

A new method to approximate the field of movements of 1-DOF linkages with lower-pairs

Alvaro Noriega^a *, Jose Luis Cortizo^a, Eduardo Rodríguez^a, Ricardo Vijande^a, Jose Manuel Sierra^a

^a *Department of Mechanical Engineering, University of Oviedo*

Campus Universitario, Edf. Oeste, Módulo 5 s/n, 33203 Gijón, Spain

* Tel.: +34 985 18 19 25; Fax: +34 985 18 19 45; E-mail address: norieaalvaro@uniovi.es

Received: 5 February 2009 / Accepted: 21 January 2010 / Published online: 9 February 2010

Abstract

This paper presents a new method to obtain an approximation of the field of movements of a 1-DOF linkage with lower pairs. The method is based on a linkage representation by natural coordinates and the storage of the constraint equations by means of a sparse cubic matrix. To obtain a discrete approximation of the field of movements, a three-stage process is used. In the first stage, a special Evolution Strategy is applied to make the population converges towards the zones where the constraints error is minimal, obtaining, at the same time, a good distribution of individuals. In the second stage, the final individuals of the ES are used as initial points for a derivative algorithm to obtain a greater accuracy. Later, the third stage is a filtering process to eliminate individuals that represent non-desired solutions. This method has been tested on simple linkages with well-known fields of movements, generating comprehensive outcomes that justify the validity of the method.

Keywords: linkage, position problem, optimization, evolution strategy

Noriega A., Cortizo J.-L., Rodríguez E., Vijande R., Sierra J.-M., A new method to approximate the field of movements of a 1-DOF linkages with lower pairs. Meccanica 45, pages 681-692, 2010

DOI 10.1007/s11012-010-9283-2

The original publication is available at www.springerlink.com

<http://www.springerlink.com/content/0638615k6w325252/fulltext.pdf>

1 Introduction

The direct position problem of a linkage consists of calculating the position of all its elements when the value of the degrees of freedom (DOFs) is known [9]. Two approaches to pose this problem exist. The geometric approach, which is the most used, is based on considering every element of the linkage as an independent solid, obtaining its position in the space from the solution of the equations which are imposed by the geometric constraints [9]. The energy-approach is based on the finite elements and it applies finite elements techniques to solve the kinematic problems [3, 4]. This approach is not discussed in this paper.

In the case of the geometric approach, the equations that define the linkage are nonlinear. As a result, the direct position problem usually has several possible solutions and even, in some cases, it has no solution.

To solve the nonlinear system of equations which the geometric approach to the position problem generates, different methods can be used. Newton's method is a direct method which has quadratic convergence in the neighbourhood of each root and is easy to implement. This method needs the Jacobian matrix of the system to be known. In this method, it is very important to have a good starting point since the success of the algorithm depends on this [13]. Other approaches are indirect since they turn the problem of obtaining the roots of the system into a minimization problem of an error function that measures the non-fulfilment of the system's equations. Therefore, different optimization methods can be used. These methods can be classified according to the order of the derivatives that they use. Then, we can distinguish between the first order methods, which use the gradient's information of the function error and the second order methods, which also use the information of the Hessian matrix. All these methods always need a starting point and they provide only one of the multiple solutions that are possible for the direct position problem.

The possible solutions that the position of a linkage can have can come from two situations. The first situation is to place an element of the linkage in a different position without affecting the rest of the linkage. An example of this is the triangular element 2-3-5 in the four-bar linkage of the Fig. 1.a which can also be placed in position 2-3-5'. The other possible situation is the existence of more than one possibility of assembly of the linkage as is shown in Fig. 1.b.

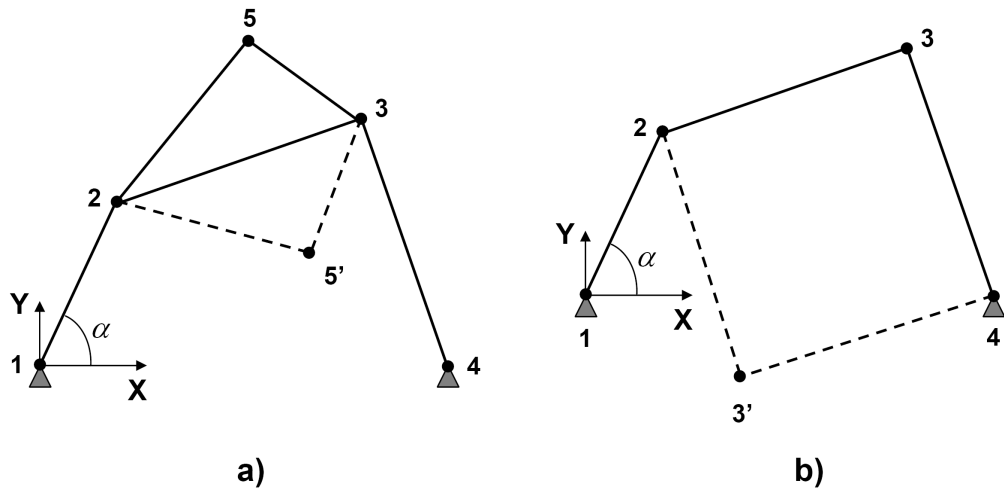


Fig. 1 Multiplicity of solutions in direct position problem

The multiplicity of solutions shown in Fig. 1.b generates different branches of operation in the linkage. These branches can cut each other causing singular configurations that complicate the linkage simulation. A classification of the types of singular configurations can be seen in [23] and [2]. In the singular configurations, some branches converge into one creating a multiple root of the system of equations. At these points, the Jacobian matrix of the equations's system is ill-conditioned and a method like the Newton one would lose accuracy. Moreover, there is the possibility that a non desired branch of movement is selected to continue the finite displacements problem.

For this reason, a method to obtain an approximation of the complete field of movements of a linkage without deforming its elements would be desirable. Furthermore, the method would be immune to the existence of singular configurations in this field.

The paper is organized as follows: In section 2, the modelization by means of natural coordinates of lower-pair linkages and its representation by means cubic matrix are shown. In section 3, the method to obtain a discrete approximation of the field of movements of a linkage is shown. In section 4, some examples of the use of the method are shown. Finally, some conclusions about the new representation of linkages and the method proposed to obtain its field of movements are shown in section 5.

2 Modelization and representation of linkages

The definition of a linkage can be made by a set of coordinates which will be the parameters that completely define the position of all its elements. Consequently, the variation of these parameters regarding the time describes the movement of the linkage [9].

The choice of these coordinates has great importance since it sets the modelization rules, the number and the complexity of the equations and other factors.

There exist different methods to modelize linkages. Some methods are based on finite elements [3, 4] and they show the advantage of a simple representation of the linkage by means of a geometric matrix. This matrix is derived from the stiffness matrix of the linkage if it were modelized as a truss and it condenses all the geometric information about the linkage.

Moreover, there exist the modelizations mentioned in [9] where the independent and dependent coordinates are described. The independent coordinates represent a classic approach for the linkages analysis while the dependent coordinates generate a system of nonlinear constraint equations that is better adapted to numerical computation. At the same time, the dependent coordinates can be classified as relative coordinates, reference point coordinates and natural coordinates.

The natural coordinates or fully Cartesian are an evolution of the reference point coordinates [8] where the reference points are moved to the pairs to avoid the use of angular variables to define the elements' orientation. They have been selected to this research because they have a simple and systematic definition of the constraint equations and a smaller number of coordinates and equations than the reference point coordinates.

2.1 Basic points and constraint equations

The modelization in natural coordinates has two basic steps. The first step is the definition of the linkage's elements by means of basic points and the second step is the construction of the constraint equations.

The basic points are situated in accordance with rules specified in [9] and their Cartesian coordinates will completely define the linkage position. To define a linkage in this particular application, the Cartesian coordinates of the basic points

will be exclusively used avoiding the use of angles. This eliminates the use of mixed coordinates because they generate trigonometric nonlinearities in the constraint equations and, as will be shown later, it is incompatible with the approach by means of cubic matrix representation.

The construction of the constraint equations is made in accordance with rules specified in [9] and can be extended both to planar linkage and to spatial ones. In Fig. 2, three examples of constraints equations in a planar linkage are shown.

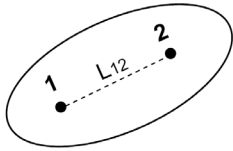
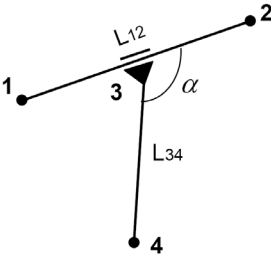
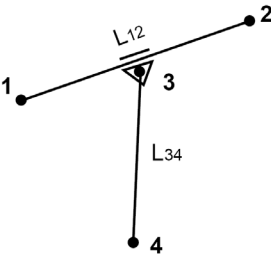
Rigid body condition	Rigid prismatic joint	Articulated prismatic joint
		
$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_{12}^2 = 0$	$(x_2 - x_1) \cdot (y_3 - y_1) - (y_2 - y_1) \cdot (x_3 - x_1) = 0$ $(x_2 - x_1) \cdot (x_4 - x_3) + (y_2 - y_1) \cdot (y_4 - y_3) - c = 0$	$(x_2 - x_1) \cdot (y_3 - y_1) - (y_2 - y_1) \cdot (x_3 - x_1) = 0$

Fig. 2 Examples of constraint equations in 2D

The posed constraint equations exclusively describe the linkage's geometry with lower pairs and they depend on a set of coordinates (stored in a vector \mathbf{x}) where the DOFs are included. Then, the system of constraint equations has more unknowns than equations and it is undetermined. In exchange, the solutions of this system are all the undeformed possible positions of the linkage, it means, the complete field of movements of the linkage.

It is worth noting that all the constraint equations in Fig. 2 are nonlinear but they have a common structure. They are quadratic and are composed with second order terms in which a constant multiplies a product of two variables and there can also exist independent terms. If trigonometric terms were included, there would not be a common and simple structure.

2.2 Cubic matrix representation

The common structure of the posed constraint equations allows the left side of each equation to be represented in the way shown in Fig. 3. In this figure, a multidimensional matrix is posed with as many dimensions as the maximum order of the products among variables (2 in the example). In each dimension, the

number of generated cells is the number of variables in the equation plus one cell more for the lower order terms (in the example in Fig. 3, these lower order terms are in the last row and column). The variables have to be ordered in a similar way in all the dimensions. In the cell corresponding to the intersection of a pair of certain variables, the factor that multiplies this product in the equation is stored. The area below the main diagonal is empty to indicate that it must not be used to avoid the data being duplicated due to the commutative property of the product.

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_{12}^2 = 0$$

↓

$$x_2^2 - 2 \cdot x_2 \cdot x_1 + x_1^2 + y_2^2 - 2 \cdot y_2 \cdot y_1 + y_1^2 - L_{12}^2 = 0$$

↓

	x_1	y_1	x_2	y_2	1
x_1	1	0	-2	0	0
y_1		1	0	-2	0
x_2			1	0	0
y_2				1	0
1					$-L_{12}^2$

Fig. 3 Representation of a constraint equation

When it is necessary to represent a system of equations, a multidimensional matrix with one more dimension than the matrix used to represent an equation is used. In this case, as the equations are quadratic, they are represented by means of a two dimensional square matrix and the system is represented by means of a three dimensional matrix or cubic matrix. The divisions of the additional dimension are called pages and they allow the coefficients of every equation to be represented. An example of it can be seen in Fig. 4 where only the coefficients different to zero are shown.

2.2.1 Storage

As can be seen in the example of Fig. 4, most of the cells of the cubic matrix contain a zero value. Therefore, it is not necessary to store the values of these cells, in a similar way to the finite element method (FEM). The difference with regard to that method is that in FEM it is only necessary to store the upper semi-band of a square stiffness matrix while in the cubic matrix representation the

elements different to zero can be in any position. For this reason, the sparse matrix storage technique is proposed. This technique lies in storing the position of the cell and its value in a two dimensional matrix of 4 columns (the three first columns for the position of the cell and the fourth for the value) and as many rows as cells whose content is not zero.

2.2.2 Evaluation

The evaluation of the constraint equations can be made in a direct way with the algorithm whose pseudocode is shown as follows:

- Input: **m** (Sparse cubic matrix with linkage data)
 x (Vector of coordinates)
- Output: **f** (Vector with the evaluations of equations' left side)
- Step 1: Initialization: Set the number c of rows of **m**. Initialize vector **f** to zero. Extend the vector of variables: $\mathbf{y} = [\mathbf{x}, 1]$
- Step 2: Main loop: For $i = 1$ to $i = c$

$$\mathbf{f}_{m_{i1}} = \mathbf{f}_{m_{i1}} + \mathbf{y}_{m_{i2}} \cdot \mathbf{y}_{m_{i3}} \cdot \mathbf{m}_{i4}$$

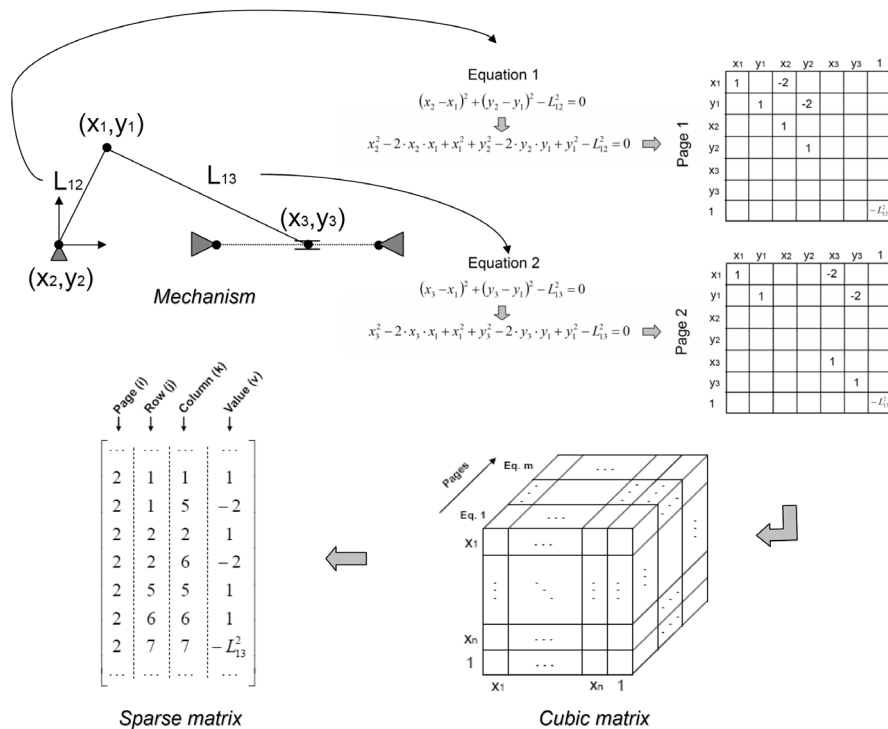


Fig. 4 Modelization, representation and data storage

2.2.3 Jacobian matrix calculation

The calculation of the Jacobian matrix is also direct, taking advantage of the proposed storage structure. The pseudocode of the algorithm that makes this calculation is shown as follows:

Input: **m** (Sparse cubic matrix with linkage data)
 x (Vector of coordinates)
 Output: **jac** (Jacobian matrix evaluated on **x**)

Step 1: Initialization: Set the number c of equations and the number n of variables. Initialize matrix **jac** to zero. Extend the vector of variables: $\mathbf{y} = [\mathbf{x}, 1]$

Step 2: Main loop: For $i = 1$ to $i = c$
 a) Extract the rows **ceq** of **m** corresponding to the equation i
 b) Calculate the number of rows nc of **ceq**
 c) For $j = 1$ to $j = n$

For $k = 1$ to $k = nc$

if $\mathbf{ceq}_{k2} = j$

$$\mathbf{jac}_{ij} = \mathbf{jac}_{ij} + \mathbf{ceq}_{k4} \cdot \mathbf{y}_{\mathbf{ceq}_{k3}}$$

end

if $\mathbf{ceq}_{k3} = j$

$$\mathbf{jac}_{ij} = \mathbf{jac}_{ij} + \mathbf{ceq}_{k4} \cdot \mathbf{y}_{\mathbf{ceq}_{k2}}$$

end

The Jacobian matrix is calculated with regard to every coordinate. Later, the columns corresponding to the constant coordinates have to be eliminated.

3 Method description

The system of m constraint equations that define the linkage can be expressed in the following way:

$$\Phi(\mathbf{x}) = \mathbf{0} \quad (1)$$

where \mathbf{x} is the vector of natural coordinates and each one of the m components of (1) is the value of the left side of each constraint equation.

The problem consists of finding the infinite solutions of the system (1) when the value of the DOFs in the formulation has not been introduced. This problem is especially interesting in 1-DOF linkages because these infinite solutions of the system represents the complete field of movements of the linkage, it means, every possible path of the linkage. In n-DOF linkages, like robots, these infinite solutions represent the workspace of the robot.

There exist different methods to approximate this set of solutions such as the continuation methods [1, 16] and the bifurcation methods [11, 12]. On the other hand, the optimization methods based on a populational approach like the genetic algorithm with operators that favour the distribution of the population [10], show a special adaptation to this problem. Therefore, it is proposed to reformulate the problem of obtaining the set of roots of (1) as a minimization problem of an error function which measures the non-fulfilment of the constraint equations:

$$error = \sum_{i=1}^m \Phi_j^2(\mathbf{x}) \quad (2)$$

It can be observed that the error has a similar structure to that of the least squares adjustment. To solve this problem, the Gauss-Newton method can be used. This method uses the knowledge of the Jacobian matrix to decide the search direction. This method is characterized by its efficiency [17]. It is also possible to use the Levenberg-Marquardt method [5, 15] in which the search direction is a cross between the Gauss-Newton direction and the direction of the gradient-descent method [21], being less efficient but more robust than the Gauss-Newton method. Both methods need the evaluation of the constraint equations and the Jacobian matrix of the system whose values can be easily computed with the representation indicated in Section 2.

The fact that there exist infinite solutions to the problem makes it has a great similarity with the problem of getting the Pareto optimal front by means of multiobjective evolutive algorithms [14]. These algorithms try to approximate this front by means of finite population \mathbf{P} of solutions \mathbf{x} (called individuals). The difference is that, in this case, the decision space has only one dimension (the error of each solution trying to fulfil the constraint equations) and the

homogeneous distribution of solutions must be obtained in the search space. This means that the posed problem does not fit any of the existent types of multiobjective problems since in this case the final aim is the population in the search space and not in the decision space.

Then, the proposed method tries to approximate all the existing branches of movement of a 1-DOF linkage in a simultaneous process by means of a population of individuals that are distributed the most homogeneously possible. Branch of movement refers to one the two possible situations indicated in Fig. 1.b. The method has three sequential stages:

1. Approximation

Its aim is to get a population as well-distributed as possible and where the individuals have the smallest possible error using a special Evolution Strategy (ES).

2. Refinement

Its aim is to reduce to the minimum the error of the population's individuals obtained in the first stage by means of the Levenberg-Marquardt least squares method.

3. Filtering

Its aim is to eliminate, by means of a filtering process, the individuals of the population which belong to branches of operation that produce non-desired configurations in some elements of the linkage or that represent linkages that are impossible to carry out in real applications.

3.1 Stage 1: Approximation

In this stage, an Evolution Strategy [20] of type (μ, μ) -ES is used. This ES is based on the Discrete Directions Mutation Evolution Strategy (DDM-ES) [19] and on the Hybrid Evolution Strategy (H-ES) [18].

This ES exclusively uses the mutation operator which is applied to every individual of the population (called parent), generating an offspring to replace itself, according to the following expression:

$$\mathbf{x}_{\text{offspring}} = \mathbf{x}_{\text{parent}} + v \cdot \mathbf{d} \quad (3)$$

where v is the mutation step that is randomly generated according to a Gaussian distribution of mean 0 and standard deviation σ (called mutation strength).

To decide the direction of the mutation of each individual (\mathbf{d}), an additional knowledge about the objective function is used since the gradient vector of the error function has the following expression:

$$\nabla_{error} = \left[\frac{\partial error}{\partial \mathbf{x}_1} \quad \dots \quad \frac{\partial error}{\partial \mathbf{x}_n} \right] \quad (4)$$

being

$$\frac{\partial error}{\partial \mathbf{x}_j} = \frac{\partial}{\partial \mathbf{x}_j} \sum_i \Phi_i^2(\mathbf{x}) = 2 \cdot \sum_i \Phi_i(\mathbf{x}) \cdot \frac{\partial \Phi_i(\mathbf{x})}{\partial \mathbf{x}_j} \quad (5)$$

The construction of the gradient vector depends exclusively on values of the evaluations of the equations and on the derivatives contained in the Jacobian matrix. These values are easily calculated by means of the sparse cubic matrix representation of the constraint equations shown in Section 2.

However, to get a distribution of individuals as homogeneous as possible, the direction of mutation in one individual (originally opposed to the gradient vector) is modified with the resultant of the repulsion forces that the rest of the individuals of the population exert on the individual that acts as parent. The resultant force has the following expression:

$$\mathbf{R}_i = \sum_{j \neq i} k \cdot \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|^\beta} \quad (6)$$

where the subscript i is applied to the parent and the subscript j is applied to the rest of individuals in the population. The constant k is called repulsion multiplier and it allows the repulsion force to be scaled while the constant β is called repulsion exponent and it controls the way in which the distance between individuals affects the repulsion force.

The unit vector that indicates the corrected mutation direction is constructed as follows:

$$\mathbf{d}_i = \frac{-\nabla_{error}_i + \mathbf{R}_i}{|-\nabla_{error}_i + \mathbf{R}_i|} \quad (7)$$

An example of the process described above can be seen in Fig. 5.

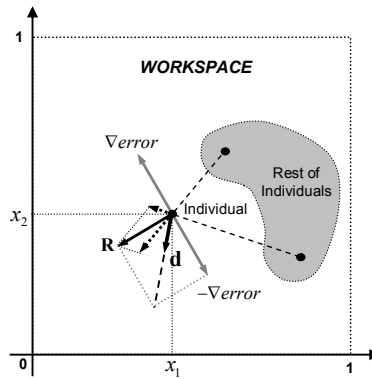


Fig. 5 Graphical example of repulsion forces and direction of mutation in 2 dimensions

Due to the fact that every variable in the search space cannot have the same range of variation, it is proposed to normalize all the ranges to the range $[0,1]$ forming an auxiliary space called *workspace* where evolution process happens. When the individuals need to be evaluated, a scaling process has to be done to carry them to the search space. The expression for this process is the following:

$$\mathbf{x}_k = \mathbf{l}_k + (\mathbf{u}_k - \mathbf{l}_k) \cdot \mathbf{x}'_k \quad (8)$$

where \mathbf{x} is the vector that represents the individual in the search space, \mathbf{x}' is the vector that represents the individual in the workspace, \mathbf{l} is the vector that contains the lower bound of the variables and \mathbf{u} is the vector with the upper bound of the variables. Moreover, as the unit vector \mathbf{d} refers to the search space, it must be also normalized to the workspace in order to use it in the mutation.

The normalization of the individuals to the workspace is also used to set a single common mutation strength σ . This mutation strength has a dynamic control which decreases the value of σ with the number of generations according to the next expression:

$$\sigma = s \cdot c_1 + (\sigma_{ini} - s) \quad (9)$$

being

$$s = \frac{\sigma_{end} - \sigma_{ini}}{g - 1} \quad (10)$$

where c_1 is the number of the current generation, g is the number of total generations to make, σ_{ini} is the mutation strength to apply in the initial generation and σ_{end} is the mutation strength to apply in the final generation. The mutation strength diminishes in a linear way with the generations.

The selection operator has no sense in this ES because it is desired that the whole population converges towards the solution, covering it with a homogeneous distribution. It does that this ES really works as μ evolution strategies of type (1,1)-ES together.

There also exists an auxiliary operator which generates independent individuals to complete the population if the offspring is situated out of the workspace (repair operator). These independent individuals are randomly generated by the repair operator according to a uniform distribution in the workspace.

The flowchart of this Stage can be seen in Fig. 6 and the following explanations must be given:

- The initial population is randomly generated according to a uniform distribution in the workspace. The population size μ is defined by the user.
- The evaluation of the population entails the previous scaling process of the individuals to the search space and it includes the calculation of the error made on the fulfilment of the constraint equations and the calculation of its gradient.
- c_1 counts the number of generations executed by the algorithm. The total number of generations g is previously defined by the user.
- The final mutation strength σ_{end} , the repulsion multiplier k and the repulsion exponent β are defined by the user.
- c_2 counts the number of individuals of the population on which the offspring, which replaces its parent, has already been generated.
- The repair operator acts when the offspring generated by mutation does not belong to the workspace and it replaces the parent with an independent individual.

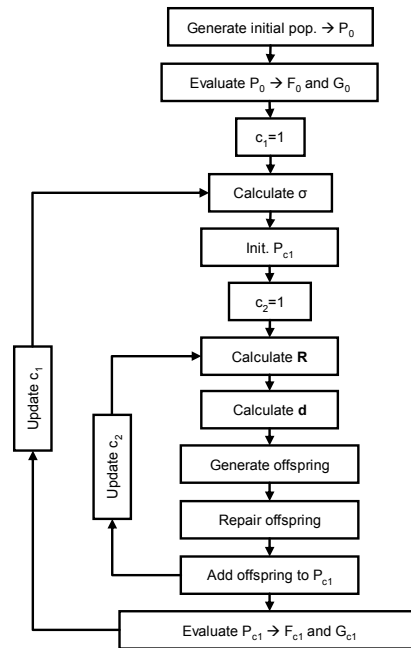


Fig. 6 Flowchart of Stage 1 algorithm

3.2 Stage 2: Refinement

In this stage, the individuals of the final population of Stage 1 are scaled to the search space and used as initial points to a least squares adjustment algorithm. This algorithm carries out a minimization process on the sum of least squares of the evaluation of constraint equations shown in Eq. (2), applying the Levenberg-Marquardt method with line search by means of mixed quad-cubic interpolation [15].

This algorithm demands the knowledge of the Jacobian matrix of the system of constraint equations which is easily computed by means of the sparse cubic matrix representation shown in Section 2.

The algorithm used converges towards the point of the set of solutions closest to the initial point and as the initial approximations are well-distributed, a quite homogeneous covering of the set of possible positions of the linkage is obtained.

3.3 Stage 3: Filtering

The refined population approximates the whole set of mathematical solutions which minimizes the error. If the elements of the linkage do not suffer deformations, the associated error to this configuration is 0. However, it is possible that with certain dimensional parameters, the linkage has no undeformed

solution and then the method converges to the solution with smallest error in the constraint equations (that is not the least deformed solution as is shown in [4]). In this case, it is necessary to introduce a filtering process to identify or eliminate these types of solutions.

Moreover, it can also be desirable to eliminate the configurations of the linkage where some elements (defined by three points) have a specific orientation or that the mobile point in a slot keeps itself between the two edges that define the slot.

All these filtering criteria can be used to adjust the output of the method to the user's wishes and to facilitate its utilization.

3.4 Parameter setting

The population size μ is chosen by taking into account the complexity expected in the field of movements of the linkage. The greater the expected number of branches of movement, the bigger must be the population size. When a better definition of the movement's branches is desired, the population size must also be increased. However, great population sizes slow down the speed and even stop the convergence. The reason for this behaviour is that in the initial stage, the repulsion forces come from every direction and their effect are almost cancelled. When the population has a certain clustering, the repulsion forces are very strong and they stop the convergence. The experimental tests carried out showed that μ values between 300 and 2000 produce good results.

The number of total generations g is chosen to make the individuals of the final population have a small enough error and, therefore, the refinement process will be as fast as possible. Obviously, the greater the number of total generations, the smaller the error of the individuals. However, an unjustified increase in g makes Stage 1 be too long, without obtaining practical improvements. In experimental tests, 100 generations have been used obtaining good results.

The mutation strength for the initial generation σ_{ini} is calculated with the following expression:

$$\sigma_{ini} = \frac{\sqrt{n}}{2} \cdot \left(\frac{1}{\mu}\right)^{\frac{1}{n}} \quad (11)$$

where n is the number of variables. This expression is similar to the one used in DDM-ES [19]. The choice of the mutation strength for the final generation σ_{end} depends on the accuracy desired. The value of σ_{end} must be smaller than σ_{ini} . However, a very small value of σ_{end} can slow down the convergence of the ES in the first Stage. In the tests carried out, and with 100 generations, a value of σ_{end} approximately 100 times smaller than σ_{ini} generates good results.

The value of the repulsion multiplier k depends on the population size. If there is a big population size, a small value of k must be chosen to avoid the repulsion hindering the algorithm convergence and vice versa. In the tests carried out, values between 1 and 0.01 were used depending on the problem. For the repulsion exponent β , the value 1 is usually chosen although in some cases it is observed that with a value of 2, better results are obtained but it depends on the number of variables and on their ranges in the search space since both values define the magnitude of the distance between individuals.

4 Examples

4.1 Four bar linkage

One of the most studied planar linkages is the four bar linkage. In spite of being a very easy linkage, depending on the dimensions of its elements, it can produce some singular configurations that hinder its simulation. In this example it is proposed to study all the possible paths of the point 5 in the coupler shown in Fig. 7.

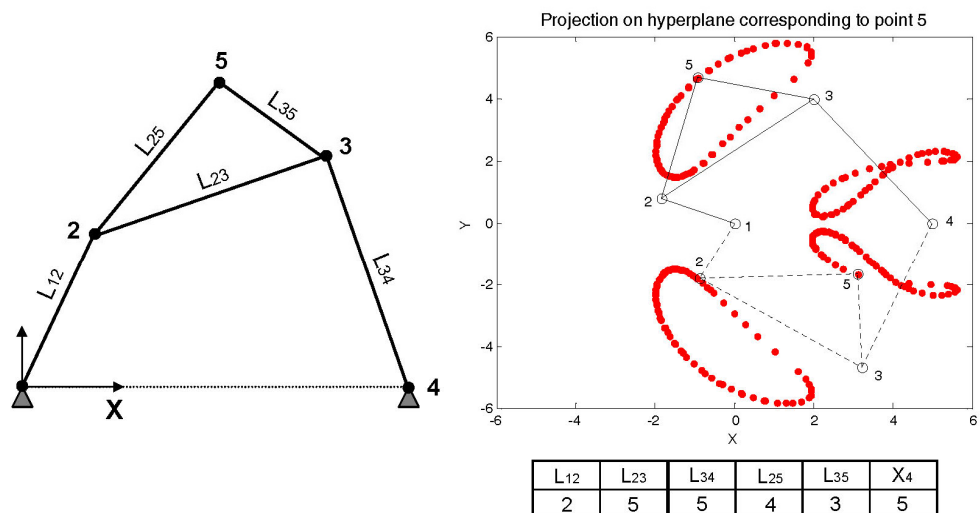


Fig. 7 Four bar linkage and discrete approximation of its field of movements

In the first example, the dimensional parameters shown in Fig. 7 are used and the space of movement for the linkage has the following limits:

$$x \in [-10,10]$$

$$y \in [-10,10]$$

This space of movement determines the lower and upper limits of each variable to be optimized in the Stage 1 of the proposed method.

The population size is 300 and it evolves during 100 generations with a σ_{end} of 0.005. The value of the repulsion multiplier k is 1 and the repulsion exponent β is 2.

In Fig. 7, the set of 300 refined solutions can be seen projected on the hyperplane corresponding to the coordinates of point 5. It is also possible to see the configurations corresponding to two different solutions. The set of undeformed possible solutions is symmetric with regard to the X axis and the four different possible paths of point 5 do not cut each other avoiding the appearance of singular configurations. A path can cut itself and this situation does not produce a singular configuration.

To check the behaviour of this method in the presence of singular configurations, the dimensional parameters of the linkage are modified to construct a parallelogram. The new parameters can be seen in the corresponding table in Fig. 8.a. The parameters of the method for the Stage 1 will be the same as in the last example.

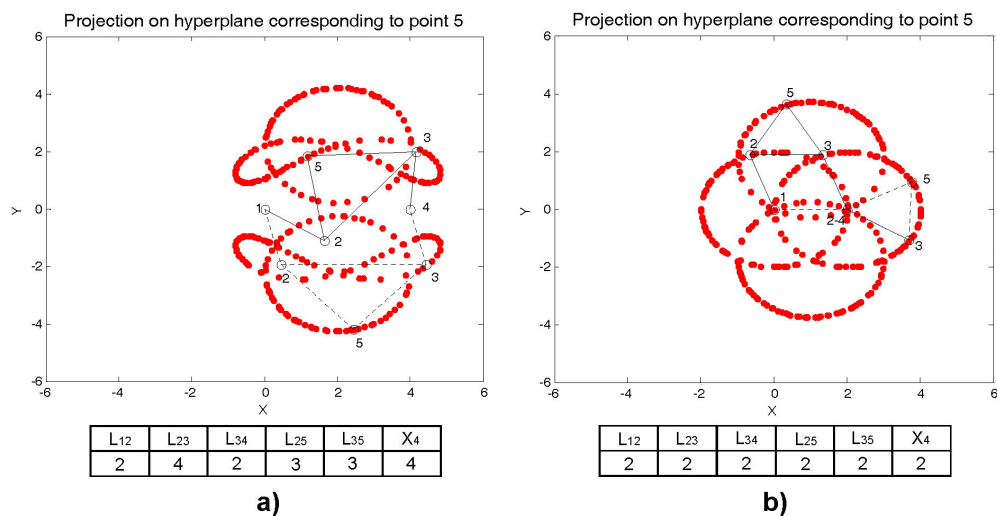


Fig. 8 Discrete approximation of the field of movements of 4-bar linkage with different dimensions

In Fig. 8.a, it can be noticed that the two possible paths of point 5 (one path is a circumference and the other path is a loop) cut each other at four points (ignoring the fact that there are two symmetric paths with regard to the X axis because they correspond to two different orientations of the element 2-3-5). However, only two of these points (diametrically opposed) concern singular configurations.

To safely distinguish what cutting points correspond to singular configurations, an order one analysis [2, 6, 22] or an order two analysis [7] can be done.

But this problem can be more complicated if the coupler has an equilateral triangle shape with side length equal to all the bars on the parallelogram. This is the case shown in Fig. 8.b. The parameters for Stage 1 of the method are the same as in the two last examples.

The refined population of this last case shows that there are four possible paths of point 5 and they have a circular shape, cutting each other at eight points. However, only six points concern singular configurations of this linkage.

Furthermore, though only the variables of point 5 are shown, each individual x contains the positions of every basic point that defines the linkage and, therefore, the configuration associated to this individual is completely defined.

4.2 Stephenson I linkage

Other well-known linkages are the Stephenson linkages (I, II y III). In this example, it is proposed to study all the possible paths of the point 7 of Stephenson I linkage shown in Fig. 9.

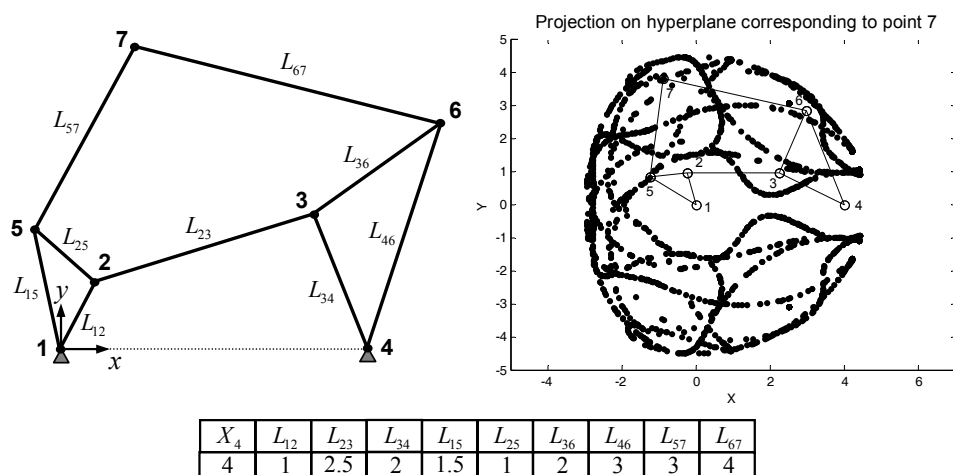


Fig. 9 Stephenson I linkage and discrete approximation of its field of movements
The space of movement for the linkage in Stage 1 has the following limits:

$$x \in [-5, 7]$$

$$y \in [-5, 5]$$

The population size is 2000 and it evolves during 100 generations with a σ_{end} of 0.007. The value of the repulsion multiplier k is 1 and the repulsion exponent β is 2.

In Fig. 9, the set of 2000 refined solutions has been projected on the hyperplane corresponding to the coordinates of point 7. In this case, the set of undeformed possible solutions is also symmetric with regard to the X axis and it includes all the possible configurations of the linkage. To make the distinction between the paths corresponding to different configurations easier, a subsequent filtering process, taking into account the user's additional criterion, may be done.

It is interesting to realize that the four bar linkage 1-2-3-4 does not meet the Grashof condition for the bar 1-2 to do complete rotations. For this reason, all the possible paths of point 7 have an abrupt ending that can be observed on the right of Fig. 9.

Conclusions

At first, the modelization and representation of the geometry of the lower-pair linkages (both planar and spatial) based on natural coordinates is posed. This method uses Cartesian coordinates exclusively and it does not use angles. This feature allows the constraint equations to have a common structure of a quadratic type and it allows their storage in a sparse cubic matrix. The advantages of this representation are that it allows direct evaluation of equations and their Jacobian matrix by means of very efficient algorithms.

Based on this linkage modelization, a method to approximate the field of movements of a 1-DOF linkage is posed. This method has three sequential stages, the first two (approximation and refinement) being the most important, while the third stage (filtering) allows a part of the field of movement to be selected to study it.

This method tries to approximate the infinite positions that the linkage has when its DOF is free by means of finite population of solutions. These infinite positions represent the workspace of the 1-DOF linkage if it were considered as a robot.

In the field of possible movements of a linkage, singular configurations can exist where the linkage has several possible branches to continue its movement. This kind of singular configurations is identified as the cross among two or more branches of operation.

It is also possible that there are configurations where there exist an instantaneous increase of DOFs in the linkage. This type of configuration generates new branches of operation for a part of the linkage.

Finally, there are singular configurations in which the linkage is blocked. This type of configurations is identified as the end of a branch of operation of the linkage.

These types of singular configurations are usually studied in the case of the workspace of robots and manipulators but they do not arouse much attention in the case of 1-DOF linkages.

The method proposed is immune to the existence of these singular configurations and it facilitates the location of some of them by means of the visualization of the complete field of movements of the linkage.

Another possible utility of the complete field of movements of a 1-DOF linkage is in dimensional synthesis of path generation. If you have a linkage with certain dimensions, it is possible to obtain, in one go, all possible paths for the point studied and, then, search for the one that best approximates to the path desired. Furthermore, the type of configuration can be discriminated by means of a filtering process. For instance, it is possible to define the orientation which every part of the linkage must have.

Another advantage of the method proposed is that it allows the position problem to be completely disconnected from the velocity and acceleration problems since the field of movements of a linkage does not depend on the time or on the initial conditions of the simulation as occurs in the outcomes derived from the kinematic simulation approaches.

This method has been tested on two different well-known linkages, the four bar linkage and the Stephenson I linkage, confirming that the method is immune to the existence of the different types of singular configurations cited before and representing a new valid tool to detect the complete workspace of a 1-DOF linkage.

References

- [1] Allgower EL, Georg K (1980) Simplicial and Continuation Methods for Approximations, Fixed Points and Solutions to Systems of Equations. SIAM Rev 22: 28-85
- [2] Altuzarra O, Pinto C, Avilés R, Hernández A (2004) A practical procedure to analyze singular configurations in closed kinematic chains. IEEE T Robot 20 (6): 929-940
- [3] Avilés R, Ajuria MBG, Hormaza MV, Hernández A (1996) A procedure base on finite elements for the solution of nonlinear problems in the kinematic analysis of mechanisms. Finite Elem Anal Des 22: 305-327
- [4] Avilés R, Hernández A, Amezua E, Altuzarra O (2008) Kinematic analysis of linkages based in finite elements and the geometric stiffness matrix. Mech Mach Theory 43 (8): 964-983
- [5] Dennis JE Jr (1977) Nonlinear least-squares. State of the Art in Numerical Analysis. Academic Press: 269-312
- [6] Di Gregorio R (2007) A novel geometric and analytic technique for the singularity analysis of one-dof planar mechanisms. Mech Mach Theory 42: 1462–1483
- [7] Fernández de Bustos I, Agirrebeitia J, Avilés R, Ajuria G (2009) Second order analysis of the mobility of kinematic loops via acceleration compatibility analysis. Mech Mach Theory 44 (10): 1923-1937
- [8] García de Jalón J (2007) Twenty-five years of natural coordinates. Multibody Syst Dyn 18 (1): 15-33
- [9] García de Jalón J, Bayo E (1994) Kinematic and dynamic simulation of multibody systems. The real-time challenge. Springer-Verlag, New York
- [10] Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, New York
- [11] Golubitsky, Schaeffer (1985) Singularities and Groups in Bifurcation Theory Vol. 1. Springer-Verlag
- [12] Golubitsky, Stewart, Schaeffer (1988) Singularities and Groups in Bifurcation Theory Vol. 2. Springer-Verlag
- [13] Grosan C, Abraham A (2008) A new approach for solving nonlinear equations systems. IEEE T Syst Man Cy A, 38 (3): 698-714
- [14] Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: A tutorial. Reliab Eng Syst Safe 91: 992–1007
- [15] Moré JJ (1977) The Levenberg-Marquardt Algorithm: Implementation and Theory. Lect Notes Math 630: 105-116
- [16] Morgan A (1987) Solving polynomial systems using continuation for engineering and scientific problems. Prentice-Hall, Englewood Cliffs, NJ
- [17] Nocedal J, Wright SJ (1999) Numerical Optimization. Springer Series in Operations Research, Springer Verlag, New York
- [18] Noriega A (2008) Síntesis dimensional óptima de mecanismos mediante estrategias evolutivas. PhD. Thesis, University of Oviedo

- [19] Noriega A, Rodríguez E, Cortizo JL, Vijande R, Sierra JM (2008) A new evolution strategy for the unconstrained optimization problem. Proceedings of Second International Conference on Multidisciplinary Design Optimization and Applications, 3-5 September 2008, Gijón, Spain
- [20] Schwefel HP (1995) Evolution and Optimum Seeking. John Wiley & Sons, INC
- [21] Snyman JA (2005) Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms. Springer Publishing, New York
- [22] Yang D-C, Xiong J, Yang X-D (2008) A simple method to calculate mobility with Jacobian. Mech Mach Theory 43: 1175–1185
- [23] Zlatanov D, Fenton RG, Benhabib B (1998) Identification and classification of the singular configurations of mechanisms. Mech Mach Theory 33 (6): 743-760