

# Optimal synthesis of planar mechanisms via an extensible-link approach

J.-F. Collard<sup>1</sup>, P. Duysinx<sup>2</sup>, P. Fiset<sup>1</sup>

<sup>1</sup>Université Catholique de Louvain (UCL), Center for Research in Mechatronics  
Place du Levant 2, 1348 Louvain-la-Neuve

<sup>2</sup>Université de Liège (ULg), Department of Mechanics, Ground Vehicles Research Group  
Chemin des Chevreuils 1, 4000 Liège  
e-mail: [jf.collard@uclouvain.be](mailto:jf.collard@uclouvain.be)

## Abstract

This paper presents a novel approach to optimize the design of planar mechanisms with revolute joints for function-generation or path synthesis. The proposed method is based on the use of an extensible-link mechanism model whose strain energy is minimized to find the optimal rigid design. This enables us to get rid of assembling constraints and the use of natural coordinates makes the objective function simpler. The optimization strategy is divided into two stages: the first one relies on multiple *partial* optimizations and provides hot starting point for the second stage which involves all the variables and all the energy contributions. The question of finding the global optimum is reviewed. Instead, a simple algorithm is proposed to explore the design space and to find several local optima among which the designer may choose the best one taking other criteria into account (e.g. stiffness, collision, size,...). Two applications are presented to illustrate the whole process.

## 1 Introduction

Optimization of complex multibody systems represents a real present interest along with the increasing development of computer resources. This is particularly true considering closed-loop mechanisms whose assembling constraints represents a particular issue when evolving the optimization process strategy. A few solutions have been proposed to deal with them [13]. For example, the authors have suggested to penalize properly the objective function using the conditioning of the assembling constraints Jacobian matrix [4]. Another well-known approach in path synthesis is to deform the mechanism subject to a perfect following of the desired path [8, 1, 2]. From this point of view, the path-following objective becomes a trivial optimization constraint while the deformation energy is the actual objective to minimize. Therefore, the mechanism assembles at best each time the optimization process computes the objective function.

In optimal design synthesis, a second issue consists in the choice of the formalism to describe the geometry of the mechanism. Among the different possibilities, one can mention the common use of relative coordinates in real form [3, 6, 10] or in complex form [16]. This formalism has the advantage to limit the number of assembling constraints but introduces trigonometric functions involving angular variables: it enhances the non-linearity of the problem and makes the optimization more complex. The use of natural – or point – coordinates is also wide-spread [8, 1, 2, 11, 7]. In comparison with relative coordinates, natural coordinates involve additional algebraic constraints. However, these equations only consist in linear and/or distance functions. This coordinate system is thus well suited to the use of gradient-based optimization techniques such as least squares methods.

The proposed method tries to combine these two features: extensible-link mechanisms and natural coordinates. The first one enables to solve the problem of non-assembly while the second one greatly simplifies the type of objective function. The associated optimization strategy is divided into two parts. The first one

is based on the minimization of the deformation energy over the followed path [6]. The result of these first multiple *partial* optimizations is then used as starting point for the second part. Then, the *total* strain energy, i.e. the sum of all the *partial* energy functions, is optimized not only with respect to some point coordinates but also to the design parameters themselves [1].

Afterwards, this improvement has enabled to outline an important issue in mechanism optimization: different local optima starting from different initial parameters. The choice of the optimal mechanism among these local optima relies on other design constraints which may be more difficult to compute and not taken into account in the original problem. Since it is interesting for the designer to keep and compare some of the best mechanisms (i.e. local optima), an exploration strategy of the design space is proposed to 'unearth' most of the possible optima.

Different kinds of requirements may be encountered in dimensional mechanisms synthesis: path or function generation, body guidance, or mixed problems. Most applications concern path synthesis problems [13, 3, 6, 10, 16, 12, 18] and the four-bar mechanism will constitute a basic example in the following. More realistic applications of function-generation synthesis will also be given based on the Ackerman steering linkage problem: a four-bar and then a six-bar synthesis [20, 15, 19].

The paper is organized as follows. In Section 2, the general optimization problem is modeled and formulated using the four-bar mechanism as example: the objective function is build and the sensitivity analysis is performed. In Section 3, the optimization strategy is developed. Section 4 presents a more realistic application of function generation synthesis for the four-bar Ackerman steering linkage. Section 5 deals with the question of finding the global optimum following by a new method to explore the design space. Before some conclusions and prospects in Section 7, a more complete application to optimize a six-bar steering linkage is presented to illustrate the concepts in Section 6.

## 2 Problem formulation

Let us consider the well-known planar example of a four-bar mechanism which has to follow a desired path (see dotted line in Fig. 1.a). In order to make the mechanism exactly follow the given path, the four-bar is modeled with extensible links which replace the rigid bars and triangle by five springs with stiffness  $k_j$  and natural lengths  $l_j$ ,  $j = 1 \dots 5$  (see Fig. 1.b).

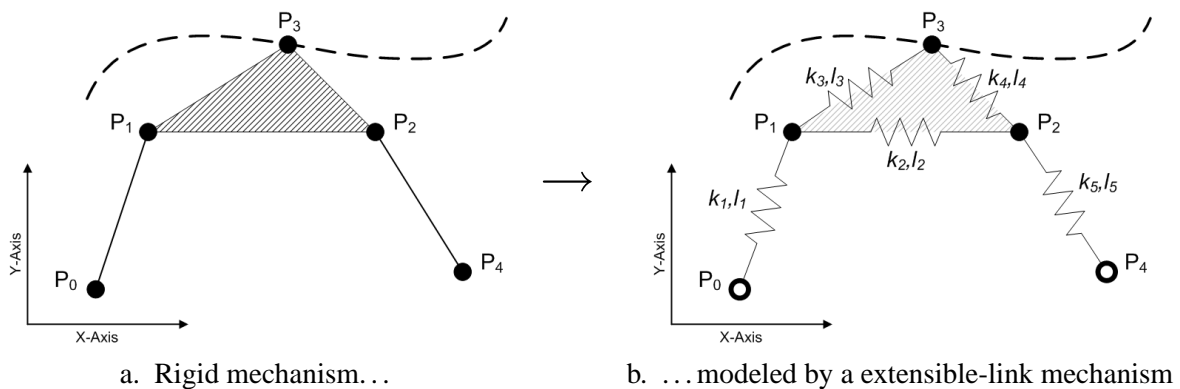


Figure 1: Model adaptation of four-bar mechanism for path synthesis

The desired path is discretized into  $N$  points, leading to  $N$  different configurations of the mechanism. When it moves, the different points  $P_0 \dots P_4$  composing the mechanism have different behaviors:  $P_0$  and  $P_4$  stay fixed to the ground,  $P_3$  follows exactly the  $N$  points composing the path and  $P_1$  and  $P_2$  are free to reach the equilibrium. All these points can thus be arranged into three groups:

- the *static* points  $P_0, P_4$ ;

- the *tracking* point  $P_3$ ;
- the *floating* points  $P_1, P_2$ .

Their absolute coordinates are saved respectively in the following vectors:  $\mathbf{s}$ ,  $\mathbf{t}$  and  $\mathbf{f}$ . As the tracking point and the floating points may have different coordinates for each configuration,  $\mathbf{t}$  and  $\mathbf{f}$  are referenced by the index  $i$ :  $\mathbf{t}_i$  and  $\mathbf{f}_i$ ,  $i = 1 \dots N$ .

Grouping the natural lengths  $l_j$  in the column vector  $\mathbf{l}$  and the stiffness parameters  $k_j$  on the diagonal of the stiffness matrix  $\mathbf{K}$ , we define the total strain energy as a scalar cost function:

$$E(\mathbf{s}, \mathbf{t}_1, \dots, \mathbf{t}_N, \mathbf{f}_1, \dots, \mathbf{f}_N, \mathbf{l}, \mathbf{K}) = \frac{1}{2} \sum_{i=1}^N (\mathbf{d}_i - \mathbf{l})^T \mathbf{K} (\mathbf{d}_i - \mathbf{l}), \quad (1)$$

where index  $i$  stands for the  $i^{th}$  configuration of the mechanism and  $\mathbf{d}_i$  is a column vector containing the five distance  $d_j^i$  computed between each pair of linked points. For the moment, the only known parameters are the  $2N * 2$  coordinates of the floating points. The stiffness parameters  $k_j$  may be chosen by the user. They play the role of weights in the sum of all the contributions to the total energy. Making a bar stiffer increases its relative importance in the cost function but this possibility is not used in the following. After these considerations, the optimization problem is stated as follows:

$$\min_{\mathbf{s}, \mathbf{f}_1, \dots, \mathbf{f}_N, \mathbf{l}} \frac{1}{2} \sum_{i=1}^N [\mathbf{d}(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i) - \mathbf{l}]^T \mathbf{K} [\mathbf{d}(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i) - \mathbf{l}], \quad (2)$$

where the actual design parameters are  $\mathbf{s}$  and  $\mathbf{l}$ . This constitutes an obvious non-linear least squares optimization problem. Defined that way, the problem remains difficult to solve. Therefore, two propositions are given below to improve the homogeneity of the problem and to avoid multiple triangle configurations.

Firstly, let us remark that the actual design parameters, the *static* points coordinates  $\mathbf{s}$  and the natural length  $\mathbf{l}$ , appear in different terms of the cost function. This makes the cost function differently sensitive to both of them. We propose to transform each static point coordinates into the natural lengths of two springs (see Fig. 2). In this way, a new floating point is inserted in  $\mathbf{f}$ , the vector  $\mathbf{s}$  is appended to the vector  $\mathbf{l}$  and two new stiffness parameters are added to the diagonal of matrix  $\mathbf{K}$ . Note that the corresponding functions  $d(\mathbf{s}, \mathbf{t}_i, \mathbf{f}_i)$  become actually the two coordinate values which are not always positive: this introduces so-called *oriented* springs according to the sign of their natural lengths. But this has no consequence on the energy formulation (1). For example, replacing *static* point  $(x_0, y_0)$  of Fig. 2 would create two additional contributions to the total energy:  $\frac{1}{2}k_x(x - x_0)^2$  and  $\frac{1}{2}k_y(y - y_0)^2$ . Thanks to this transformation, all the design parameters may be grouped into the same vector  $\mathbf{l}$ .

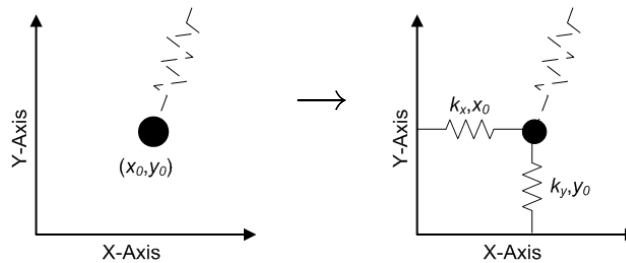


Figure 2: New model of static points

The second proposition relates to the three springs composing the triangle  $P_1, P_2, P_3$ . Fixing the points  $P_1$  and  $P_2$ , two stable positions remain for  $P_3$ : above or below the  $P_1 - P_2$  line. To remove the ambiguity, the use of *oriented* springs (see above) is proposed to locate univoquely  $P_3$  with respect to  $P_1$  and  $P_2$ . Thus, the two springs  $P_1 - P_3$  and  $P_2 - P_3$  are replaced by two *orthogonal oriented* springs as shown in Fig. 3. In

this example, both contributions of springs 3 and 4 are replaced by:  $\frac{1}{2}k_a(a - a_0)^2$  and  $\frac{1}{2}k_b(b - b_0)^2$ , with  $a = \frac{\overrightarrow{P_1P_2} \cdot \overrightarrow{P_1P_3}}{\|\overrightarrow{P_1P_2}\|}$  and  $b = \frac{\overrightarrow{P_1P_2} \times \overrightarrow{P_1P_3}}{\|\overrightarrow{P_1P_2}\|} \cdot \hat{z}$ .

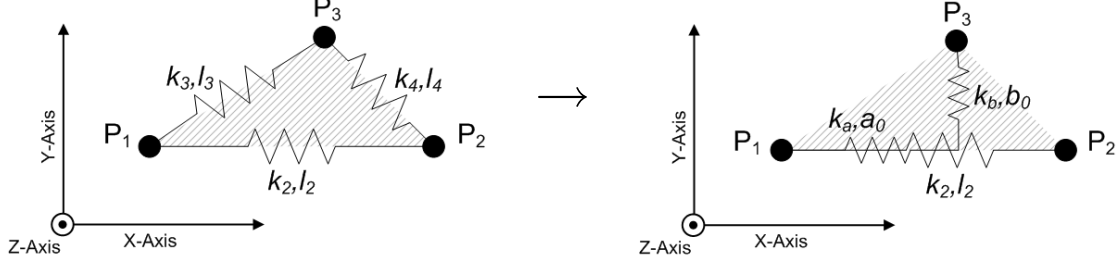


Figure 3: New model of triangle element

Finally, taking both propositions into account, the total cost function (1) becomes:

$$E(\mathbf{t}_1, \dots, \mathbf{t}_N, \mathbf{f}_1, \dots, \mathbf{f}_N, \tilde{\mathbf{l}}, \tilde{\mathbf{K}}) = \frac{1}{2} \sum_{i=1}^N (\tilde{\mathbf{d}}_i - \tilde{\mathbf{l}})^T \tilde{\mathbf{K}} (\tilde{\mathbf{d}}_i - \tilde{\mathbf{l}}), \quad (3)$$

leading to the following rearranged optimization problem:

$$\min_{\mathbf{f}_1, \dots, \mathbf{f}_N, \tilde{\mathbf{l}}} \frac{1}{2} \sum_{i=1}^N [\tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{l}}]^T \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{l}}], \quad (4)$$

where the tilde symbol stands for both modifications described above. A corresponding configuration of the four-bar model is illustrated in Fig. 4. Let us note the size of the different vectors for this simple four-bar example: 9 components in  $\tilde{\mathbf{l}}$ , 8 in  $\mathbf{f}_i$  and 2 in  $\mathbf{t}_i$ . Therefore, for the  $N$  configurations,  $9 + 8 * N$  optimization variables have to be taken into account for the problem (4), which may be a lot.

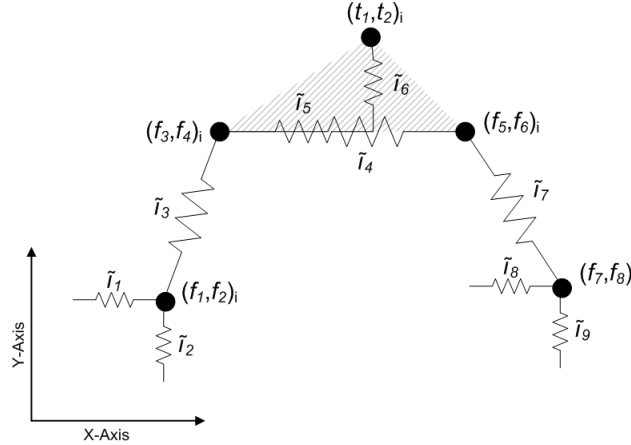


Figure 4: One configuration of the four-bar model for path synthesis

In problem (4), two kinds of optimization variables are now considered:  $\tilde{\mathbf{l}}$  and  $\mathbf{f}_i$ ,  $i = 1 \dots N$ . The gradients of the total energy function with respect to both these vectors are given by:

$$\frac{\partial E}{\partial \mathbf{f}_i} = \frac{\partial \tilde{\mathbf{d}}^T(\mathbf{t}_i, \mathbf{f}_i)}{\partial \mathbf{f}_i} \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{l}}] \quad (5)$$

$$\frac{\partial E}{\partial \tilde{\mathbf{l}}} = - \sum_{i=1}^N \tilde{\mathbf{K}} [\tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{l}}] \quad (6)$$

Let us point out that (5) only depends on the  $i^{th}$  configuration if the design parameters  $\tilde{\mathbf{I}}$  are fixed. This may greatly simplify the optimization problem and is the basis of the first part of optimization strategy described in the next Section.

### 3 Optimization strategy

The optimization strategy is divided into two subsequent parts. Its algorithm flowchart is sketched in Fig. 5. The first part is inspired from [6] who proposed to minimize the deviation of each variable dimensions over a cycle and to update the mean value after each cycle. The main difference here is the use of natural coordinates instead of relative coordinates which make the objective function far more non-linear due to trigonometric functions.

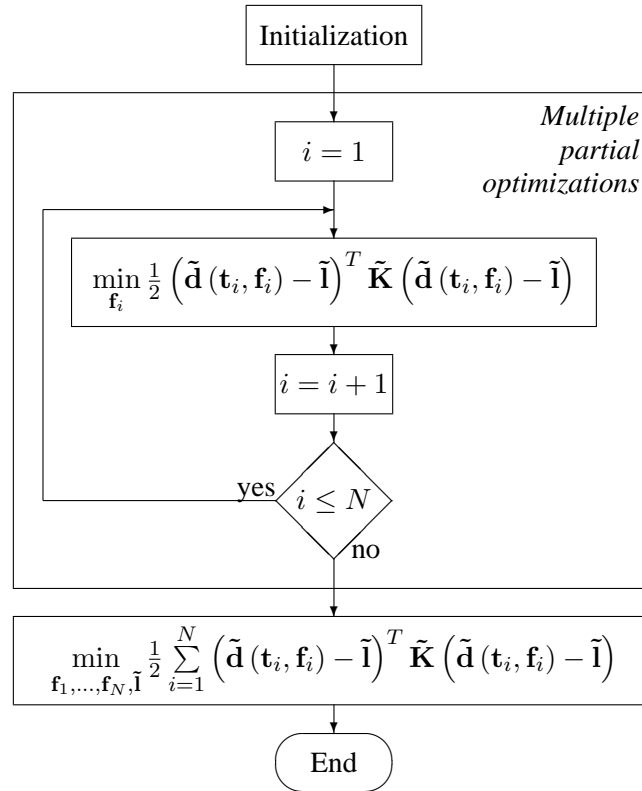


Figure 5: Partial and total synthesis flowchart

Starting from given values of the design parameters  $\tilde{\mathbf{I}}$ , the algorithm begins minimizing the total energy with respect to the  $\mathbf{f}_i$ . This is equivalent to solving  $N$  partial optimization problems because the  $\mathbf{f}_i$  are independent and  $\tilde{\mathbf{I}}$  is constant:

$$\begin{aligned}
 & \min_{\mathbf{f}_1, \dots, \mathbf{f}_N} \frac{1}{2} \sum_{i=1}^N \left[ \tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{I}} \right]^T \tilde{\mathbf{K}} \left[ \tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{I}} \right] \\
 \Leftrightarrow & \sum_{i=1}^N \min_{\mathbf{f}_i} \frac{1}{2} \left[ \tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{I}} \right]^T \tilde{\mathbf{K}} \left[ \tilde{\mathbf{d}}(\mathbf{t}_i, \mathbf{f}_i) - \tilde{\mathbf{I}} \right] \quad (7)
 \end{aligned}$$

After one optimization cycle over the  $N$  configurations begins the second part of the algorithm. It consists in a *total* optimization process of problem (4) that involves all the floating points coordinates and all the design parameters. To help the optimizer, this process uses the results of the first part as hot starting points for the  $\mathbf{f}_i$ .

As explained in Section 2, the number of optimization parameters may increase rapidly (e.g.  $9 + 8N = 169$  variables for the four-bar mechanism with  $N = 20$  synthesis points) if the mechanism and/or the path get more complex. As the parameter space is larger, a more robust optimization algorithm is needed. For instance, the so-called dog-leg algorithm [14]: this trust-region method is also well-known and useful to solve systems of nonlinear equations.

## 4 Application to four-bar steering linkage synthesis

This Section presents an interesting application of function generation synthesis. The goal is to optimize steering linkage of vehicles. In the first subsection, the function to generate is established from the Ackermann condition. Secondly, the proposed optimization strategy is applied to the synthesis of a four-bar steering linkage.

### 4.1 The Ackermann condition

One of the main requirements of the steering mechanism of a vehicle is to give to the steerable wheels a correlated turning, ensuring that the intersection point of their axis lies on the extension of the rear wheel axis (point  $P$  in Fig. 6). The Ackermann relation [22] of correct turning is:

$$\cot \delta_o - \cot \delta_i = \frac{l}{L}, \quad (8)$$

where  $\delta_o$  and  $\delta_i$  are the outer and inner wheel angles respectively,  $l$  is the track width and  $L$  the wheelbase of the vehicle. Only the track width-wheelbase ratio influences this Ackermann steering relation.

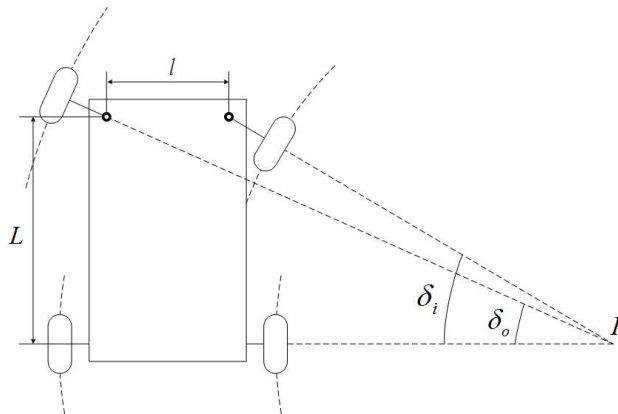


Figure 6: The Ackermann condition

### 4.2 Four-bar steering linkage synthesis

The modeling of the four-bar steering linkage is worked out according to the rules depicted in Section 2. To satisfy the Ackermann condition (8), the correlated path-following of the wheel centers are imposed while the inner wheel angle takes 40 different values between 0 and the maximum (see Fig. 7). Also observe in the Figure that the static points are not transformed into floating points because they do not belong to the design parameters. These parameters consists a priori in three natural lengths:  $a$ ,  $b$  and  $l$ . However, the problem symmetry reduces their number to only two (e.g.  $a$  and  $b$ ) because  $l = \|\vec{P_0P_5}\| - 2a$ . Remark

that the function is penalized around the origin of the design space to avoid singular configurations of the mechanism. The objective function (3) is extended here as follows:

$$E_{ext} = \begin{cases} E & \text{if } \|\overrightarrow{P_0P_2}\| \geq d_{min} \\ \frac{1}{2} \left( \|\overrightarrow{P_0P_2}\| - d_{min} \right)^2 & \text{if } \|\overrightarrow{P_0P_2}\| < d_{min} \end{cases}, \quad (9)$$

where the threshold value  $d_{min}$  is a chosen realistic minimum distance (e.g.  $d_{min} = 10$  cm).

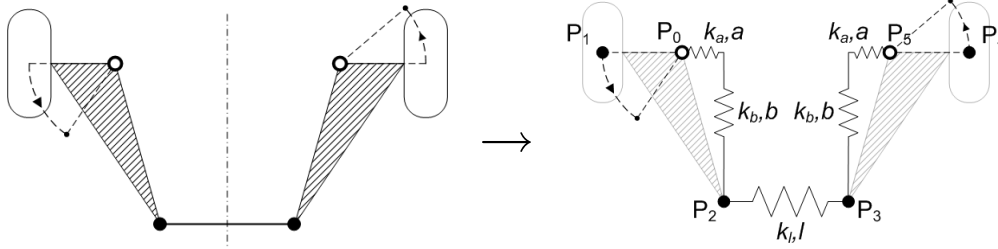


Figure 7: Model adaptation of four-bar linkage for function-generation synthesis

As for the "total" synthesis algorithm, it is observed that the optimization process may reach one of both local optima [19, 5]. Starting from different initial parameters, it is sometimes hard to guess where it converges. Fig. 8 shows that running the algorithm from initial points located on a uniform 7-by-7 grid, these processes lead to 47 relevant optimization results. Among the latter – symbolized by non-bold 'o' and 'x' –, 11 of them converge to one local optimum – symbolized by bold 'x' – while the 36 others reach another one – symbolized by bold 'o' – which is actually the global one. The optimization method does not always yield the global optimum.

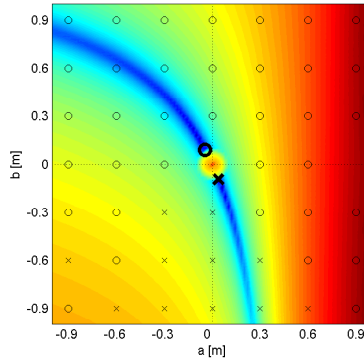


Figure 8: Optimization of the steering linkage from different starting points

	$a$ [cm]	$b$ [cm]	Energy [J]	Max error [deg]	RMS error [deg]
Leading	-4.40	8.98	$7.98 \cdot 10^{-7}$	0.71	0.28
Trailing	3.90	-9.21	$1.01 \cdot 10^{-6}$	0.80	0.32

Table 1: Two best four-bar Ackermann steering linkages

The two best linkages are drawn in Fig. 9: a trailing one and a leading one. Their steering error functions are plotted in Fig. 10 which represents the deviation of the outer wheel angle with respect to the Ackermann condition when the inner wheel turns from  $0^\circ$  to  $40^\circ$ . The numerical results are shown in Table 1. It is interesting to remark that the optimum energy value is linked to the steering error: the leading linkage has the smallest total deformation and also the smallest steering error. Nevertheless, the small difference between

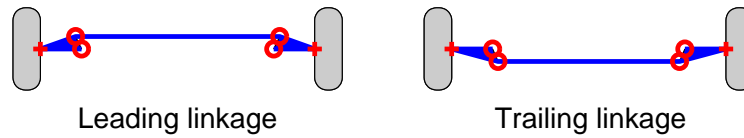


Figure 9: Two local optima found for the four-bar steering synthesis

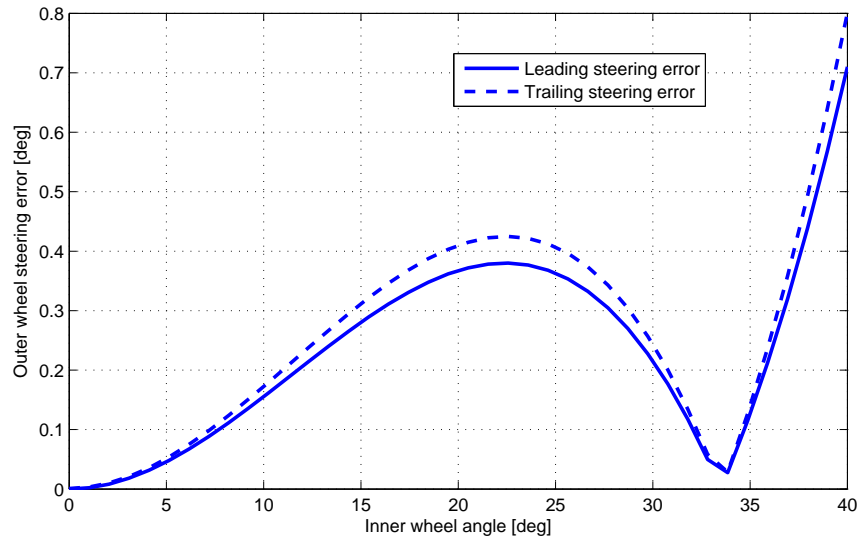


Figure 10: Steering error of both optimum linkages

both mechanisms performances does not justify the selection of one instead of the other: an additional design criterion has to be introduced to choose between both local optima.

## 5 Exploration of the design space

Regarding the example of the previous Section, it is observed that the optimization process does not always reach the unique global optimum according to the chosen starting point. However, it may be interesting to propose several local optima to the designer. The first Subsection describes a simple novel method to explore the entire parameters space in order to find local optima. In the second Subsection, we explain a possible method to choose the final best mechanism. Both Subsections are illustrated in the application of the next section.

### 5.1 Exploration with the *nucleation* method

The first idea to explore the design space could be to perform multiple optimization processes starting from different uniformly-distributed initial points. A two-point-width uniform grid (Fig 11.a) is used first and refined thereafter by adding a new points between each segment: this provides a three-point-width grid (Fig 11.b). Continuing the refinement this way, this enables us to reuse the computation of the previous grid (gray points) as shown in the following of Fig 11. However, it becomes very time-consuming since the number of optimization processes exponentially increases (see second column of Table 2). A new method called '*nucleation*' is thus proposed to cope with this drawback.

This original method takes its inspiration from the nucleation process of crystal materials. Its algorithm flowchart is sketched in Fig. 12. The key idea is to create and make nucleus crystals growing from best



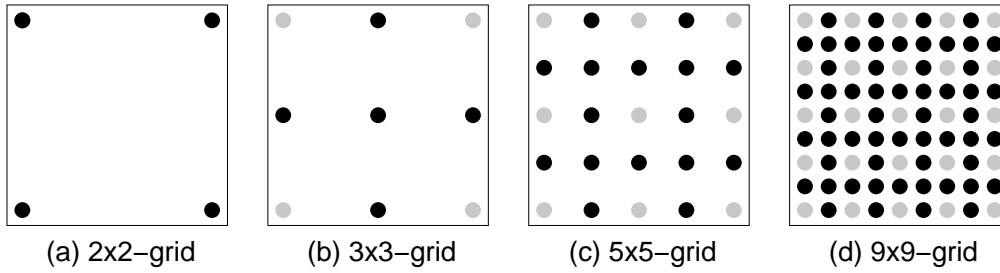


Figure 11: Refinement of the starting points grid

locations on a given grid. These best locations are subsequently chosen with respect to the value of objective function. The growth of the nuclei is stopped when they reach other nuclei or the design space boundary.

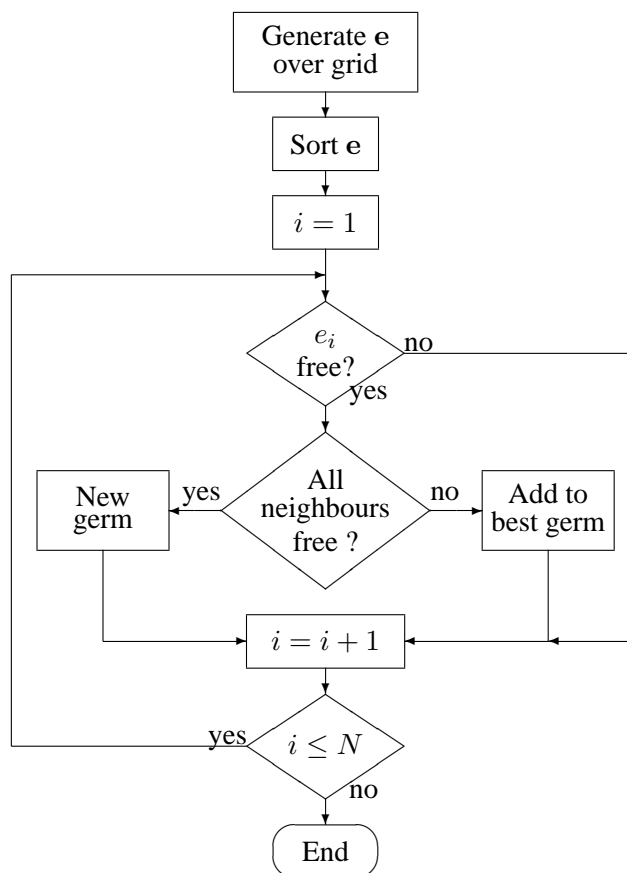


Figure 12: Nucleation algorithm flowchart

Practically, this algorithm, sketched in Fig. 12, begins with the discretization of the whole design space into a grid of equally-spaced points. The objective function  $e$  is computed over the grid and these points are sorted according to their objective value. In our case, this objective is the total strain energy (7). Then, each point of this sorted list is scanned to classify it: if all its neighbors are free, i.e. not yet scanned and with worse objective values, a new nucleus is created with the point and its neighbors; otherwise, it is added to the nucleus of its best neighbor. All these operations are made until the whole list is fully scanned. The final result is the grouping of all the points around different nuclei.

Taking the simple example of Fig. 13, a 3x3 grid is explored. The integer number represent the value of the energy in each point. Five iterations are needed to group the 9 points. The two first iterations

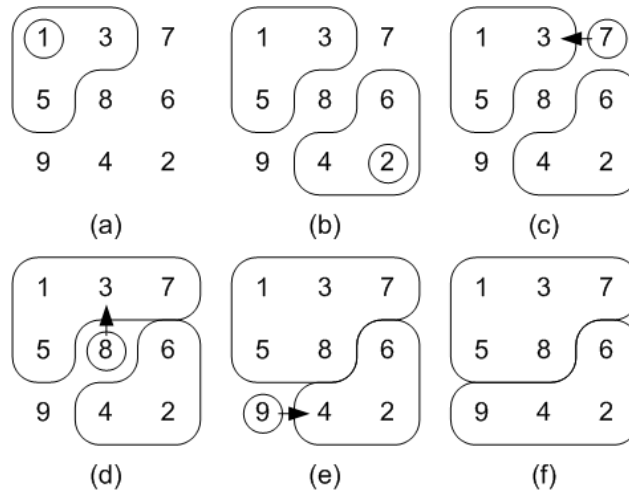


Figure 13: *Nucleation* algorithm example

(Fig. 13.a and 13.b) create the two first nuclei around points of energy 1 and 2. The considered neighbors, inside the boundaries, are located up, down, left and right from the center point. In iteration 3 (Fig. 13.c), point 7 is added to the nucleus of its best neighbor (point 3). In the same way, point 8 is added to the nucleus of point 3 and point 9 to the nucleus of point 4 (Fig. 13.d and 13.e). The final configuration with two nuclei is illustrated in Fig. 13.e.

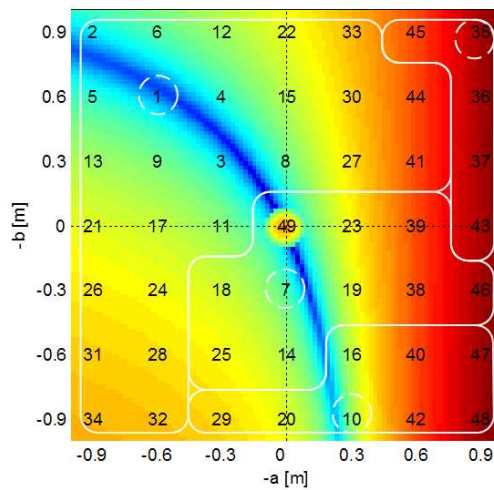


Figure 14: Four-bar steering example with *nucleation* algorithm

If the four-bar steering linkage synthesis is now considered with the *nucleation* algorithm, four nuclei are obtained using only the partial minimization (7) as shown in Fig. 14. Thanks to this technique, the total optimization (4) may be run only 4 times instead of 49 (see Section 4.2) to obtain the two local optima.

## 5.2 Final choice among the local optima

Once the design space is explored to form nuclei, the total synthesis may begin starting from the best candidate of each nucleus. This provides us many local optima with respect to the minimum deformation. Thereafter, a last design optimization step has to be performed to find the final mechanism. This last criterion is applied on the corresponding *rigid* mechanisms and is obviously different from the one used to create

the nuclei. Concerning the steering linkage synthesis, the *nucleation* process is based on the minimum deformation energy (7). However, the last objective will be the actual steering error of the rigid mechanism (as shown in Fig. 10 for the four-bar linkage) or even a more practical objective to choose the best candidate (see the end of Section 6). It is therefore computed by simulation of the rigid mechanism instead of the extensible one. This is illustrated in the next Section.

## 6 Application to six-bar steering linkage synthesis

The goal of this application is the same as for the four-bar steering linkage. The main difference consists in the model complexity. The four-bar was parameterized with only two dimension variables. Here, the six-bar model, sketched in Fig. 15, is composed of five design parameters –  $a, b, l_1, l_2, y$  – which are reduced to four because of the symmetry [20]. For example,  $l_1$  can be expressed in terms of  $a, b, l_2$  and  $y$ :

$$l_1 = \sqrt{\left(\frac{l - l_2}{2} - a\right)^2 + (b - y)^2} \quad (10)$$

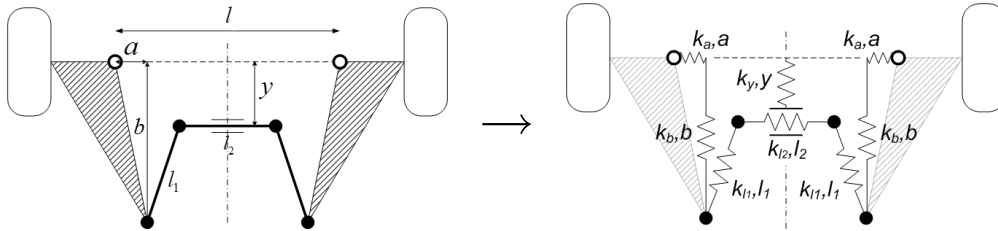


Figure 15: Model adaptation of six-bar steering linkage

Applying the *nucleation* method to group the points on a given grid is particularly more relevant for the six-bar than the four-bar linkage as shown in Table 2. Let us remember that the *nucleation* process has enabled us to reduce the number of total synthesis from 49 – 7x7 grid – to 4 runs for the four-bar (See Section 5.1 and Fig. 14). In the case of the six-bar, this reduction factor –  $\frac{\#points}{\#nuclei}$  – increases with the size grid and can reach 294 for a 17-point-width grid, compared to  $49/4 = 12$  for the four-bar. This represents a considerable gain of CPU-time.

Grid width	Number of points	Number of nuclei	Reduction factor	Number of local optima
2 pts	16	1	16	1
3 pts	81	2	41	2
5 pts	625	13	48	6
9 pts	6561	48	137	12
17 pts	83521	284	294	9

Table 2: Numerical results of the design space exploration of six-bar linkage by *nucleation* method

Starting from the best candidate of each nucleus, the total optimization (4) is performed. The number of local optima is observed in 2. The *nucleation* process applied on the most refined grid highlights 9 local optimum mechanisms. As previously explained, before the selection of the most relevant local optimum, the optimization has to be refined with respect to the steering error as explained in Section 5.2. This last objective is computed by simulating the steering behavior of the corresponding rigid mechanism as already shown in Fig. 10. The final candidates resulting from this optimization refinement are reported in Table 3 and sketched in Fig. 16.

	$a$	$b$	$l_2$	$y$	Energy	Max error	RMS error	Mean dev.
	[m]	[m]	[m]	[m]	[J]	[deg]	[deg]	[% of center]
(a)	-0.99	0.09	1.33	0.100	$6.78 \cdot 10^{-12}$	0.0015	0.0003	>100 %
(b)	0.89	-0.46	1.37	0.101	$1.72 \cdot 10^{-11}$	0.0024	0.0005	>100 %
(c)	0.09	-0.12	1.21	0.095	$2.01 \cdot 10^{-08}$	0.0220	0.0031	>100 %
(d)	-0.83	-0.53	1.28	0.108	$2.10 \cdot 10^{-09}$	0.0685	0.0151	>100 %
(e)	-0.09	0.20	1.08	-0.104	$4.48 \cdot 10^{-08}$	0.1026	0.0335	>100 %
(f)	-0.06	0.08	1.11	0.093	$1.56 \cdot 10^{-05}$	0.3215	0.1232	45 %
(g)	0.09	-0.04	1.34	-0.099	$9.20 \cdot 10^{-06}$	0.3659	0.1245	>100 %
(h)	-0.11	0.09	0.61	0.601	$1.11 \cdot 10^{-06}$	0.5977	0.1923	2 %
(i)	0.09	0.04	0.75	-0.953	$2.34 \cdot 10^{-04}$	0.5647	0.1718	37 %

Table 3: The nine local optima after optimization refinement

Compared with the results of the four-bar steering linkage in Fig. 10, all the nine local optima improve both the maximum and the RMS steering errors. Moreover, these values are so small that it could be interesting to add a more practical criterion. A robustness criterion is proposed: the sensitivity analysis of the steering error with respect to the optimum design parameters. A perturbation of 0.5 mm – which could correspond to absolute precision machining – is chosen for each of the four dimensions and also for all combination of them. The steering error cost function is thus computed 16 times around each of the nine optima. In the last column of Table 3, the mean deviation of the objective over the 16 neighbors of each optimum is represented. This may give a basis for the last decision of the designer. Undeniably, mechanism (h) is the most robust in the sensitivity sense. However, the designer could choose mechanism (f) because it is more compact than (h) or (i). All these considerations obviously depend on the way each criterion is taken into account by the designer.

## 7 Conclusion and prospects

Based on a strain energy approach of extensible-link mechanisms coupled with the use of natural coordinates, an original optimization method has been developed to solve path synthesis and function generation problems of planar mechanisms. Divided into two stages, the method tries first to find good starting points with multiple *partial* optimizations and then uses them in a *full* synthesis of the mechanism. It seems efficient but does not guarantee to obtain the global optimum as it was illustrated via the four-bar steering linkage application.

Following that observation, the question of finding the global optimum has been reviewed and extended with the exploration of the design space to find the most of local optima. To avoid 'astronomical' CPU time, a simple method inspired from crystals nucleation has been proposed to divide the design space into nuclei centered on local optima. Starting from the latter, the optimization has been refined and a last criterion applied to choose the final mechanism.

In terms of prospects, our effort will concentrate on developing other exploration strategies of the design space. The comparison of these different strategies and their results will be useful to validate them. It could be also interesting to classify the different local optima based on different types of criteria. We also intend to extend the method further to topologies with prismatic joints or to three-dimensional mechanisms. Extending the application field is also an interesting prospect: other kinematic objectives instead of path or function-generator synthesis or even dynamical ones could be investigated. The more challenging issue of topology optimization of mechanisms could be tackled on the basis of this work, involving for example an additional higher-level optimization process as proposed by [17], or optimizing simultaneously the topology and the dimensions of mechanisms as developed by [21], or [9] for path-generation problems using truss representation.

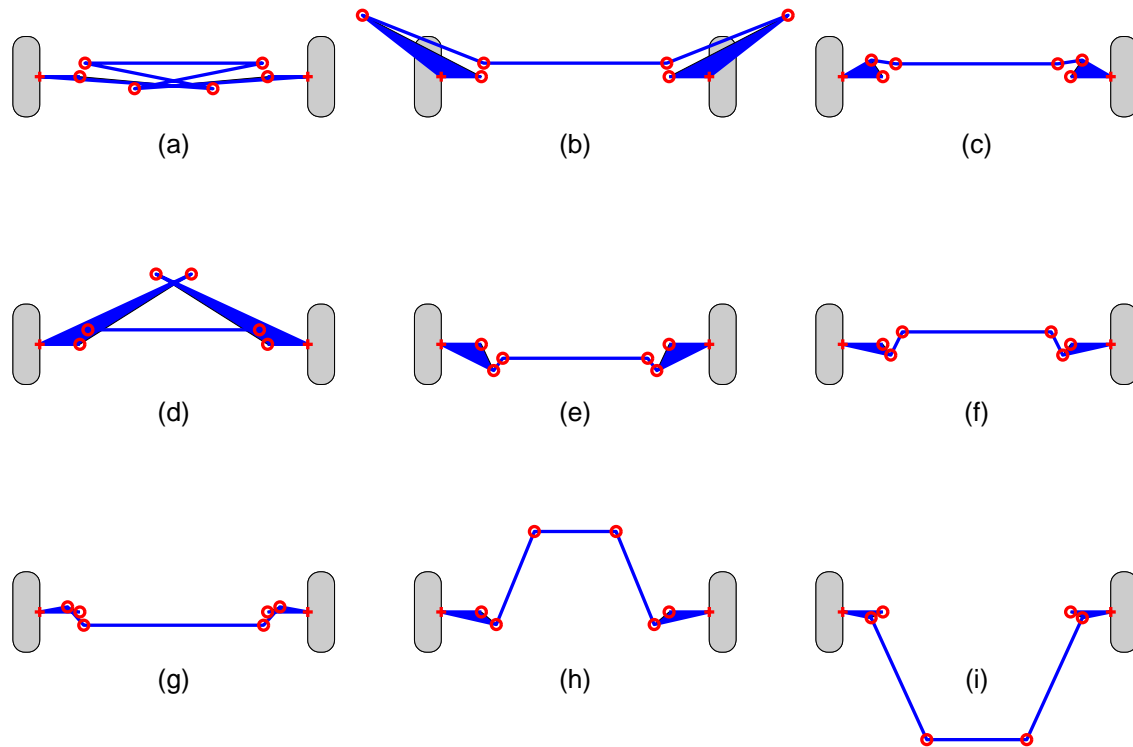


Figure 16: The nine local optima after optimization refinement

## Acknowledgments

This research has been sponsored by the Belgian Program on Interuniversity Attraction Poles, *Advanced Mechatronic Systems - AMS* (IUAP5/06), initiated by the Belgian State — Prime Minister's Office — Science Policy Programming (IUAP V/6). The scientific responsibility is assumed by its authors.

## References

- [1] J.A. Alba, M. Doblaré, and L. Gracia. A simple method for the synthesis of 2D and 3D mechanisms with kinematic constraints. *Mechanism and Machine Theory*, 35:645–674, 2000.
- [2] R. Avilés, J. Vallejo, G. Ajuria, and J. Agirrebeitia. Second-order methods for the optimum synthesis of multibody systems. *Structural and Multidisciplinary Optimization*, 19:192–203, 2000.
- [3] J.A. Cabrera, A. Simon, and M. Prado. Optimal synthesis of mechanisms with genetic algorithms. *Mechanism and Machine Theory*, 37:1165–1177, 2002.
- [4] J.-F. Collard, P. Fisette, and P. Duysinx. Contribution to the optimization of closed-loop multibody systems: Application to parallel manipulators. *Multibody System Dynamics*, 13:69–84, 2005.
- [5] J.-F. Collard, P. Fisette, and P. Duysinx. Optimal synthesis of mechanisms using time-varying dimensions and natural coordinates. In *Proc. of the Sixth World Congress of Structural and Multidisciplinary Optimization*, 2005.
- [6] John M. Hansen. Synthesis of mechanisms using time-varying dimensions. *Multibody System Dynamics*, 7:127–144, 2002.

- [7] Ole Fjendbo Jensen and John M. Hansen. Treating the problem of non-assembly for dimensional synthesis of spatial mechanisms. In Schönfeld & Ziegler, editor, *The Fifth World Congress of Structural and Multidisciplinary Optimization*, pages 431–432. ISSMO, Italian Polytechnic Press, may 2003.
- [8] J. M. Jiménez, G. Alvarez, J. Cardenal, and J. Cuadrado. A simple and general method for kinematic synthesis of spatial mechanisms. *Mechanism and Machine Theory*, 32(3):323–341, 1997.
- [9] A. Kawamoto. Path-generation of articulated mechanisms by shape and topology variations in non-linear truss representation. *International Journal for Numerical Methods in Engineering*, 64:1557–1574, 2005.
- [10] M.A. Laribi, A. Mlika, L. Romdhane, and S. Zegloul. A combined genetic algorithm-fuzzy logic method GA-FL in mechanisms synthesis. *Mechanism and Machine Theory*, 39:717–735, 2004.
- [11] Mauro Da Lio, Vittore Cossalter, and Roberto Lot. On the use of natural coordinates in optimal synthesis of mechanisms. *Mechanism and Machine Theory*, 35:1367–1389, 2000.
- [12] Francisco T. Sánchez Marín and Antonio Pérez González. Global optimization in path synthesis based on design space reduction. *Mechanism and Machine Theory*, 38:579–594, 2003.
- [13] R.J. Minnaar, D.A. Tortorelli, and J.A. Snyman. On nonassembly in the optimal dimensional synthesis of planar mechanisms. *Structural and Multidisciplinary Optimization*, 21:345–354, 2001.
- [14] M.J.D. Powell. *Nonlinear Programming*, chapter A new algorithm for unconstrained optimization. Academic Press, 1970.
- [15] S. Pramanik. Kinematic synthesis of a six-member mechanism for automotive steering. *ASME Journal of Mechanical Design*, 124:642–645, dec 2002.
- [16] R. Sancibrian, F. Viadero, P. García, and A. Fernández. Gradient-based optimization of path synthesis problems in planar mechanisms. *Mechanism and Machine Theory*, 39:839–856, 2004.
- [17] Kai Sedlaczek, Timo Gaugele, and Peter Eberhard. Topology optimized synthesis of planar kinematic rigid body mechanisms. In *Proc. of the Multibody Dynamics 2005, Eccomas Thematic Conference*, 2005.
- [18] P.S. Shiakolas, D. Koladiya, and J. Kebrle. On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique. *Mechanism and Machine Theory*, 40:319–335, 2005.
- [19] P.A. Simionescu and D. Beale. Optimum synthesis of the four-bar function generator in its symmetric embodiment: the ackermann steering linkage. *Mechanism and Machine Theory*, 37:1487–1504, 2002.
- [20] P.A. Simionescu, M.R. Smith, and I. Tempea. Synthesis and analysis of the two loop translational input steering mechanism. *Mechanism and Machine Theory*, 35:927–943, 2000.
- [21] M. Stolpe and A. Kawamoto. Design of planar articulated mechanisms using branch and bound. *Mathematical Programming*, 103:357–397, 2005.
- [22] J.Y. Wong. *Theory of Ground Vehicles*, chapter Handling characteristics of Road Vehicles, pages 335–387. John Wiley & Sons, 3rd edition, 2001.