ones are not L_{th}. At this stage of the structural analysis, these parameters have already been stored for all bodies with active external kinematic pairs but must be modified due to internal kinematic pairs.

The function checks if any two bodies i and j of a kinematic chain contained in $candidates\_chain_{\mathrm{SG}}$ form KP(i,j) and, if so it adds: 0.5 units to the columns i and j from $vector_{\mathrm{PAIR}}$, $S_C/2$ units to the same columns from $vector_{\mathrm{GDL}}$ and $n_C/2$ units to the ones from $vector_{\mathrm{BL}}$.

Next, the condition for structural group, Eq. (5), is applied to the kinematic chain under study and, if it is a structural group, matrices: $vector_{\mathrm{SG}}$, $vector_{\mathrm{NSG}}$, $candidate_{\mathrm{SG}}$, $Results_{\mathrm{SA}}$ and variable $order_{\mathrm{SG}}$ will be updated, included the new DOF assignment of the external kinematic pairs which become active to the bodies that form kinematic pair with the bodies of the obtained SG. If Eq. (5) is not fulfilled, the parameters: $vector_{\mathrm{PAIR}}$, $vector_{\mathrm{GDL}}$ and $vector_{\mathrm{BL}}$ will be reset to their original values without accounting for internal kinematic pairs.

When **GROUPS_SG_3D** is applied to the MBS studied as an example, the function returns Eq. (11):

$$Results_{\mathrm{SA}} = \begin{pmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{11}$$

Function GROUPS_SG_3D

Output: $Results_{SA}$, $order_{SG}$

foreach *chain* $\in$ $candidates\_chain_{SG}$ **do**
  for *e=1:length(chain)-1* **do**
    for *f=e+1:length(chain)* **do**
      % evaluate P, DOF and n_c
      if $Matrix_{GDL}([chain(e), chain(f)]) \sim= 0$ **then**
        $vector_{PAIR}([chain(e), chain(f)]) = vector_{PAIR}([chain(e), chain(f)]) + 0,5$;
        $vector_{GDL}([chain(e), chain(f)]) = vector_{GDL}([chain(e), chain(f)]) + Matrix_{GDL}([chain(e), chain(f)])/2$;
      end
      if *KnownInputs(chain(e),chain(f))* $\sim= 0$ **then**
        $vector_{BL}([chain(e), chain(f)]) = vector_{BL}([chain(e), chain(f)]) + KnownInputs(chain(e), chain(f))/2$;
      end
    end
  end
  $P = \sum vector_{PAIR}([chain])$;
  $S_C = \sum vector_{GDL}([chain])$;
  $L_G = \sum vector_{BL}([chain])$;
  $N_m = length(chain)$;
  % check SG condition and update matrices accordingly
  if S_c - L_G = *ProblemType* · (P - N_m) **then**
    $Results_{SA}(1, [chain]) = order_{SG}$; ;
    $order_{SG} = order_{SG} + 1$;
    update $vector_{SG}([chain])$, $vector_{NSG}([chain])$, $candidate_{SG}([chain])$;
    % DOF assignment
    for *g=2:N* **do**
      if $matrix_{GDL}([chain],g) \sim= 0$ & $vector_{SG}(g) \sim= 0$ **then**
        $Results_{SA}$(g,[chain])=$order_{SG}$;
      end
    end
  % if the kinematic chain is not a SG the original values are restored
  else
    restore all
  end
end

where the first row contains the kinematic structure of the multibody system: SG-I{2} and SG-II{3,4} which coincides with the kinematic structure obtained using the graph-analytic method and shown in Fig. 8(g)). The remaining rows indicate the sequence in which the DOFs are assigned from one body in a structural group to the bodies in their external kinematics pairs. For example, in Eq. (11), the values $Results_{SA}(2, 1) = 1$ and $Results_{SA}(4, 1) = 1$ shows that body 1 (first column) has assigned DOFs to bodies 2 and 4. Then, the value $Results_{SA}(3, 2) = 2$ informs us that the second assignment of DOFs goes from body 2 (second column) to body 3 (third row).

3.5. Module 5. Results and modelling

The last module in the **MBSSA3D** algorithm offers the user two options: 'Structural Analysis Results' and 'Structural Modelling'. The objective of the latter is ongoing and not implemented. It should call an interface program that automatically generates and stores the kinematic/dynamic model of the multibody system in a text file that can be read from an external library to perform these types of analyses. We have two mature libraries in mind at this moment: MBDSBEG from [6], and MBSLIM from [36].

The option 'Structural Analysis Results' shows the kinematic structure obtained for the MBS analysed (Fig. 11). When several solutions (structural transformations) are obtained, they can be ordered attending to different criteria, as the number of SG in the kinematic structure, or the number of SG available in a specified library.

The most relevant information of the kinematic structure is formatted for the interface program and organised in different columns (see Fig. 11 upper side) being a explicit representation of the matrix's $Results_{SA}$ (Eq. (11)) rows and columns, meaning:

- Structural Groups: lists the ordered set of structural groups obtained in the format $/i, j/$, where i and j are bodies of this structural group. For the example used along the paper, first SG is $/2/$ and second one $/3, 4/$.
- N of bodies: shows the number of bodies in the SG of the same row.

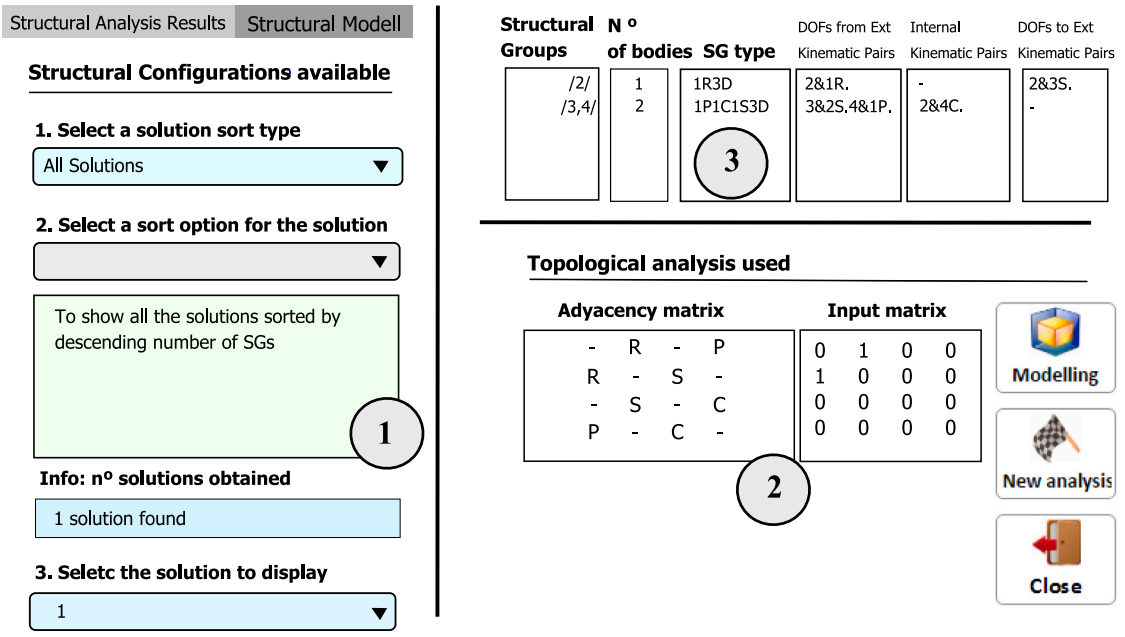


Fig. 11. Results window in the **MBSSA3D** app for the structural analysis of multibody system shown in Fig. 8(a).

- SG type: the name assigned to this type of structural group. The format for the name is a sequence of characters combining the numbers of kinematic pairs of a certain type (i.e. 1P1C1S means one prismatic, one cylindrical, one spherical) ending with a '2D' or a '3D' for planar or spatial MBS, respectively. This is thought to be the baseline for the next steps required in the modelling libraries connection.
- DOFs from external kinematic pairs: lists the active external kinematic pairs from which bodies of the SG of the same row participate. The format is i&jT. where i is the body that belongs to the SG, j is the body out of the SG, and T is the type of kinematic pair; the dot is used to end the chain of characters and to separate different external active kinematic pairs. In the example: 3&2S.4&1P. means that KP(3,2) is spherical and KP(4,1) is prismatic.
- Internal pairs: lists the internal kinematic pairs in the same format that the external ones.
- DOF to external kinematic pairs: lists how many DOFs assigns one body i that becomes SG to other bodies j of the KP(i, j) which becomes active to the later (j). The format is similar to the preceding item and, in the example, 2&3S. means that body 2 assigned the three DOFs of the S spherical KP$(2, 3)$ to body 3.

The bottom part of the interface Fig. 11 is dedicated to inform the analyst about the MBS topological input (the extended adjacency matrix and $KnownInputs$ matrix) used for the structural analysis. That will be specially useful when different solutions are handling by the algorithm on structural transformation and singularities management in the future.

4. Case studies

In this section, the algorithm **MBSSA3D** has been applied for the kinematic substructuring of spatial multibody systems. The results are validated with the graph-analytical method introduced in Section 2 and compared to the kinematic structures offered by a graph-analytical LGA, and the computational linear graph algorithm introduced in [13].

4.1. Rotor of a helicopter

Fig. 12.a shows part of the main rotor control mechanism of a helicopter introduced by Kecskeméthy in [11] with the corresponding topological graph in Fig. 12.b. The mechanism consists of 15 movables bodies and the frame **1**, joined through 21 joints giving rise to four prismatic pairs: KP(1-2), KP(1-3), KP(1-4) and KP(1-13), five revolute pairs: KP(8-9), KP(11-9), KP(13-9, KP(13,15) and KP(15-16), six universal pairs: KP(2-5), KP(3-6), KP(4-7), KP(1-10), KP(8-12) and KP(11-14), and six spherical pairs KP(5-8), KP(6-9), KP(7-11), KP(10-9), KP(12-16) and KP(14-16). Moreover, the rotational movements of KP(9-13) and KP(13-15) and one of the two DOFs in KP(11-14) are considered as the independent coordinates.

This mechanism is originally modelled in [11] with 18 nonlinear constraint equations from six independent closed-loops open at their spherical joints. Following Fig. 12.b and opening at the spherical joints, the kinematic structure is formed by these six modules with the corresponding bodies: CL-I $\{1, 13, 9, 10\}$, CL-II: $\{1, 13, 9, 6, 3\}$, CL-III: $\{13, 9, 11, 14, 16, 15\}$, CL-IV$\{1, 13, 9, 11, 7, 4\}$, CL-V: $\{13, 15, 16, 12, 8, 9\}$, and CL-VI: $\{1, 13, 9, 8, 5, 2\}$. The kinematic graph in Fig. 12.c, reveals the closed-loops dependency between closed-loops.

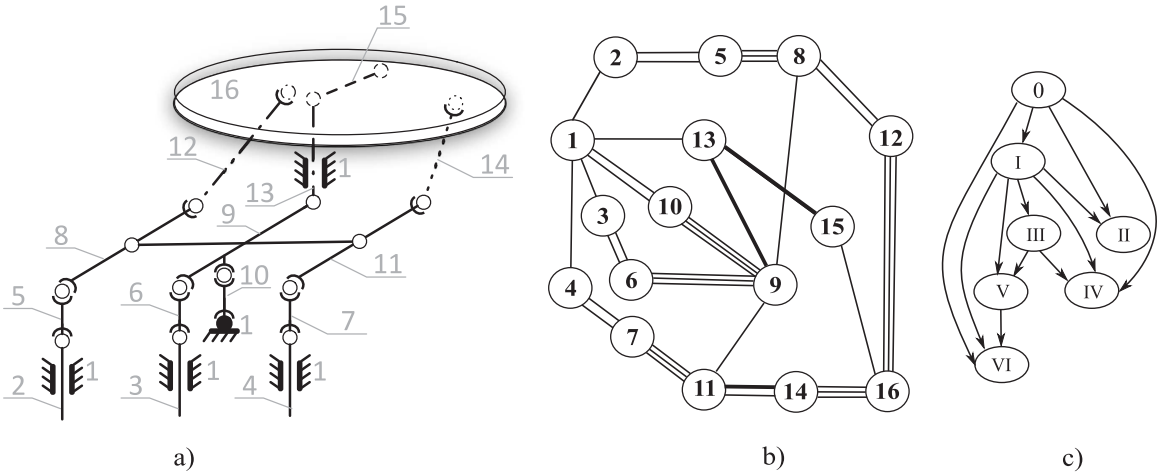


Fig. 12. (a) Main rotor control mechanism of a helicopter introduced by Kecskeméthy in [11]. b) Topological graph considering bold lines in KP(9-13), KP(13-15), and one of the two DOFs in KP(11-14) as independent coordinates. c) Kinematic graph for the six independent closed-loops when cut at the spherical joints.

Table 4
Parameters for computational kinematic substructuring of the rotor using the LGA described in [13].

List		Items
P	=	[0, 1, 1, 1, 2, 3, 3, 9, 13, 1, 9, 8, 1, 11, 13, 15, 5, 10, 6, 7, 16, 12]
loop_terminators	=	[[9, 18], [9, 19], [14, 21], [11, 20], [16, 22], [8, 17]]
parent_joints	=	[None, 'S ', 'E', 'M', 'T', 'F', 'N', 'P', 'B', 'C', 'H', 'Q ', 'A', 'I', 'J', 'K', 'U', 'D', 'G', 'O', 'L', 'R']

Table 5
Results for the computational kinematic substructuring of the rotor using the LGA described in [13].

Substructure id	Loops	Loop members	Loop correspondence
1	i	1, 13, 10, 9, 18	CL-I
2	ii	1, 13, 3, 6, 9, 19	CL-II
3	iv	1, 13, 4, 9, 7, 11, 20	CL-IV
4	vi	1, 13, 2, 9, 5, 8, 17	CL-VI
5	[iii], [v]	[13, 9, 15, 11, 16, 14, 21], [13, 15, 9, 8, 12, 16, 22]	CL-III + CL-V

LGA computational solution

Once the closed-loops and their body members have been identified, we may provide the computational linear graph algorithm described in [13] with the information shown in Table 4. Virtual bodies $\{17, 18, 19, 20, 21, 22\}$ have been included between the kinematic pairs: KP(4-7), KP(8-9), KP(5-8), KP(6-10), KP(13-15) and KP(11-15).

The algorithm offers a kinematic structure divided into the five kinematic substructures in Table 5. In the last column, we identify the correspondence between the loops in the visual and the computational LGA methods. Same results mean that we made most of the algorithm's work and the provided information (Table 4) was closed to the solutions. This kinematic structure is not used by their authors for a combination of sequential/parallel solution of the closed loops, but to reorganise the elements of the Jacobian matrix of the multibody system. As the fifth substructure is composed by two loops (CL-III and CL-V), the corresponding Jacobian sub-matrix must be less efficient than a kinematic structure containing six modules.

As the kinematic structure of a multibody system is not unique, we tested the computational algorithm with a different set of closed-loops, searching for a sixth module. In this second case, the closed-loops and members are: CL-I $\{1, 13, 9, 10\}$, CL-II: $\{1, 2, 5, 8, 12, 16, 14, 11, 7, 4\}$, CL-III: $\{1, 2, 5, 8, 12, 16, 15, 13\}$, CL-IV$\{1, 2, 5, 8, 9, 10\}$, CL-V: $\{1, 9, 10, 13, 3, 6\}$, and CL-VI: $\{1, 2, 5, 8, 12, 16, 14, 11, 9, 10\}$. Including the virtual bodies $\{17, \ldots, 22\}$ now between the kinematic pairs: KP(9-13), KP(1-4), KP(15-16), KP(8-9), KP(6-9) and KP(9-11). The new information for the algorithm is shown in Table 6, and the new results, in Table 7.

The new kinematic structure consist in the six substructures shown in Table 7. Again, we find an exact correspondence with the visual LGA results for this structural transformation, and an important effort has been done collecting the information needed by the algorithm.

MBA graph-analytical solution

The kinematic structure based on structural groups must be obtained for the kinematic and dynamic analysis based on group equations [6]. Applying the concepts introduced in Section 2 to the rotor control mechanism and considering the same input

Table 6
Modified parameters in one structural transformation for the computational kinematic substructuring of the rotor.

List		Items
P	=	[0, 1, 1, 7, 2, 3, 11, 5, 10, 1, 14, 8, 1, 16, 13, 12, 13, 4, 16, 8, 9, 11]
loop_terminators	=	[[9, 17], [1, 4], [19, 15], [20, 9], [6, 21], [9, 22]]
parent_joints	=	[None, 'S ', 'E', 'N', 'T', 'F', 'O', 'U', 'D', 'C', 'I', 'Q ', 'A', 'L', 'J', 'R', 'B', 'M', 'K', 'P', 'G', 'H']

Table 7
Substructures for the structural transformation of the rotor using computational LGA [13].

Substructure id	Loops	Loop members	Loop correspondence
1	i	1, 10, 13, 9, 17	CL-I
2	iii	1, 2, 13, 5, 8, 12, 16, 19, 15	CL-III
3	iv	1, 2, 10, 5, 8, 20, 9	CL-IV
4	v	1, 3, 10, 9, 6, 21	CL-V
5	vi	1, 10, 2, 5, 8, 12, 16, 14, 11, 9, 22	CL-VI
6	ii	2, 5, 8, 12, 16, 14, 11, 7, 1, 4	CL-II

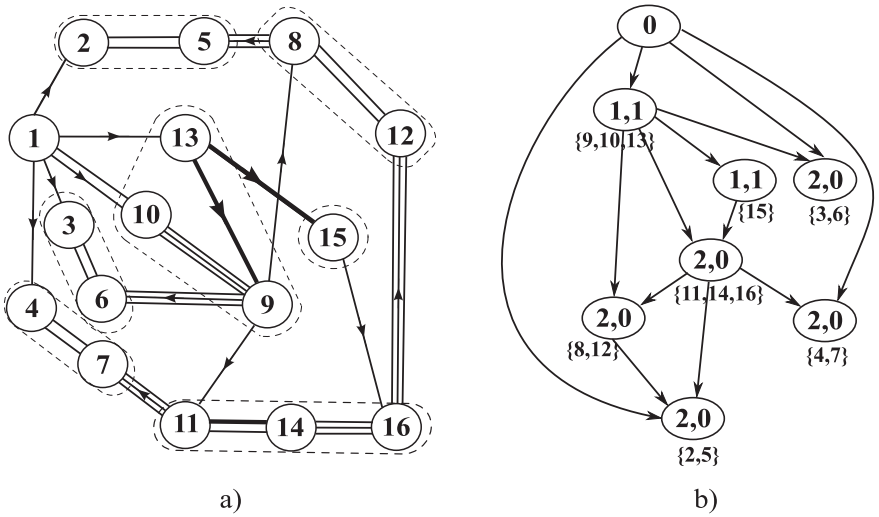


Fig. 13. (a) Topological graph and structural groups obtained for the rotor control mechanism in Fig. 12. b) Structural graph for the seven independent structural groups.

movements as in [11], an experienced analyst would obtain the structural graph in Fig. 13.a showing the seven structural groups obtained and their corresponding bodies. Fig. 13.b shows the corresponding structural graph, where the careful distribution of the SGs reflects their dependency and how they must be solved (from the upper to the lower level). For example, SG-I{9,10,13} is at the first level and must be solved first. Then, SG-II{15} and SG-III{3,6} are at the second level and might be solved in parallel if explicit parallelism is considered [7] or sequentially otherwise. At the third level, SG-IV{11,14,16} must be analysed. Then, at the fourth level, SG-V{8,12} and SG-VI{4,7} may be solved in parallel and finally, at the lower level, solving the last SG-VII{2,5} will end the solution of the whole multibody system.

MBSSA3D Computational solution

The new module 'Results' for the computational solution with MBSSA3D algorithm offers the main results in tabular form. The first column in Table 8 shows the same structural groups than with the graph-analytical method. The other columns and Table 9 offer information devoted to classifying this specific solution among other kinematic structures obtained if structural transformation is launched. Also, this information is needed for an interface program that automatically generates the kinematic or dynamic model of the multibody system. Both, the automatic structural transformations and the interface to generate the kinematic and dynamic model are ongoing.

The obtained SGs are ordered in the matrix $Results_{SA}$.M first row, which defines the kinematic structure of the problem at hand. The solution offered by the MBSSA3D algorithm (Eq. (12)) coincided with the one provided in Fig. 13.a.

$$Results_{SA} = \begin{pmatrix} 0 & 7 & 3 & 5 & 7 & 3 & 5 & 6 & 1 & 1 & 4 & 6 & 1 & 4 & 2 & 4 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \tag{12}$$

Table 8
Kinematic structure offered by MBSSA3D for the rotor control mechanism with the original independent coordinates.

SGs	Num bodies	SG type	DOFs from Ext KP	Int KP	DOFs to Ext KP
9,10,13	3	1R1P1U1S3D	10&1U.13&1P.	9&10S.9&13R.	9&6S.9&8R.9&11R.13&15R.
15	1	1R3D	15&13R.	–	15&16R.
3,6	2	1P1U1S3D	3&1P.6&9S.	3&6U.	–
11,14,16	3	2R1U1S3D	11&9R.16&15R.	11&14U.14&16S.	11&7S.16&12S.
4,7	2	1P1U1S3D	4&1P.7&11S.	4&7U.	–
8,12	2	1R1U1S3D	8&9R.12&16S.	8&12U.	8&5S.
2,5	2	1P1U1S3D	2&1P.5&8S.	2&5U.	–

Table 9
MBSSA3D offers additional information to classify and order kinematic structures depending on their complexity and inclusion in a library. For example, a smaller number of SGs implied higher complexity and included SGs are ready for kinematic and dynamic analysis.

SG type	Num of SGs	SG on library
1R1P1U1S3D	1	0
1R3D	1	1
1P1U1S3D	3	1
2R1U1S3D	1	0
1R1U1S3D	1	1

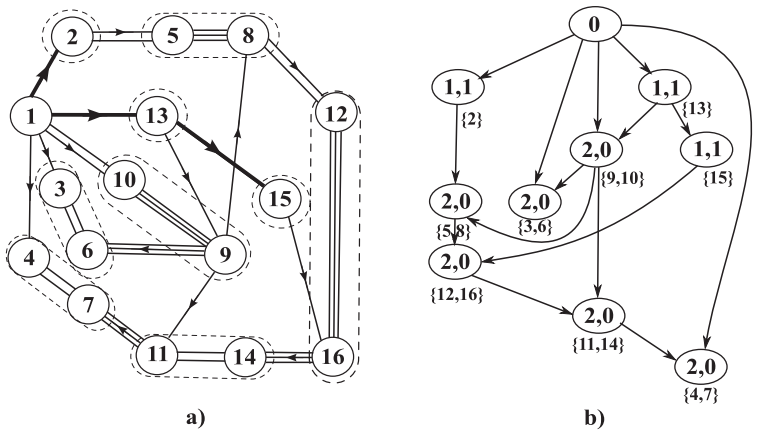


Fig. 14. Results for the structural transformation of the mechanism shown in Fig. 12.a with new input movements: KP(1-2), KP(1-13) and KP(13-15). a) Structural graph. (b) structural diagram.

From the group-equations topological approach, this mechanism may be modelled using the SGs gathered in Table 9. However, we see that some of the SGs: 1R1P1U1S3D and 2R1U1S3D are not included in the library (MBDSBEG), and then, their model and solution are not available. If there were no more options, we would need to generate the model and the closed-form or iterative solutions of these structural groups as shown in [6]. However, one of the main advantages of the developed computational MBA is that it may perform a structural transformation and classify and organise the different kinematic structures by simply modifying the knownInputs matrix.

If we consider the DOFs of the kinematic pairs KP(1-2), KP(1-13) and KP(13-15) as independent coordinates, the algorithm obtains the kinematic structure shown in Table 10. We validate this solution with the graph-analytical method in Fig. 14. This solution offers more SGs (simpler), and all of them are included in the library (Table 11), so we might use them to solve the kinematics and the dynamics of this multibody system. The interface program that would perform these analyses automatically is ongoing, but we may build the needed files manually, as introduced in [6].

$$Results_{SA} = \begin{pmatrix} 0 & 1 & 5 & 9 & 6 & 5 & 9 & 6 & 4 & 4 & 8 & 7 & 2 & 8 & 3 & 7 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \tag{13}$$

4.2. Micro-manipulator

Fig. 15.a shows a micro-manipulator for electronic microscopes, closed to the one presented in [20]. The original system has been modified for validation purposes and to avoid singularities as, for example, the redundant DOFs introduced by one body joined only by two spherical kinematic pairs.

Table 10
Structural groups obtained with input movements: KP(1-2), KP(1-13) and KP(13-15).

SGs	Num bodies	SG type	DOFs from Ext KP	Int KP	DOFs to Ext KP
2	1	1P3D	2&1P.	–	2&5U.
13	1	1P3D	13&1P.	–	13&9R.13&15R.
15	1	1P3D	15&13R.	–	15&16R.
9, 10	2	1R1U1S3D	9&13R.10&1U.	9&10S.	9&6S.9&8R.9&11R.
3, 6	2	1P1U1S3D	3&1P.6&9S.	3&6U.	–
5, 8	2	1R1U1S3D	5&2U.8&9R.	5&8S.	8&12C.
12, 16	2	1R1U1S3D	12&8U.16&15R	12&16S.	16&14S.
11, 14	2	1R1U1S3D	11&9R.14&16S.	11&14U.	11&7S.
4, 7	2	1P1U1S3D	4&1P.7&11S.	4&7U.	–

Table 11
Additional information of the kinematic structure. If all the obtained SGs are defined in the library, the multibody system might be solved.

SG type	Num of SGs	SG on library
1P3D	2	1
1R3D	1	1
1R1U1S3D	4	1
1P1U1S3D	2	1

Table 12
MBSSA3D Results for the 6 DOFs micro-manipulator. Input movements are defined over the linear actuators.

SGs	Num bodies	SG type	DOFs from Ext KP	int KP	DOFs to Ext KP
4,5,6,7	4	3R1C1ER3D	4&1ER.7&1R.	4&5C.5&6R.6&7R.	5&3S.5&21S.
2,3	2	1C1ER1S3D	2&1ER.3&5S.	2&3C.	–
10,11,12,15	4	3R1C1ER3D	12&1R.15&1ER.	10&11R.10&15C.11&12R.	10&9S.10&14S.10&21S.
8,9	2	1C1ER1S3D	8&1ER.9&10S.	8&9C.	–
13,14	2	1C1ER1S3D	13&1ER.14&10S.	13&14C.	–
21	1	2S3D	21&5S.21&10S.	–	21&18S
18,19,20	4	3R3S3D	20&1R.&18&21S.	18&19R.19&20R.	18&17S.
16,17	2	1C1ER1S3D	16&1ER.17&18S.	16&17C.	–

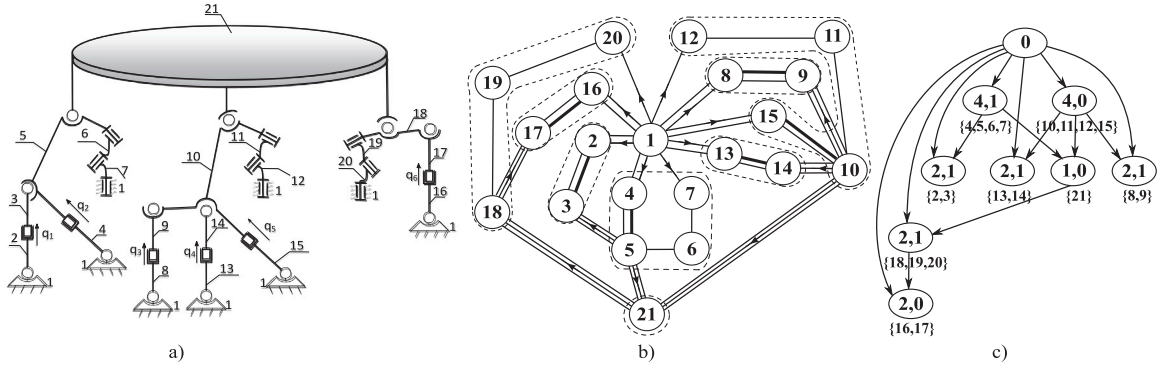


Fig. 15. (a) 6 DOF micro-manipulator mechanism extracted from [20]. (b) Structural graph of the micro-manipulator. (c) Kinematic structure using the graph-analytical method introduced in Section 2.

The mechanism has a platform (body 21) with three main branches divided into smaller ones. The input movements are defined on the six cylindrical joints as q_i (i = 1, …,6). Bodies 2, 4, 8, 13, 15, and 16 are joined to the frame with universal joints. Joints KP(3-5), KP(5-21), KP(9-10), KP(10-21), KP(10-14), KP(18-21) and KP(17-18) are spherical, and the rest are rotational joints.

With these input movements, the structural graph and the kinematic structure obtained using graph-analytical methods are shown in Fig. 15.(c) which coincides with the results offered by the algorithm in Table 12 and Eq. (14). Also, the results show that none of the obtained SGs is available in the library.

$$Results_{\mathrm{SA}} = \begin{pmatrix} 0 & 2 & 2 & 1 & 1 & 1 & 1 & 4 & 4 & 3 & 3 & 3 & 5 & 5 & 3 & 8 & 8 & 7 & 7 & 7 & 6 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \tag{14}$$

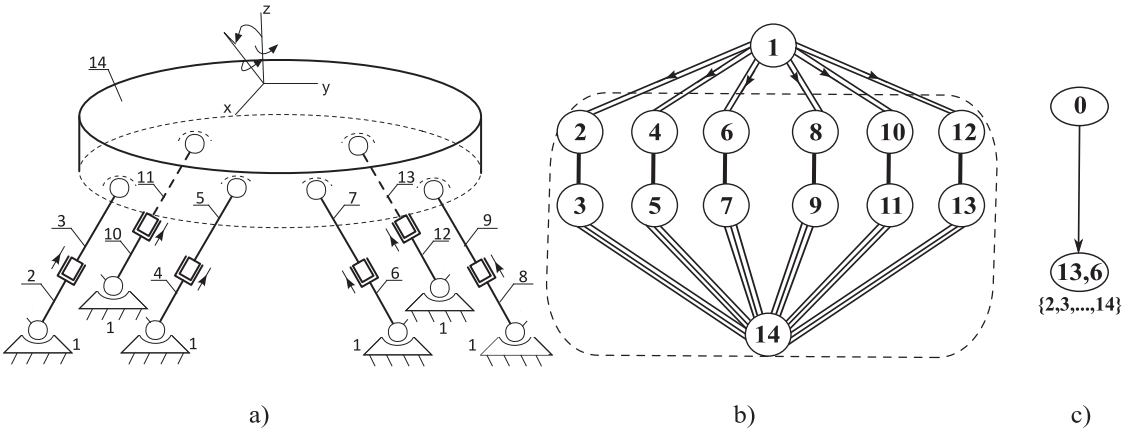


Fig. 16. (a) 6 DOFs Stewart's platform. (b) Structural graph. (c) Kinematic structure based on a graph-analytical method.

Table 13
Only one complex structural group is obtained in the original Stewart platform.

SGs	Num bodies	SG type	DOFs from Ext KP	
2,3,4,5,6,7,8,9,10,11,12,13,14	13	6P6U6S1F3D	2&1U.4&1U.6&1U.8&1U.10&1U.12&1U.14&1F. 2&3P.4&5P.6&7P.8&9P.10&11P.12&13P. 3&14S.5&14S.7&14S.9&14S.11&14S.13&14S.	... ...

Table 14
Structural Groups obtained with six input movements defining the position and orientation of the floating terminal KP(1-14). In (*) the given KP are: 14&3U.14&5U.14&7U.14&9U.14&11U.14&13U.

SGs	Num bodies	SG type	DOFs from Ext KP	Int KP	DOFs to Ext KP
14	1	1F3D	14&1F.	–	*
2,3	2	1P1U1S3D	2&1S.3&14U.	2&3P.	–
4,5	2	1P1U1S3D	4&1S.5&14U.	4&5P.	–
6,7	2	1P1U1S3D	6&1S.7&14U.	6&7P.	–
8,9	2	1P1U1S3D	8&1S.9&14U.	8&9P.	–
10,11	2	1P1U1S3D	10&1S.11&14U.	10&11P.	–
12,13	2	1P1U1S3D	12&1S.13&14U.	12&13P.	–

4.3. Stewart's platform

Another spatial multibody system of interest is the parallel kinematic manipulator based on Stewart's platform. It is a 6 DOFs high accuracy positioning mechanism with aerospace, nautical or medical applications. The six linear prismatic actuators with input movements KP(2-3), KP(4-5), KP(6-7), KP(8-9), KP(10-11) and KP(12-13) are joined to the terminal (body 14) and the frame with universal and spherical joints respectively (Fig. 16.a).

When structural analysis is performed using the MBSSA3D algorithm, only one complex structural group with thirteen bodies and six input movements forms the kinematic structure (Table 13 and Eq. (15)). This result coincides with the obtained using graph-analytical methods (Fig. 16.c).

$$Results_{SA} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix} \tag{15}$$

Again, we may specify in the MBSSA3D algorithm certain changes, as in the knowninputs matrix, to study new kinematic structures. For example, we might be interested in the path-planning and control of the terminal, which is considered a floating body connected to the frame with a 6 DOFs fictitious kinematic pair allowed in the algorithm. The inverse kinematic problem must calculate the displacements of the actuators in order to achieve the defined position and orientation of the terminal. The kinematic structure offered by the algorithm is shown in Table 14 and Eq. (16) which coincides with the one obtained using graph-analytical methods (Fig. 17.c). The structural transformation offers six structural groups, already defined in the library (Table 15), modelled with a reduced number of coordinates and high parallelism because, once the terminal is solved, these six SGs might be solved in parallel, as shown in the kinematic route 23, in Fig. 7.

Improvements in the efficiency of the group-coordinates approach, which is based on MBA for kinematic substructuring, have been reported in the literature using the simulation software PARCSIM (PARallel Computing SIMulator) [7].

Table 15
Only two types of SGs are needed and their kinematic and dynamic models are included in a library as MBDSBEG.

SG type	Num of SGs	SG on library
1F3D	1	1
1P1U1S3D	6	1

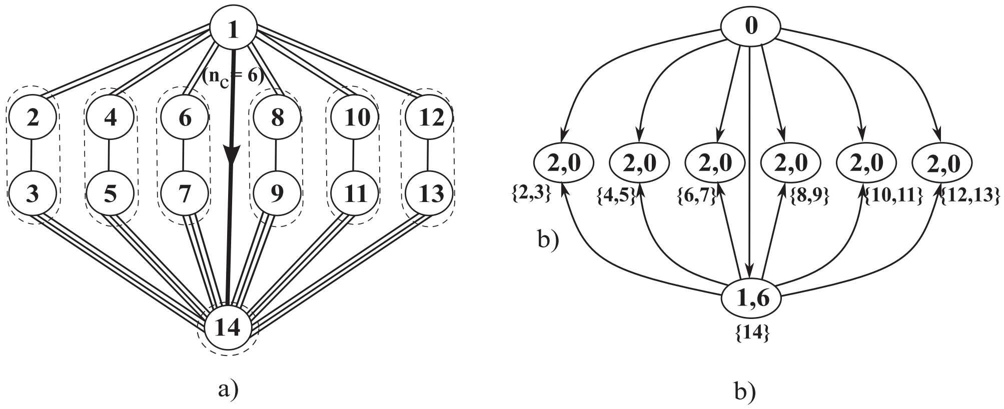


Fig. 17. Structural transformation for inverse kinematic problem of 6 DOFs Stewart platform. (a) Structural graph. (b) Structural diagram obtained using graph-analytical methods.

The first row in the results matrix assigns to each body (column) the number of SG it pertains to, which is a counter increased by the algorithm each time an SG is found. The dependency among the structural groups must be specified to define the kinematic structure of the multibody system. This information is obtained from the remaining rows in matrix $Results_{SA}$, as explained in Eq. (11). In the kinematic structure (Eq. (16)), we see these dependencies clearer than in the graph-analytical method (Fig. 17.b) because this matrix shows which body assigns DOFs from one SG to another body of a dependent SG. In our example, the frame (first column) assigns DOFs to bodies (rows) {2, 4, 6, 8, 10, 12 and 14} and the first obtained SG {14} (value '1' in the first row) provides the second assignment to bodies (rows) {3, 5, 7, 9, 11, 13}.

$$Results_{SA} = \begin{bmatrix} 0 & 2 & 2 & 3 & 3 & 4 & 4 & 5 & 5 & 6 & 6 & 7 & 7 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (16)$$

5. Robustness and efficiency of the new proposed approach

This section compares the robustness and the efficiency of the MBSSA3D approach introduced in this paper to the former version [21]. While the previous version was limited to planar multibody systems, the MBSSA3D deals with spatial ones; that is its main advantage. However, to give an idea about the efficiency of both approaches and compare their robustness, some tests have been launched on an Intel Core i7-8550U Quad Core 2.5 GHz CPU, RAM 8 GB DDR4 Microsoft Windows 10 Chinese Version, 64-bit OS, and Matlab 2020a programming environment. The most relevant information is shown in Table 16.

In the first test series, the kinematic structure of planar linkages with increasing complexity is obtained. The linkages have been automatically generated by adding $n_{AG} = 2, 4, 8$ Assur groups of a specific class ($k = \mathrm{I}, \mathrm{II}, \mathrm{III}$) to a 1-DOF group of primary elements. The first Assur group is joined to the primary element and the frame, and any added Assur group is aleatorily joined to the existing bodies. This process generates the linkages named: 'ASSUR k-n_{AG}' as, for example, ASSUR I-2, ASSUR I-4, ASSUR I-8, ASSUR II-2, or ASSUR III-4. More complex linkages are included as case studies, obtained as structural transformations of the original ones (e.g. ASSUR I-4), by considering the controlled DOF between the two last bodies, giving rise to the linkages with the suffix 'B'

Table 16
Number of structural groups SG_{SAXD} and computational cost t_{SAXD} (in seconds) for the cases studies launched to compare robustness and efficiency of the 2D [21] and the MBSSA3D computational algorithms.

Case study	Bodies	Joints	SG_{SA2D}	SG_{SA3D}	t_{SA2D} (s)	t_{SA3D} (s)
2D multibody systems						
ASSUR I-2	6	7	3	3	0.13409	0.90490
ASSUR I-4	10	11	5	5	0.10449	0.92285
ASSUR I-8	18	19	9	9	0.11734	0.98798
ASSUR I-2B	6	7	2	2	0.23736	0.84546
ASSUR I-4B	10	11	3	3	0.24463	0.92714
ASSUR I-8B	18	19	–	3	–	4.65035
ASSUR II-2	10	15	3	3	0.09860	0.91421
ASSUR II-4	18	27	5	5	0.21787	1.16003
ASSUR II-8	34	51	9	9	0.16228	2.33128
ASSUR II-2B	10	15	–	1	–	0.90840
ASSUR II-4B	18	27	–	2	–	5.39380
ASSUR II-8B	34	51	–	–	–	–
ASSUR III-2	14	21	3	3	2.82447	1.04206
ASSUR III-4	26	39	5	5	8.50998	5.22487
ASSUR III-2B	14	21	–	1	–	1.39049
ASSUR III-4B	26	39	–	1	–	70940.31
3D multibody systems						
Rotor Control (Fig. 12)	16	21	N/A	7	N/A	1.11330
Rotor Control (Fig. 14)	16	21	N/A	9	N/A	1.20191
Micromanipulator	21	28	N/A	8	N/A	1.15843
Stewart Platform (Fig. 16)	14	18	N/A	1	N/A	0.96645
Stewart Platform (Fig. 17) ST	14	18	N/A	7	N/A	0.80041

(e.g. ASSUR I-4B). In Table 16, the number of bodies and joints of the case studies, together with the number of structural groups (SG_{SA2D}, SG_{SA3D}) obtained and the computational cost (t_{SA2D}, t_{SA3D}) for the 2D (SA2D) and the 3D (SA3D) approaches are listed.

The tests for planar multibody systems with Assur groups of class I and II, with 2 to 8 Assur groups, show that when the structural groups obtained are of reduced size, the 2D version [21] is faster than the 3D proposed in this paper. However, when the structural groups increase complexity, as in case studies ASSUR III-2 and ASSUR III-4, the 3D version takes the advantage. Moreover, when the original algorithm searches for structural groups above ten bodies, it fails, as shown in all case studies with the 'B' suffix except in ASSUR I-2B and ASSUR I-4B.

The differences in computational cost between the original mechanisms and the corresponding transformed ones (with suffix 'B') are due to the exponential increase of time needed to check the condition of structural group of a kinematic chain, as its length (complexity) increases. For example, the case study ASSUR I-8 defines a mechanism formed by 18 bodies: the frame, one rotating crank (group of primary elements) and 8 Assur groups of class I (two bodies each). The kinematic structure of this case study offers nine structural groups: one group of primary elements and eight Assur groups of class I. These groups are reduced in size and easily found (the MBSSA3D algorithm solves this case study in 0.98798 s). However, suppose we change the position of the controlled DOF in this mechanism as indicated above: case study ASSUR I-8B. The kinematic structure of this structural transformation offers three structural groups only and is solved in 4.65035 s with the MBSSA3D algorithm. The 2D version cannot solve this case study which confirms an improvement in our searching algorithm.

Similarly, the case study ASSUR III-4 defines a mechanism composed of 26 bodies. Its kinematic structure offers five structural groups: one group of primary elements and four Assur groups of class III (six bodies each). Still, the algorithm finds these Assur groups in a reasonable time (5.22 s). However, the structural transformation ASSUR III-4B with the same topology has a kinematic structure formed by only one structural group of 26 bodies. The number of kinematic chain combinations of increasing length, from 1 to 26 bodies, that the search algorithm must verify is huge. The MBSSA3D algorithm finds a solution at a high computational cost, but the 2D version fails.

Multibody systems with kinematic structures based on a reduced number of highly complex structural groups are not very common. Still, even if the proposed algorithm MBSSA3D had to face the kinematic substructuring of such cases, it might succeed using different approaches: increasing the length of the kinematic chains, as demonstrated in the case studies or by studying new possible structural transformations.

The second test series is devoted to 3D multibody systems. The 2D version does not apply, and the MBSSA3D algorithm solves all the original cases and their structural transformations efficiently.

6. Conclusions

In this work, we discuss different approaches in the literature for kinematic substructuring of multibody systems, and compile some advantages of a method based on mobility criteria to methods based on linear graph theory. Despite its usefulness, computational algorithms for kinematic substructuring of spatial multibody systems based on mobility criteria are not available in the literature.

We have developed the MBSSA3D app for the computational structural analysis of spatial multibody systems with any number of degrees of freedom. The theoretical steps given in a graph-analytical method are introduced through the spatial crank–slider case study. The MBSSA3D architecture, functions and algorithms have been explained in detail as applied to that mechanism for a better understanding. A graphical user interface has been developed to help users perform kinematic substructuring.

One of the modules in MBSSA3D identifies and deals with certain types of singularities that may appear in multibody systems, known as exceptions to the Kutzbach–Grubler mobility criteria. We classify some of the exceptions and reveal the fundamentals of how they should be treated in computational kinematic substructuring. However, a detailed explanation of these algorithms and their application to different case studies with singularities is ongoing.

The MBSSA3D has been applied to several case studies, both planar and spatial. The results have been successfully compared to the literature and obtained using graph-analytical methods. In addition, one of the case studies has been compared to a computational method based on a linear graph algorithm.

The case studies showed practical benefits of the computational approach developed in this work, such as the ease generation of kinematic structures based on structural transformations with respect to the linear graph algorithm. A dedicated module has been developed to organise and classify kinematic structures from structural transformations under different criteria. The user may choose between maximising the number of structural groups, which implies more simplicity and efficiency in kinematic and dynamic analysis or maximise the coincidence between the obtained structural groups and those available in a particular library. In case of complete coincidence, as shown in some case studies and reported in the literature, the kinematics and dynamics of the multibody system may be directly solved from its kinematic structure. An ongoing interface module will automatically perform this task.

All the results offered by MBSSA3D are organised in matrices and tables for readability and analysis. We have shown that the modularity and re-usability of these modules may be exploited to find the optimal route in a tree of kinematic routes combining sequential and parallel execution of the modules. We have also demonstrated the robustness of the MBSSA3D algorithm with respect the 2D version as the complexity of the kinematic structure of the multibody system increases.

Future works should introduce the treatment of singularities and their application to different case studies. In addition, an interface module that reads the outputs generated by **MBSSA3D** should be completed for the automatic kinematic and dynamic modelling and analysis of multibody systems.

Another interface module should connect the obtained kinematic structures to the simulation software PARCSIM to exploit the parallelism capabilities of kinematic substructuring, optimising the computational cost of kinematic and dynamic simulations in a given hardware and software architecture.

Many structural transformations may appear in medium to large multibody systems, and the structural analysis of all of them would be very time-consuming. Therefore, the MBSSA3D app should let the analysts customise some preferences and restrictions to automatically generate the structural transformations of interest and obtain their kinematic structures. Analysing the most interesting topologies will reduce computational costs and achieve the best kinematic structures sooner. In addition, the algorithm should include a database to store all the results and facilitate consulting operations during the analysis or a long time after.

Finally, a new module should also perform the topological synthesis of multibody systems in future developments, as both synthesis and analysis problems are based on the same principle of mechanisms generation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The support of the Spanish Ministry of Economy and Competitiveness (MINECO) under project DPI2016-81005-P and the support of Spanish Ministry of Science and Innovation (MICINN) under project PID2020-120270GB-C21 are greatly acknowledged.

References

[1] A.A. Shabana, Dynamics of multibody systems, 2nd, Cambridge Univ Press, 1998.
[2] M. Gerardin, A. Cardona, Flexible Multibody Dynamics: A Finite Element Aproach, John Wiley & Sons, 2001.
[3] E.J. Haug, Computer-aided kinematics and dynamics of mechanical systems, Allyn and Bacon, Boston, MA, 1989.
[4] P.E. Nikravesh, Computer aided analysis of mechanical systems, Prentice Hall, Englewood Cliffs, NY, 1988.
[5] J. García de Jalón, Kinematic and Dynamic Simulation of Multibody Systems, Springer-Verlag, NY, 1993.
[6] M. Saura, P. Segado, D. Dopico, Computational kinematics of multibody systems: Two formulations for a modular approach based on natural coordinates, Mech. Mach. Theory 142 (2019) 1–22.
[7] J.-C. Cano, J. Cuenca, D. Giménez, M. Saura-Sánchez, P. Segado-Cabezos, A parallel simulator for multibody systems based on group equations, J. Supercomput. 75 (2018) 1368–1381.
[8] T. Kuroiwa, A. Motoe, H. Takahara, K. Hiraoka, Simulation system for the design of link mechanisms: Topological and kinematic analyses of mechanisms, JSME Int. J. 30 (268) (1987) 1652–1659.
[9] A. Kecskeméthy, On closed form solutions of multiple-loop mechanisms, in: Computational Kinematics, Springer Netherlands, Dordrecht, 1993, pp. 263–274.
[10] P. Fanghella, Kinematics of single-loop mechanisms and serial robot arms: A systematic approach, Meccanica (ISSN: 1572-9648) 30 (6) (1995) 685–705.
[11] A. Kecskeméthy, T. Krupp, M. Hiller, Symbolic processing of multiloop mechanism dynamics using closed-form kinematics solution, Multibody Syst. Dyn. 1 (1) (1997) 23–45.
[12] J.J. McPhee, On the use of linear graph theory in multibody system dynamics, Nonlinear Dynam. 9 (1) (1996) 73–90.

[13] K.T. Wehage, R.A. Wehage, B. Ravani, Generalized coordinate partitioning for complex mechanisms based on kinematic substructuring, Mech. Mach. Theory (ISSN: 0094-114X) 92 (2015) 464–483.
[14] G. Kinzel, C. Chang, The analysis of planar linkages using a modular approach, Mech. Mach. Theory 19 (1984) 165–172.
[15] M.R. Smith, Z. Ye, Simplified data structure for analysing mechanisms using character handling techniques, Comput. Aided Des. (ISSN: 0010-4485) 16 (4) (1984) 197–202.
[16] J.N. Hammar, Considering cost effective computer-aided mechanism design, Comput. Aided Des. (ISSN: 0010-4485) 19 (10) (1987) 534–538.
[17] P. Fanghella, C. Galletti, A modular method for computational kinematics, in: Computational Kinematics, Springer Netherlands, Dordrecht, 1993, pp. 275–284.
[18] M.R. Hansen, A general method for analysis of planar mechanisms using a modular approach, Mech. Mach. Theory 31 (8) (1996) 1155–1166.
[19] J.R.M. Ho, Higher-order kinematic error sensitivity analysis and optimum dimensional tolerancing of dyad and non-dyad mechanisms, University of Manitoba, 1997.
[20] M.Z. Kolovsky, A.N. Evgrafov, Y.A. Semenov, A.V. Slousch, Advanced theory of mechanism and machines, Springer, 2000.
[21] M. Saura, A. Celdrán, D. Dopico, J. Cuadrado, Computational structural analysis of planar multibody systems with lower and higher kinematic pairs, Mech. Mach. Theory 71 (2014) 79–92.
[22] T.S. Mruthyunjaya, Kinematic structure of mechanisms revisited, Mech. Mach. Theory 38 (4) (2003) 279–320.
[23] G.Z. Baranov, A course in the theory of mechanisms and machines, Mir, 1988.
[24] A. Evgrafov, D. Kozlikin, History of Mechanism and Machine Science, in: Distinguished Figures in Mechanism and Machine Science, Vol. 26, Springer, 2014.
[25] J. Buskiewicz, A method of optimization of solving a kinematic problem with the use of structural analysis algorithm (SAM), Mech. Mach. Theory 41 (2006) 823–837.
[26] H. Varbanov, T. Yankova, K. Kulev, S. Lilov, S&a - expert system for planar mechanism design, Expert Syst. Appl. 31 (31) (2006) 558–569.
[27] Q. Zeng, Y. Fang, Structural synthesis and analysis of serial-parallel hybrid mechanicsm with spatial multi-loop kinematic chains, Mech. Mach. Theory 49 (2012) 198–215.
[28] S.V. Shah, S.K. Saha, J.K. Dutt, Modular framework for dynamic modeling and analyses of legged robots, Mech. Mach. Theory 49 (2012) 234–255.
[29] W.A. Khan, V.N. Krovi, S.K. Saha, J. Angeles, Modular and recursive kinematics and dynamics for parallel manipulators, Multibody Syst. Dyn. 14 (3) (2005) 419–455.
[30] W.A. Khan, C.P. Tang, V.N. Krovi, Modular and distributed forward dynamic simulation of constrained mechanical systems - a comparative study, Mech. Mach. Theory 42 (5) (2007) 558–579.
[31] M. Koul, S.K. Saha, S.V. Shah, Reduced-order forward dynamics of multiclosed-loop systems, Multibody Syst. Dyn. 31 (4) (2014) 451–476.
[32] P. Fanghella, C. Galletti, G. Torre, An explicit independent-coordinate formulation for the equations of motion of flexible multibody systems, Mech. Mach. Theory 38 (5) (2003) 417–437.
[33] L. Romdhane, H. Dhuibi, H.B.H. Salah, Dynamic analysis of planar elastic mechanisms using the dyad method, Proc. Inst. Mech.Eng. Part K: J. Multi-Body Dyn. 217 (2003) 1–14.
[34] S.S. Kim, C.H. Lee, A recursive subsystem synthesis method for repeated closed loop structure in multibody dynamics, J. Mech. Sci. Technol. 23 (4) (2009) 946–949.
[35] A. Callejo, Y. Pan, J. Ricón, J. Kovecses, J. García de Jalón, Comparison of semirecursive and subsystem synthesis algorithms for the efficient simulation of multibody systems, J. Comput. Nonlinear Dyn. 12 (2016) 11–20.
[36] D. Dopico, F. González, A. Luaces, M. Saura, D.G. a Vallejo, Direct sensitivity analysis of multibody systems with holonomic and nonholonomic constraints via an index-3 augmented Lagrangian formulation with projections, Nonlinear Dyn. 93 (2018) 2039–2056.
[37] A. Sljoka, O. Shai, W. Whiteley, Cheking mobility and decompositions of linkages via peeble game algorithm, in: Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC, USA, 2011, pp. 1–10.
[38] M. Terushkin, S. Offer, Applying rigidity theory methods for topological decompositions and synthesis of gear trains systems, in: ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Chicago, IL, USA, 2012.