TENTH WORLD CONGRESS ON THE THEORY OF
MACHINES AND MECHANISMS
Oulu, Finland, June 20–24, 1999

FOUR–BAR: AN EDUCATIONAL SYNTHESIS– AND DESIGN TOOL FOR PLANAR FOUR BAR MECHANISMS

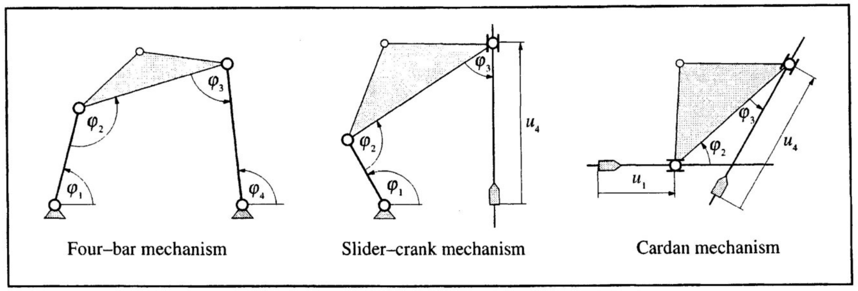
G. Huber, P. Dietmaier
Technische Universität Graz, Institut für Allgemeine Mechanik
Kopernikusgasse 24, A–8010 Graz
e-mail: dietmaier@mech.tu-graz.ac.at

Keywords: *Planar four–bar mechanism, educational software*

1 Introduction

The kinematics of a planar four–bar linkage for rigid bodies connected by revolute pairs with parallel axis of rotation has been investigated for many years and the available literature [e. g. Bottema and Roth 1990] covers almost all kinematical aspects of this most simple planar mechanism (Figure 1). In introductory classes on kinematic geometry or mechanism analysis and design this is one of the mechanisms most often used as a first example. It is therefore not a surprise that there are several educational and commercial software packages available which treat one or the other kinematical aspect of this mechanism in more or less detail. However, most of these computer programs are either not robust enough or are not capable of handling special cases which makes them not very useful for educational purposes. The software introduced in this paper was therefore primarily developed as a supplementary educational tool which overcomes most of this drawbacks. The presented computer program extends the capabilities of a similar software developed for the DOS operating system by Hiesleitner [Hiesleitner 1991] and one of the authors. Beside the four–bar mechanism the new software can now also tread the special cases of the slider–crank and the Cardan mechanism. Furthermore, several new synthesis

Figure 1. The planar four–bar mechanism and its special cases.



features were added. One of the design goals for this software was its usability on several hardware platforms and operating system. Therefore we used the rather new and prospering Java™ programming language [Gosling et al. 1996].

2 Features of the *Four–Bar* software

The *Four–Bar* software, a Java™ application, at the current state is capable of analyzing and synthesizing the general four–bar linkage as well as its special cases the slider–crank mechanism and the Cardan mechanism (see Figure 1). Our principle design goal was to develop a software–package primarily for educational usage. With the help of a graphical user interface we tried to make the handling as intuitively as possible. So for example, the position of the joints, the coupler and the coupler point can be specified by dragging the corresponding parts into the desired position. On the other hand we tried hard to make sure that the presented kinematical results are as complete as possible. Therefore, all possible positions for all possible closure modes of the currently specified mechanism are computed automatically. The user dose not have to specify one of the joints as a driven joint. As a result the user obtains immediately a rather complete overview of the kinematic properties of a particular mechanism.

2.1 Analysis tasks

In the current version of the program we support various analysis tasks which are carried out immediately. This means when the user is changing any dimension of the current mechanism using the mouse, all requested analysis tasks are executed so fast as if the results are seemingly changing continuously.

One of the analysis tasks performed by the program is the computation of the coupler–curve for a specified coupler–point. As mentioned above, all parts of the coupler–curve (for all closure modes) are computed fully automatically. The part of the coupler–curve which is traced by the current assembly mode of the mechanism is displayed as solid black line. The part(s) of the coupler–curve which can not be reached by the current assembly on the other hand is (are) shown as a gray line as depicted in Figure 2. By clicking on the gray part(s) of the coupler-curve the user can switch between the different assembly modes. From this sequence of actual screen shots we can also observe the influence of the dislocation of the right hand fixed–pivot on the shape and the number of parts of the corresponding coupler–curve. This series also verifies the completeness of the computed coupler–curve as well as the fact that the program is capable of handling difficult border line cases.

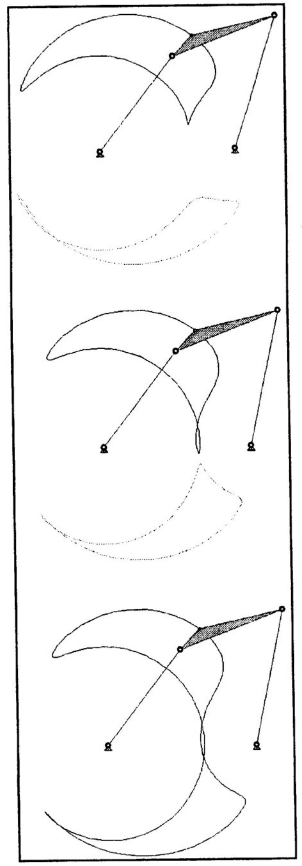


Figure 2. The number of closure modes changes when the right fixed pivot is moved (actual screen shots).

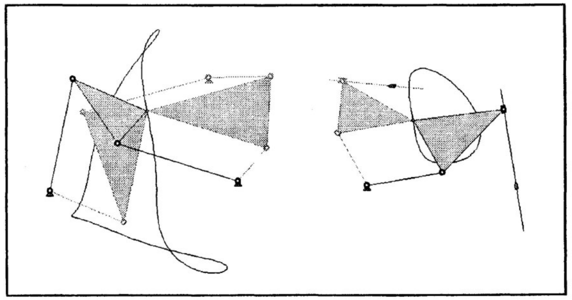


Figure 3. The two Roberts cognates of the four-bar mechanisms and the one cognate for the slider–crank mechanism.

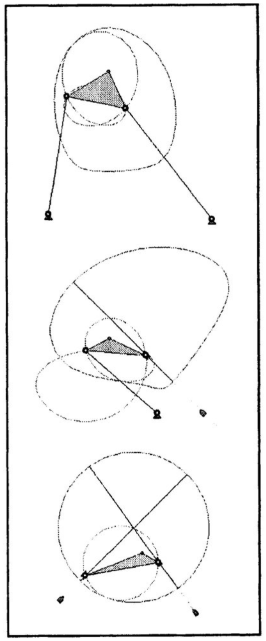


Figure 4. Pole–curves.

From the literature [Roberts 1875] it is well known that the coupler–curve of a particular planar four-bar mechanism is not just generated by this but also by two other mechanisms, so called path–cognates. For the planar slider–crank mechanism, as a special case of the planar four-bar mechanism, there exits just one of these cognates. *Four–Bar* computes the cognates and displays them as shadow images together with the actual mechanism (Figure 3). When the actual mechanism is animated the cognates move collectively with it and in this way Roberts theorem is presented in the most convincing way. Clicking on one of the shadow images the user can turn a cognate into the mechanism which can be modified and analyzed with respect to other topics.

The motion of the coupler relative to the base (represented here by the two fixed pivots) can also be accomplished by a rolling motion of one specific curve attached to the base and another curve attached to the coupler. These so called pole–curves can also be depicted by the software (Figure 4). Animating the representations of Figure 4 demonstrates the equivalence of the rolling motion of the pole–curves and the motion of the coupler determined by the corresponding mechanisms.

Besides the coupler–curve and the pole–curves the program can also display, as one example of an envelope generated by a curve attached to the coupler, the envelope of a straight line attached to the coupler (again for all closure modes). Furthermore, the program can graphically display the relations between the various position parameters (φ_i, u_i; see also Figure 1), transmission functions. Moreover, for a single position or during animation the program can show the velocity and acceleration of points on the coupler, the circles of Bresse and De La Hire, and the circle of curvature of the coupler curve.

2.2 Synthesis tasks

For all the analysis tasks *Four–Bar* can perform the essential design parameters of the corresponding mechanism must be specified. In contrast to that, however, the synthesis task consists of finding the essential dimensions of one type of mechanism ina way that the particular mechanism exhibits a certain behavior. While *Four–Bar is* able to carry out all of the analysis tasks in real time this dose not hold true on the other hand for all of the synthesis tasks. However, there

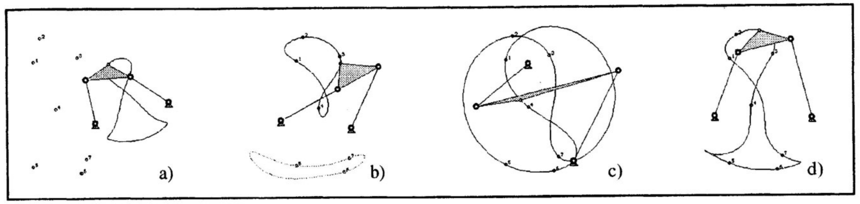


Figure 5. Coupler–curve synthesis: a) initial configuration; b) coupler–points dispersed over two closure modes; c) coupler–points dispersed over a single closure mode; d) like c) but with consecutive point arrangement.

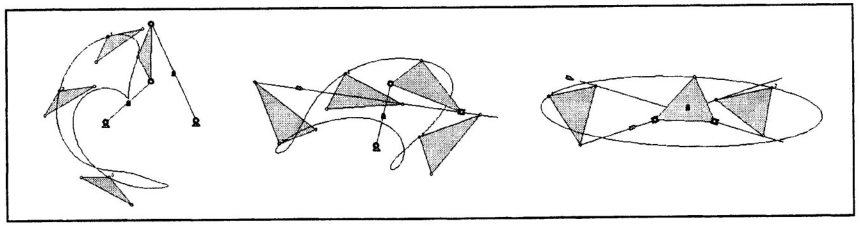


Figure 6. Coupler–position synthesis with restrictions on some of the design parameters.

exist theoretical results which would have made it possible to also solve some of the synthesis tasks again in real time. For other synthesis tasks these results do either not exist or the solution procedures are way to complicated to be performed in real time. We therefore decided to use a more general and unified approach for the synthesis tasks and give up on the option to solve some of them in real time. So, after the user has provided the necessary information for a particular synthesis task the program starts modifying the initial (current) mechanism until the analysis problem is solved. During the analysis process the user can watch the alterations made by the program and can help the program to find a particular solution by making changes to the mechanisms interactively. The capabilities of the program with respect to synthesis tasks are best demonstrated by the use of the following two examples.

Figure 5 shows the results of a coupler–curve synthesis. The goal of this synthesis is the determination of the essential dimensions of a mechanism so that the coupler–curve passes through a set of prescribed points as accurate as possible. The program offers for such a synthesis task three different options: 1. Determine the mechanism such that the given points are as close as possible to the coupler–curve no matter which closure mode it belongs to (b). 2. Compute a mechanism such that all given points are as close as possible to a part of the coupler–curve of the same closure mode (c). 3. Find a mechanism so that all given points are consecutively passed by a part of the coupler–curve of the same closure mode (d).

Figure 6 shows the results of a coupler–position synthesis for two and three coupler–position respectively. In comparison to the coupler–curve synthesis the coupler–position synthesis targets to find a mechanism where the coupler when moving passes through a set of prescribed positions as accurate as possible. Furthermore it is possible to specify that either the shape of coupler has to remain unchanged, the length of a crank must not be altered, or that a fixed pivot must stay in place. These restrictions are indicated by a black lock on the corresponding parts.

Beside these two synthesis tasks the program can also determine the design–parameters of a mechanism so that the envelope traced by a straight line attached to the coupler approaches a given set of points as close as possible. In addition to that an input–output synthesis, requiring a transmission–function passing through a given set of pairs of position parameter values, can be performed by the program.

3 Algorithmic background

3.1 Computation of all positions of one closure mode

An essential part of any analysis software for planar or spatial mechanisms is a robust position analysis procedure. In addition to the required robustness our position analysis procedure should also be able to perform the analysis automatically, and all positions belonging to a particular closure mode should be found. Furthermore, it should not be necessary to define one of the position variables as a prescribed input or driver. In order to achieve these goals we propose the following procedure which has been proven to be very successful.

We assume that it is possible to assemble the mechanism in the first place and that the position parameters (φ_1, φ_2, φ_3 and φ_4 for the four–bar mechanism) of one assembly are already known. Then we can compute in the first step all possible neighboring configurations of the current position (denoted by $\underline{x}_i = [\varphi_1, \varphi_2, \varphi_3, \varphi_4]$) by consecutively adding and subtracting an increment ($\Delta\varphi$) to each of the position variables. For the planar four–bar mechanism we therefore obtain up to four neighboring configurations for the variation of each of the position variables (see Figure 7) and a total of at most sixteen neighboring positions ($\underline{x}_i^{(k)}$, $k = 1,\ldots,m$; $m \le 16$). However, of all the possible neighboring positions only those of which the magnitude of all changes of the position parameters do not exceed the increment, are considered for further selection. In the next step we try to figure out which of the neighboring positions can be viewed as the best continuation of the previous position sequence. We therefore compute the normalized vector dot products

$$\frac{(\underline{x}_i^k - \underline{x}_i)\circ(\underline{x}_i - \underline{x}_{i-1})}{|\underline{x}_i - \underline{x}_1|} \qquad (1)$$

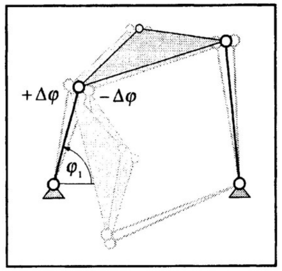


Figure 7. The four neighbor assemblies obtained by incrementing and decrementing the position variable φ_1 by $\Delta\varphi$.

for all k, and select this neighboring position $\underline{x}_i^{(k)}$ for which this product is maximized (see Figure 8). This procedure of finding the best possible neighboring position is repeated until the current position $\underline{x}_i$ arrives within the neighborhood of the position first computed $\underline{x}_1$ (all coordinates of $|\underline{x}_i - \underline{x}_1|$ do not exceed $\Delta\varphi$) for the current closure mode. Now, if

$$(\underline{x}_2 - \underline{x}_1)\circ(\underline{x}_i - \underline{x}_1) < 0 \text{ and } (\underline{x}_2 - \underline{x}_1)\circ(\underline{x}_{i+1} - \underline{x}_1) > 0 \qquad (2)$$

is satisfied, then the current position $\underline{x}_i$ is the last position computed on the current closure mode. Otherwise we arrived at a "double position" witch means that we have to continue the procedure until Equ. 2 is satisfied. The actual procedure in the *Four–Bar* software uses a a more or less arbitrary value of 3° for the increment $\Delta\varphi$.

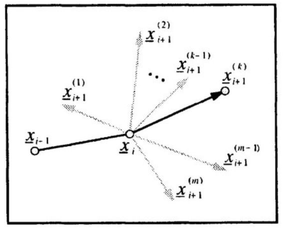


Figure 8. Selection of the best neighboring position.

3.2 Strategies used for synthesis tasks

The coupler–curve synthesis, the straight–line–envelope synthesis, and the transmission–function are treated in more or less the same way. Each of these synthesis problems is set up as an optimization problem with the sum of the squared distances between the prescribed points and the corresponding closest points on the respective curve (coupler–curve, envelope, transmission–function) as the objective function. These objective functions on the other hand depend on the essential design parameters of the mechanism being synthesized. The goal of course is to minimize these objective functions. In the program, the above mentioned curves are each represented by polygons (one polygon for each closure mode). The shortest distance between a given point and a polygon is computed as the minimum of the distance to the vertex closest to this point and the smallest distances to the adjacent polygon–segments. Depending on the chosen synthesis option (see Figure 5) either all polygons (closure mode dose not matter), just one of the polygons (points must belong to one closure mode), or just parts of one polygon (successive points must lie on the related remaining parts of the polygon) are searched for the vertex of smallest distance. Restrictions related to fixed design parameters can easily be taken into account by simply reducing the number of unknowns of the optimization problem.

For the task of the coupler–position synthesis the possible coupler positions of a mechanism are represented by a succession of point pairs, the positions of the two moving pivots. A given coupler position is also denoted by a point pair. The distance between a given coupler position and one position of the coupler motion is then measured by the sum of the distances of the corresponding point pairs. The coupler–position synthesis problem can therefore be treated in a similar way as the other synthesis problems.

Synthesis problems usually possess a finite or even infinite number of solutions. Consequently the interaction with the user is necessary in order to find a desired solution.

4 Conclusions

A new and rather complete educational computer program for the analysis and the synthesis of planar four–bar linkages, the planar slider–crank mechanism, and the Cardan mechanism was introduced. The software uses a graphical interface, is simple to handle and is highly interactive. Topics treated by the program are: Visualization of the coupler–curve, pole–curves, envelope–of a straight line, transmission functions, Roberts theorem. For a single position or during animation the velocities and accelerations of points on the coupler, the circles of Bresse and De La Hire, and the circle of curvature of the coupler can be shown. Additionally, coupler–curve synthesis, coupler–position synthesis, and input–output synthesis can be performed. The software is available via the internet, it runs on all major computer platforms, and is free of charge.

References:

Bottema, O., Roth, B., Theoretical Kinematics, Dover Publications Inc., New York, 1990.

Gosling, J., Joy, B., Steele, G., The Java™ Language Specification, Addison–Wesley, Reading, Massachusetts, 1996.

Hiesleitner, N., Entwicklung einer Software zur graphischen Darstellung der Kinematik des Viergelenks, Diploma–Thesis, University of Technology, Graz, 1991.

Roberts, S., On three–bar–motion in plane space, Proceedings of the London Mathematical Society, VII, 1875.