Kinematic Synthesis for Finitely Separated Positions Using Geometric Constraint Programming

Edward C. Kinzel
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907

James P. Schmiedeler
Department of Mechanical Engineering,
The Ohio State University,
Columbus, OH 43210

Gordon R. Pennock
Fellow ASME
School of Mechanical Engineering,
Purdue University,
West Lafayette, IN 47907

This paper presents an original approach to the kinematic synthesis of planar mechanisms for finitely separated positions. The technique, referred to here as geometric constraint programming, uses the sketching mode of commercial parametric computer-aided design software to create kinematic diagrams. The elements of these diagrams are parametrically related so that when a parameter is changed, the design is modified automatically. Geometric constraints are imposed graphically through a well-designed user interface, and numerical solvers integrated into the software solve the relevant systems of equations without the user explicitly formulating those equations. This allows robust algorithms for the kinematic synthesis of a wide variety of mechanisms to be "programmed" in a straightforward, intuitive manner. The results provided by geometric constraint programming exhibit the accuracy and repeatability achieved with analytical synthesis techniques, while simultaneously providing the geometric insight developed with graphical synthesis techniques. The key advantages of geometric constraint programming are that it is applicable to a broad range of kinematic synthesis problems, user friendly, and highly accessible. To demonstrate the utility of the technique, this paper applies geometric constraint programming to three examples of the kinematic synthesis of planar four-bar linkages: Motion generation for five finitely separated positions, path generation for nine finitely separated precision points, and function generation for four finitely separated positions. [DOI: 10.1115/1.2216735]

Keywords: kinematic synthesis, computer-aided design, geometric constraint programming, graphical techniques, motion generation, path generation, function generation

1 Introduction

Kinematic synthesis is the determination of the parameters that define a mechanism such that it will perform a desired task [1–3]. Tasks are commonly identified as belonging to one of three categories: Motion generation, path generation, and function generation [4–6]. Motion generation is the controlled translation and rotation of a moving reference frame relative to a fixed reference frame. Path generation is the controlled translation of a single point in a moving frame relative to a fixed frame without regard to the orientation of the moving frame. Function generation is the controlled correlation of an output motion to an input motion. The methods for prescribing the performance of a desired task are also commonly divided into three categories: Finitely separated positions, infinitesimally separated positions, and multiply separated positions. The categorizations of both the tasks and the methods for prescribing the performance of those tasks are based primarily on the types of constraints associated with them.

Traditionally, there are three approaches to the kinematic synthesis of planar mechanisms: Trial and error, graphical techniques, and analytical techniques. The most rudimentary approach is trial and error [7,8], wherein the specified output, a generated motion, path, or function, is plotted and the mechanism parameters are adjusted until the output satisfies the constraints of the prescribed performance. The behavior of most mechanisms, however, is highly nonlinear, so this approach tends to be both difficult and nonintuitive even when augmented with aids such as an atlas of possible solutions. For this reason, more deterministic synthesis procedures have been developed, particularly since the landmark work of Burmester [9]. One such approach that generates solutions relatively quickly is to perform reasonably simple graphical constructions based upon the constraints of the prescribed performance. These graphical techniques tend to facilitate the development of intuition and geometric insight, but when executed by hand, can suffer from poor accuracy. The other deterministic approach is to formulate the constraints and solve the problem analytically so that the result is both accurate and repeatable. While these analytical techniques are favorable for addressing complex problems, they can produce seemingly "black-box" solutions, so they are generally not effective for developing intuition. In addition, analytical techniques are usually limited to a specific type of synthesis problem.

In recent years, a number of dedicated mechanism synthesis computer applications have been developed, generally to implement the analytical synthesis techniques with some form of graphical user interface. Examples include KINSYN [10], MECHSYN [11], LINCAGES [12], RECSYN [13], SYNTRA [14], IMSC [15], SYMECH [16], CH [17], SAM [18], and WATT [19]. The survey article written by Erdman [20] provides a comprehensive list of mechanism synthesis packages using interactive graphics. Another approach is to create mechanism synthesis codes specific to existing computer applications. Two examples are the MATLAB code provided by Waldron and Kinzel [21] and the TK Solver code provided by Norton [22]. Similar to the specialized applications, separate modules in these codes have been developed for each type of mechanism to be synthesized.

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received April 20, 2005; final manuscript received October 25, 2005. Review conducted by Thomas R. Chase.

This paper details a novel approach to the kinematic synthesis of planar mechanisms which the authors refer to as geometric constraint programming. The technique leverages the capabilities of existing parametric design software to capture the advantages of both graphical and analytical approaches. The technique can be implemented in most modern parametric design software, so it is not the creation of a new stand-alone package or a computer code in the traditional sense, nor is it the direct implementation of classic graphical techniques within a CAD package. Rather, geometric constraint programming involves a versatile use of the constraint-based sketching mode that is available in most modern parametric design software. Kinematic diagrams are constructed in this sketching mode to enforce the geometric constraints that map the prescribed performance input information to the synthesized mechanism. As an example of this mapping, consider the synthesis of a planar four-bar linkage for motion generation through five finitely separated positions. The coupler link is constrained to have a specified position and orientation at five prescribed locations. The mapping, through the constraints, of this single set of input information to the synthesized four-bar linkage is not one-to-one. In fact, the mapping is one-to-a-finite-number since there may be as many as six different four-bar linkages that satisfy the geometric constraints [2,3].

Most kinematic synthesis problems, however, are underconstrained, as in the case of synthesizing a four-bar linkage for motion generation through four finitely separated positions. In this example, the mapping from a known set of inputs to a suitable four-bar linkage is one-to-infinity. This is not to say that the designer has greater difficulty in obtaining a solution to an underconstrained problem. Rather, with infinitely many satisfactory four-bar linkages from which to choose, the designer has the freedom to make arbitrary selections or formulate an optimization problem to fully define the linkage. Experience with, and a developed intuition for this type of problem can greatly facilitate solving the optimization problem and ultimately lead to a better design. Geometric constraint programming enables the development of this intuition in the kinematic synthesis of mechanisms without sacrificing the accuracy of the resulting solution.

The dramatic improvement in constraint management techniques within the past decade (for example, [23]) is the key development that has made geometric constraint programming feasible. The constraint manager in the parametric design software organizes the underlying nonlinear equations so that they can be efficiently solved by the program's iterative equation solver. The user has no access to the constraint manager, nor is the user aware of how or when it is functioning behind the graphical user interface. The ability of the software to solve an increasingly large number of nonlinear equations very quickly has been developed specifically to facilitate the design of solid models. To the best of the authors' knowledge, this paper is the first published account of how these powerful tools can be directly applied to the synthesis of planar mechanisms.

The description of geometric constraint programming in this paper does not represent the development of a new theory for the synthesis of planar mechanisms. While this fact may initially appear disconcerting, the true novelty and significance of the new synthesis technique is precisely that it does not require an advanced theory for the solution of difficult problems, for example, motion generation through five finitely separated positions. The relative simplicity of the synthesis technique is, in fact, its elegance. Certainly, an understanding of the underlying theory facilitates efficient application of the technique, but it is in no way a fundamental requirement. Similarly, because the foundation of the technique is simple imposition of geometrical constraints, it is universally applicable to the synthesis of planar mechanisms, even those for which no rigorous theory is available. Therefore, the fundamental contribution of this paper is to introduce a technique for the synthesis of planar mechanisms that is: (i) Applicable to a broader range of problems; (ii) more helpful in the development of intuition; and (iii) accessible to a larger number of mechanism designers than existing approaches.

The remainder of this paper is organized as follows. Section 2 describes general implementation of the geometric constraint programming technique and identifies the required constraints that are available in parametric design software. Section 3 applies the technique to the synthesis of a four-bar linkage for motion generation through five finitely separated positions, and Sec. 4 applies it to the synthesis of a four-bar linkage for path generation through nine finitely separated precision points. To highlight the application of geometric constraint programming to an underconstrained problem, Sec. 5 presents the synthesis of a four-bar linkage for function generation through four finitely separated positions. Section 6 addresses the applicability of the technique to kinematic synthesis of more complex linkages and of linkages that satisfy nonstandard constraints. Finally, Sec. 7 provides a summary and identifies additional features of parametric design software that would improve the application of geometric constraint programming to the kinematic synthesis of mechanisms.

2 Geometric Constraint Programming

As the speed of computer processing increases, the challenge of many computer-aided mechanism design problems is not so much solving the governing nonlinear equations as much as it is formulating the problem in an intuitive manner. With such a formulation, the method of solution helps the designer to develop geometric insight. Geometric constraint programming (henceforth denoted as GCP) is a new technique that facilitates the intuitive formulation of kinematic synthesis problems within the sketching mode of parametric design software. In its most basic form, GCP entails drawing in a single planar diagram one mechanism to satisfy each individual constraint of the synthesis problem and then enforcing the equivalence of all the parameters of the numerous mechanisms. For example, in the problem of motion generation for five finitely separated positions, five distinct four-bar linkages can be drawn such that each one of the five distinct coupler links is a coincident with one distinct precision position. Then, the dimensions of all five four-bar linkages are constrained to be equal, and the corresponding center points of all five linkages are constrained to be coincident. The synthesis procedure is thereby "programmed" through application of geometrical constraints within parametric design software. A similar problem can be solved directly with the same diagram by simply repositioning the specified locations of the coupler link. A key feature of GCP is that it can be applied more elegantly to kinematic synthesis problems wherein the designer can exploit the known motion of certain points to avoid constructing numerous individual mechanisms. This is illustrated in the synthesis examples presented in Secs. 3 and 4.

As a new technique for kinematic synthesis via constraint-based sketching, GCP has a number of important advantages compared to commercially available synthesis software:

(i) GCP can be applied to a broader range of synthesis problems, allowing more diverse problems to be posed in a geometric manner. For the purpose of illustration, this paper demonstrates the application of GCP to three examples of the kinematic synthesis of planar four-bar linkages. It is important to note, however, that the technique is completely general, so it can be used to synthesize much more complex planar mechanisms. Unlike existing software packages that include separate modules for each type of mechanism to be synthesized, GCP can be applied in the same manner regardless of the type of mechanism. The differences lie only in the constraints that are applied, not the mechanism types. Furthermore, GCP can be applied in a straightforward manner to synthesis problems with nonstandard constraints or with the traditionally separate categories of

tasks integrated into a single problem. For example, motion generation and path generation can be combined to constrain the moving reference frame to pass through a series of points, only some of which have a specified orientation of the reference frame associated with them.

(ii) GCP is user friendly. The parametric nature of modern CAD software enables the designer to redefine the inputs and automatically produce the corresponding new mechanism without any repetition of the GCP procedure. This stems from the constraint mapping between the specified inputs and the synthesized mechanism. The formulation often lends itself to a more intuitive problem solving process, and since the user interface is entirely graphical, the technique provides more of a geometric insight than does a purely analytical solution. Unlike graphical techniques executed by hand, however, the accuracy and repeatability using GCP are consistent with the accuracy and repeatability of the computer. Furthermore, it is relatively easy to incorporate GCP into an integrated mechanism design and analysis loop. Within the software, the sketch of the synthesized mechanism can be used to automatically generate a solid model that can then be exported for a dynamic force analysis or a finite element analysis.

(iii) GCP is highly accessible to the designer. Because of the user friendly characteristics, GCP is accessible to designers who do not have a formal training in kinematics or an expertise in the kinematic synthesis of mechanisms. This does not call into question the rigor of the technique. On the contrary, the approach exploits existing features of parametric design software to rapidly perform the computationally intensive aspects of synthesis, leaving the designer to focus on intuitive, graphical constructions. Also, unlike dedicated synthesis packages, parametric design software is readily accessible to the designer of mechanisms in industry and most mechanical engineering students in academe. Therefore, GCP offers enhanced synthesis capability without additional software cost.

In GCP, the user "programs" the mathematical equations governing the problem by imposing geometric constraints. The technique is particularly efficient for kinematic synthesis problems when the user can impose the eight constraints that are listed below, all of which are available in some form in the sketching mode of most modern parametric design software. Typically, they appear as graphical icons, each denoted with a symbol that indicates the type of constraint. For example, in SOLIDEDGE [24], which was used to generate the figures in this paper, the sketching mode is known as the "Draft" mode, and the constraints are found in the "Relationships" toolbar. (UNIGRAPHICS [25] is the parent software to SOLIDEDGE, so its sketching mode and constraints are quite similar.) In PRO/E [26], they appear in the "Constrain" option under the "Sketch" pull-down menu in the "Sketch" mode. In SOLIDWORKS [27], they are found in the "Sketch Relations" toolbar of the "Sketch" mode. In INVENTOR [28], they are grouped in a single pull-down menu within the "Drawing Sketch panel" of the "Sketch" mode in the "Drawing" environment. The user clicks on the constraint icon with the mouse and then clicks on the relevant entities, such as lines, to which to apply the constraint. The relationship imposed by the constraint, then, is maintained until the user might possibly choose to delete it. The eight constraint names identified here are simply meant to provide a consistent convention for commonality across different software packages.

(i) *Strong Connect*—A single point is constrained to coincide with another point. Throughout this paper, it is assumed that this constraint is automatically applied when a line is drawn from an endpoint or the midpoint of another line. A revolute joint is defined in this manner. In some CAD packages, concentricity is a separate constraint, but it can be enforced by the Strong Connect constraint.

(ii) *Weak Connect*—A single point is constrained to lie on a line or a curve. Throughout this paper, it is assumed that this constraint is automatically applied when a line is drawn from a point on another line or curve other than an endpoint or the midpoint. A pin-in-slot joint is defined in this manner. In some parametric CAD packages, a single tool is used for both the Strong Connect constraint and the Weak Connect constraint.

(iii) *Dimension Lock*—A linear dimension or an angular dimension is constrained to have a fixed quantity. A binary link is defined by locking the length of the line segment. Also, a ternary link, quaternary link, etc., is defined by locking the lengths and relative angles of the line segments that form the link. This constraint differs from the general dimensioning of unconstrained lengths and angles, which simply measures these variable quantities.

(iv) *Position Lock*—A point, line, or curve is constrained to be fixed to the ground (i.e., a fixed reference frame). A center point for a linkage can be defined in this manner.

(v) *Equality*—Two lines (or arcs) are constrained to be of equal length (or radius).

(vi) *Parallelism*—A line is constrained to be parallel to another line. This can be used with the Strong Connect constraint to form a collinear constraint. In some CAD packages, collinearity is a separate constraint. Also in some CAD packages, the horizontal and vertical constraints are special cases that constrain a line to be parallel to one axis of a fixed reference frame.

(vii) *Perpendicularity*—A line is constrained to be perpendicular to another line.

(viii) *Tangency*—A curve or a line is constrained to be tangent to another curve or line. A higher pair, such as the connection between the cam and follower of a cam-follower mechanism, is defined in this manner.

There is a mathematical formulation for each of these constraints that is used by the constraint manager in the software, but the designer only needs to impose the constraints through the graphical user interface. The synthesis examples in Secs. 3–5 of this paper require only the first five constraints.

To apply GCP to a general kinematic synthesis problem, the first step is to determine the appropriate task requirements, which may include the positions of a rigid body, precision points on a path, and functional relationships between the input and the output of the mechanism to be synthesized. This step may also involve sketching an initial draft of the mechanism's installation to locate obstructions and suitable positions for center points. The second step is to select the simplest mechanism that will satisfy the combination of task requirements. A number of these mechanisms are then drawn, each one satisfying an individual task requirement. These conditions are imposed by applying the appropriate constraints from the list of eight provided above. The numerous mechanisms are then constrained to be identical through subsequent application of the Equality constraint to the link lengths and the Strong Connect constraint to the fixed points. If this step fails, the mechanism is most likely overconstrained, and task requirements can be eliminated or a higher-order mechanism can be selected without reformulating the problem. If, on the other hand, the problem remains underconstrained, the Dimension Lock constraint or Position Lock constraint can be applied to completely define the final mechanism.

Most CAD packages enable the designer to draw a copy of this final mechanism that can be moved through its range of motion to

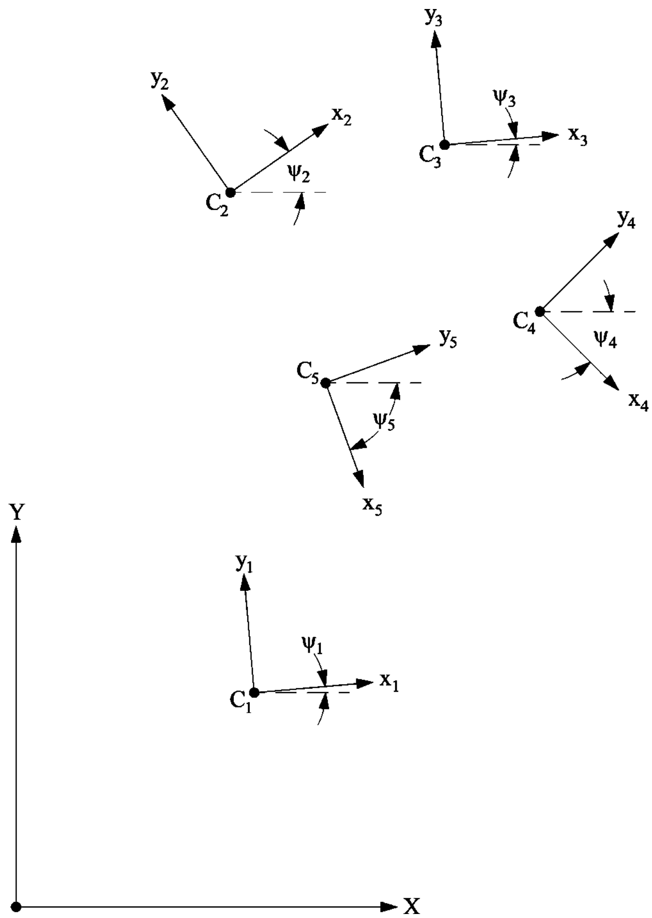

Fig. 1 Five finitely separated positions for motion generation

validate the design and enable visualization. This provides a means to check for the problems of circuit, branch, and order defects and the option to add graphical analysis. (Branch defect refers to a possible design that satisfies all of the prescribed constraints at each of the precision positions but cannot be moved continuously between these positions without being taken apart and reassembled. Order defect refers to a synthesized linkage that can reach all of the precision positions but not in the desired order.) Ultimately, because the diagram is in effect a program that maps the task requirements to the mechanism parameters through the imposed constraints, it can be saved and reused to solve similar problems by simply redefining the task requirements.

For purposes of illustration, the following three sections provide examples of the application of GCP to the synthesis of planar four-bar linkages for motion generation, path generation, and function generation. In the first two examples, the knowledge that two points in the coupler links trace circular paths is leveraged to solve the problem more efficiently than through application of the more general approach described above. This highlights the flexibility of the technique and emphasizes that a pre-existing geometric understanding of the kinematic synthesis problem can streamline the solution strategy.

3 Motion Generation for Five Finitely Separated Positions

Figure 1 shows five finitely separated positions of a moving Cartesian reference frame, denoted as x_i and y_i $(i=1,\ldots,5)$. The five locations of the origin of the moving frame, denoted as C_i, are chosen so that the path of this point approximately traces the letter "P." The five positions of the moving frame place a sufficient number of constraints on the system to map the set of inputs to a finite number of four-bar linkages. Note that, in general, more than one four-bar linkage will satisfy the constraints; however, each one is completely defined by the five positions.

Table 1 Cartesian coordinated and orientations of the five specified positions

Position i	X_i	Y_i	ψ_i (deg.)
1	50	45	5
2	45	150	35
3	90	160	5
4	110	125	−45
5	65	110	−70

The positions and orientations of the moving frame are most conveniently defined relative to a fixed Cartesian reference frame, denoted as X and Y in Fig. 1. The Cartesian coordinates X_i and Y_i of the origin and the orientation angle ψ_i of the moving frame (i.e., the input constraints) are listed in Table 1. This method enables the designer to redefine the input positions quickly and easily by simply changing the dimensions. If changes are made after the four-bar has been synthesized, the linkage will dynamically update to satisfy the altered constraints. Note that the linear dimensions can be presented without units because they can be scaled uniformly to any convenient scale. An alternative approach for defining the five positions is to draw the moving reference frames without dimensions and reposition them with the mouse. The moving frames can then be fixed relative to the fixed reference frame by applying the Position Lock constraint. With this approach, the Position Lock constraints can be released after the synthesis procedure is complete, allowing the designer to reposition the moving frames with the mouse while the synthesized mechanism dynamically updates to satisfy the constraints. Using either approach to define the five positions, an approximate solution can be developed initially, and the positions can then be adjusted so as to refine the final solution.

With the positions fully defined, the GCP procedure is employed to simultaneously locate a circle point (denoted A) and center point (denoted O_A) pair. The first step is to draw a portion of the coupler link, a triangle defined by points A, C, and D. Point D need only be a consistent point in the moving frame, but for convenience here, it is chosen to be a point on the x-axis such that the distance CD has a fixed value. Point A can be chosen arbitrarily for one position, say position 1, and congruent triangles $\Delta A_iC_iD_i$ can be constructed for each of the other four positions, as shown in Fig. 2. This construction is achieved by locating each point A_i in roughly the proper position and then applying Equality constraints to the two corresponding sides of the triangle $(A_iC_i = A_1C_1, A_iD_i = A_1D_1)$. The lengths of line segments A_iC_i and A_iD_i are not explicitly dimensioned. Care must be taken to avoid selecting an initial point A_i that, after application of the Equality constraints, yields a triangle that is not congruent due to a reflection about the line C_iD_i. If this mistake is made, it can be corrected by redrawing the triangle in the proper orientation. Congruency of the triangles can be strictly enforced by constraining the angles $\angle A_iC_iD_i$ to all be equal or by introducing a second set of triangles $\Delta A_iC_iE_i$, wherein CE has a fixed length and Equality constraints are applied to the A_iE_i segments. Experience has shown, however, that adding such extra constraints can slow the computer processing.

At this stage, each point A_i can still be repositioned by clicking and dragging with the mouse, but repositioning any one A_i will automatically affect the locations of the other A_is to satisfy the constraints. Without loss of generality, the point A is chosen as a circle point of the four-bar linkage. The ability to modify the positions of the five A_i's without violating the established Equality constraints is then exploited to locate the corresponding center point O_A. An arbitrary circle is drawn, and the Weak Connect

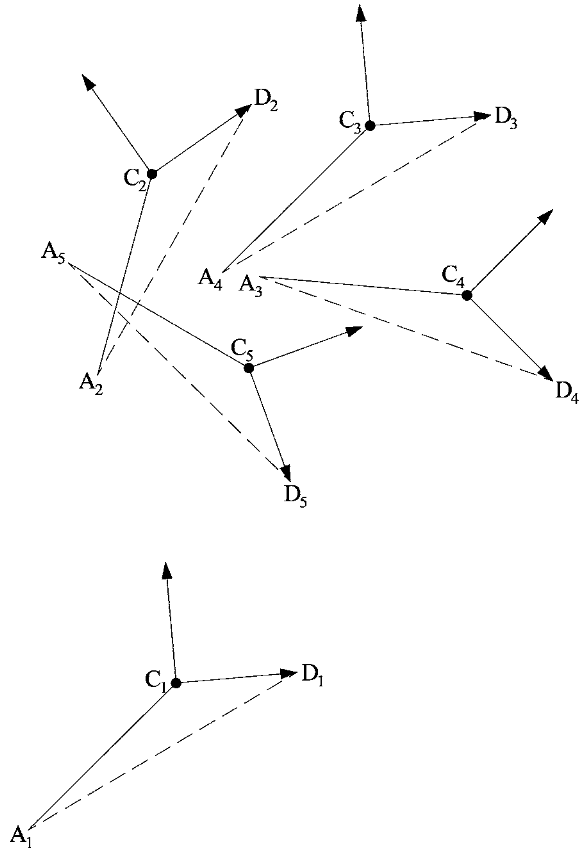

Fig. 2 Congruent triangles $\Delta A_iC_iD_i$

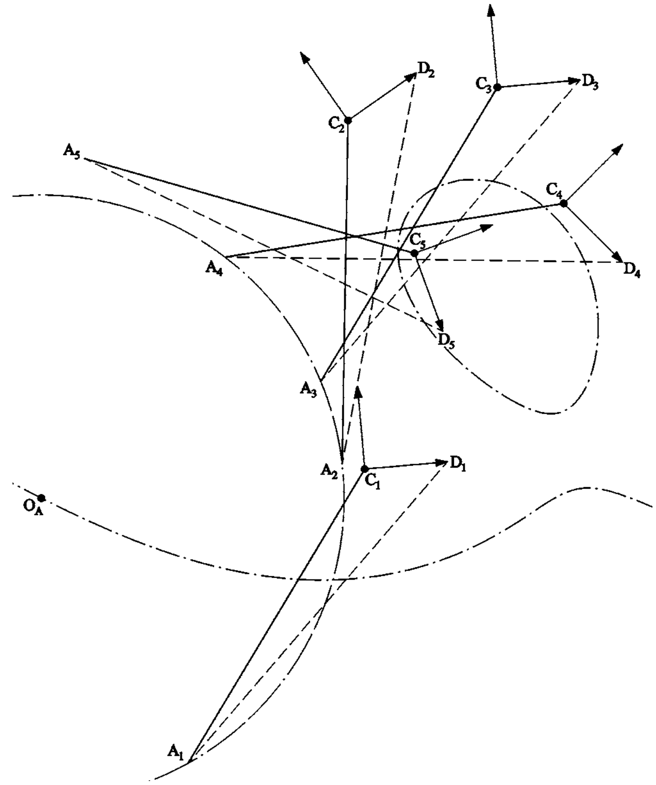

Fig. 3 The center point curve for positions 1–4

constraint is applied to constrain any four of the five A_i's to lie on the circumference of the circle. See Fig. 3. Application of these four constraints will, in general, change the position of the center of the circle, the diameter of the circle, and the locations of all five A_i's. This can be observed by comparing the triangles in Figs. 2 and 3. The center of the circle, however, is not uniquely located by the imposed constraints, as it can lie anywhere on the center point curve defined by the four positions. Although there is no need to plot the center point curve in practice, a portion of this curve for positions 1–4 is shown in Figs. 3 and 4 simply for illustrative purposes. If the triangles $\Delta A_iC_iD_i$ are not strictly constrained to be congruent, as described in the previous paragraph, care must be taken that as the locations of the A_i's change through-out this process, the change does not amount to a reflection about the line segments C_iD_i. In such a case, the triangles are no longer congruent, and a solution cannot be obtained.

The fifth A_i will lie on the circle only when the center of the circle coincides with a point of intersection between the center point curve and a different center point curve defined by the fifth position and any three of the original four positions. In this example, there are four real intersections of any two center point curves in addition to the three poles that are common to both curves; therefore, there are four possible center points, the Cartesian coordinates of which are listed in Table 2. In order to facilitate convergence of the numerical solver in the CAD package, the center of the circle is dragged along the center point curve to locate it very near such an intersection point before the Weak Connect constraint is applied to constrain the fifth A_i to lie on the circle. Any movement of the center of the circle is already constrained to be along the center point curve (normally not plotted), and the designer can easily observe the proximity to an intersection point by noting the proximity of the circle to the point A_i not yet lying on it. Once the final Weak Connect constraint has been applied, O_A, a possible center point, is coincident with the center of the circle, and the points A_i are the five positions of the corresponding circle point. This step is shown in Fig. 4.

The process is repeated for a second circle point B by drawing five more triangles $\Delta B_iC_iD_i$ (similar to the previous construction for A_i). The resulting center point O_B is shown in Fig. 4. In this process, the line AB in the coupler link could be drawn and Equality constraints applied to the line segments A_iB_i to explicitly enforce triangle congruence. As in dimensioning the angles $\angle A_iC_iD_i$, though, applying such additional constraints can slow the processing. With the center point O_B located in a manner similar to that of O_A, a four-bar linkage is completely defined. Figure 5 shows the four-bar linkage for which the center points O_A and O_B are chosen to be O_1 and O_4, respectively, from Table 2. The figure also shows the five design positions and the coupler curve traced by point C. The dimensions of the synthesized four-bar linkage are $O_AA=106.24$, $AB=135.14$, $O_BB=202.82$, $AC=101.48$, and $BC=146.89$. Of the five other four-bar linkages that can be obtained from combinations of the four possible center points, only one exhibits neither branch nor order defects. This four-bar linkage is formed by choosing O_A and O_B to be O_1 and O_2, respectively. The dimensions of this linkage are $O_AA=106.24$, $AB=84.95$, $O_BB=96.30$, $AC=101.48$, and $BC=146.89$.

The sequential application of constraints allows the designer to map the five input positions to the synthesized four-bar linkage parameters without explicitly writing or directly solving the equations. As with any kinematic synthesis technique, there are several issues to consider. First, a practical issue arises when the center point curve has two branches, as does the curve in Figs. 3 and 4. As the designer drags the center of the circle along the center point curve to approach a five-position center point, a rather significant deviation from one branch of the curve is required for the numerical solver of the CAD package to converge to a point on the other branch of the curve. Therefore, to locate all possible five-position center points when the curve is dual-branched, the designer must take care to inspect both branches. Secondly, a

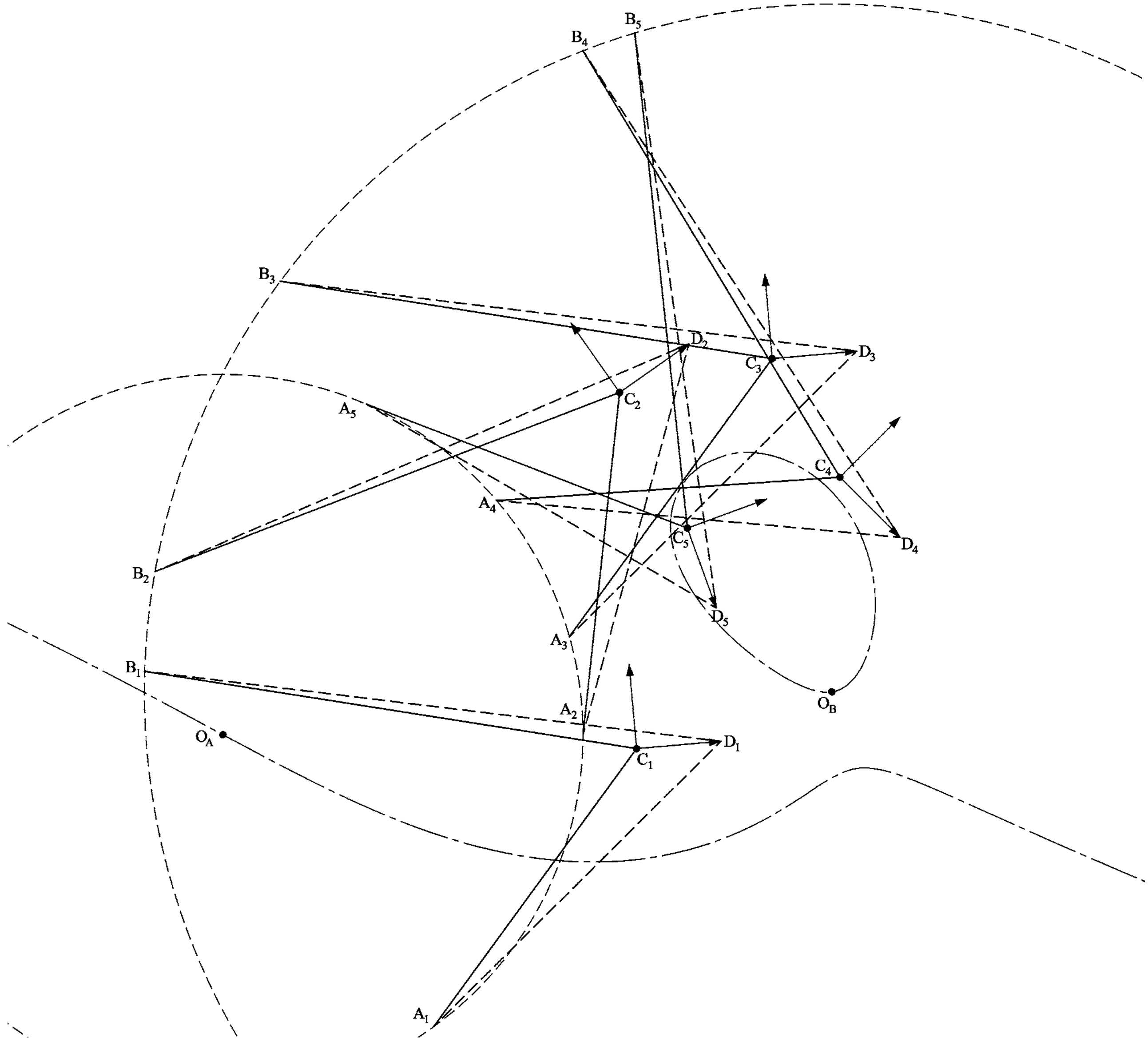

Fig. 4 Center points O_A and O_B and circle points *A* and *B*

given set of constraints sometimes allows for no viable solution. The flexibility of the approach, however, enables a designer to iteratively modify the task requirements, the precision positions in this example, and pursue a viable solution until one is found. In the event that the solution exhibits a branch or order defect, a similar approach of modifying the positions and evaluating the resulting mechanisms can be pursued. In an underconstrained problem, such as motion generation through four finitely separated positions, the designer has the flexibility to relocate the center points without altering the precision positions. In all of these cases, the intuition developed by formulating the problem geometrically and working directly with the graphical interface facilitates resolution of the relevant issue.

Table 2 The Cartesian coordinates of the four possible center points

Center points	*X*	*Y*
O_1	−72.03	49.11
O_2	75.01	80.76
O_3	100.52	126.65
O_4	107.62	61.76

4 Path Generation for Nine Finitely Separated Points

Nine finitely separated precision points are defined here such that the desired coupler curve is similar to the "P" shaped curve of the previous example. See Fig. 6. Again, the points can be defined by specifying the Cartesian coordinates X_i and Y_i or by simply applying the Position Lock constraint to each point after it is drawn. Once the solution procedure is complete, the locations of the points can be modified by changing the dimensions or releasing the Position Lock constraint and moving them, enabling the designer to redefine the problem while the synthesized four-bar linkage dynamically updates to satisfy the constraints. The Cartesian coordinates of the nine precision points, denoted as C_i $(i=1,\ldots,9)$, are listed in Table 3.

The synthesis procedure for the path generation is similar to that for the motion generation. The primary difference is that with no constraint on the orientation of the coupler link, both circle point/center point pairs of the four-bar linkage must be located

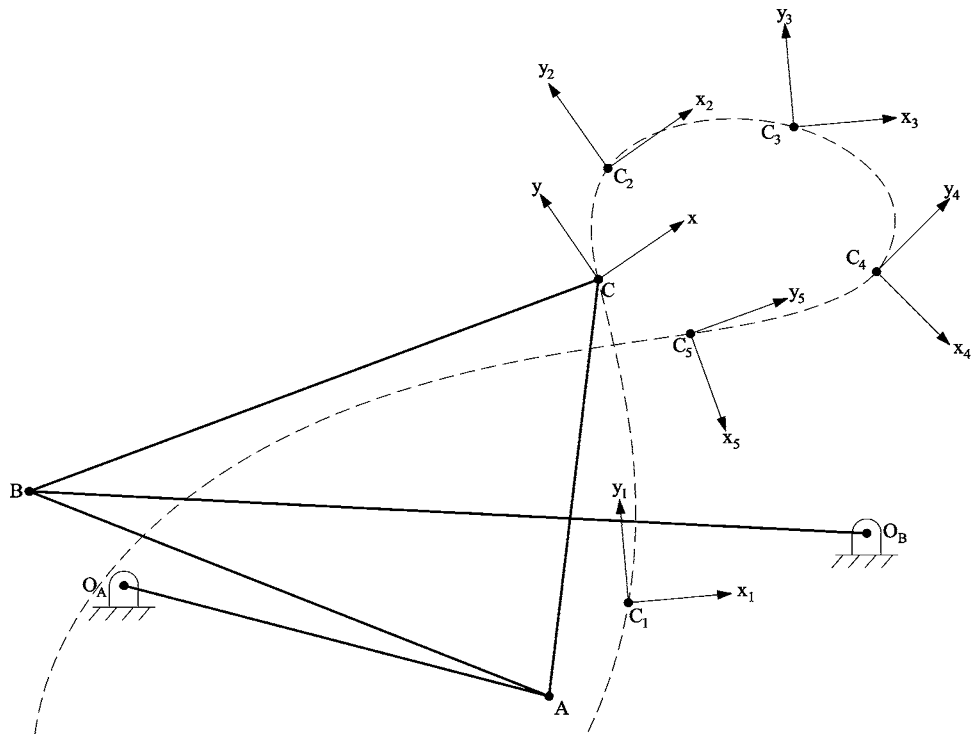

Fig. 5 The synthesized four-bar linkage and the coupler curve

simultaneously. The procedure is initiated by drawing a portion of the coupler link, a triangle ΔABC in which the points A and B are chosen arbitrarily for one position. The nine triangles $\Delta A_iB_iC_i$ are made congruent by applying Equality constraints to their respective sides in a manner similar to that of the motion generation

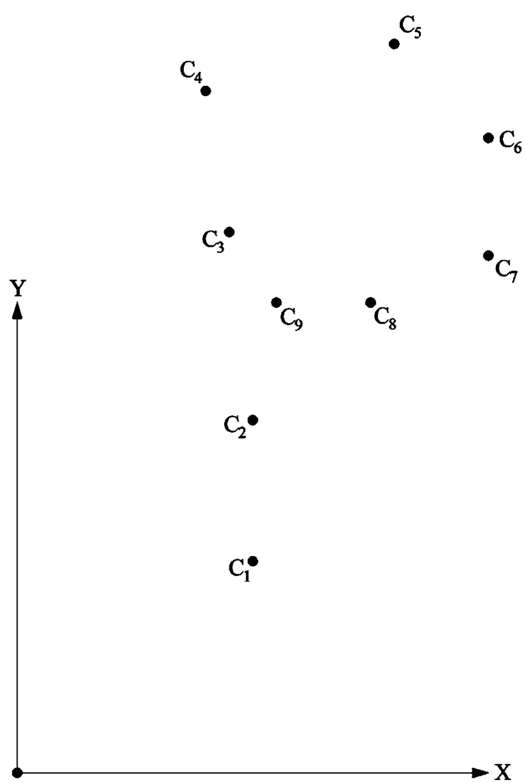

Fig. 6 Nine finitely separated precision points for path generation

example. Again, care must be taken to ensure that the triangles are truly congruent, since adding extra constraints for nine triangles would dramatically reduce the computational speed. Two arbitrary circles are drawn, and the Weak Connect constraint is applied to constrain the nine points A_i to lie on the circumference of one of these circles. Likewise, the nine points B_i are constrained to lie on the circumference of the other circle. This construction is shown in Fig. 7. Application of these constraints will in general change the locations of the two circles, the diameters of the circles, and the shape of the triangles $\Delta A_iB_iC_i$. To facilitate convergence of the numerical solver in the CAD package, the designer may need to move the centers of the circles close to their final locations before enforcing the Weak Connect constraints on the last few points. The center points of the four-bar linkage are located at the centers of the two circles after all of the constraints have been applied. The synthesized four-bar linkage for this example and its coupler curve are shown in Fig. 8. The center points are O_A $(-46.00, 94.31)$ and O_B $(52.20, 71.17)$, and the dimensions of the four-bar linkage are $O_AA=77.50$, $AB=40.00$, $O_BB=49.65$, $AC=71.86$, and $BC=45.73$.

In practice, it may be more common to synthesize a four-bar linkage using fewer than nine precision points, in which case some of the link dimensions are established by the designer using the Dimension Lock constraint. After an initial linkage is synthe-

Table 3 Cartesian coordinates of the nine precision points

Position i	X_i	Y_i
1	50	45
2	50	75
3	45	115
4	40	145
5	80	155
6	100	135
7	100	110
8	75	100
9	55	100

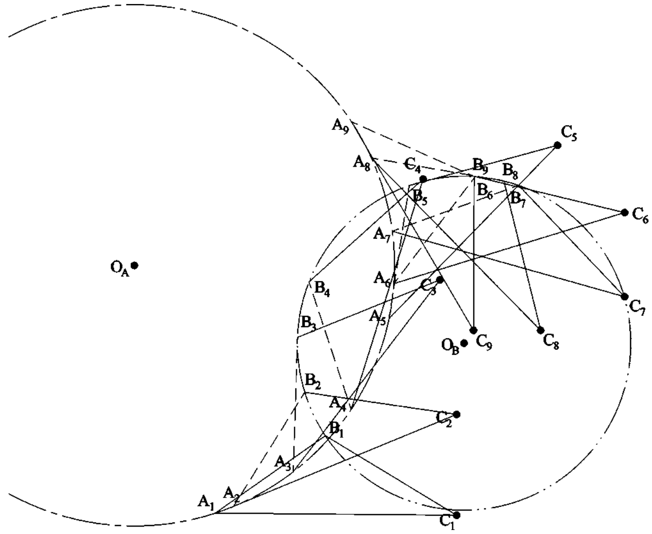

Fig. 7 The construction to locate the center points and circle points

sized for a relatively small number of points, the final linkage can be fine-tuned by adding precision points where needed to better approximate the desired coupler trajectory. Another advantage of GCP is that it facilitates the generation of the cognate linkages as defined by the Roberts-Chebychev theorem [3,9,21]. By simply constraining triangles to be similar, the designer can quickly generate two additional linkages that trace the same coupler curve but have different center points.

5 Function Generation for Four Finitely Separated Positions

To highlight the application of GCP to an underconstrained synthesis problem, this section presents the design of a four-bar linkage for four finitely separated positions of the turning links. The linkage is synthesized to approximately generate the function $y = x^2$. The required output is defined as $y = \theta_4$, and the input is defined as $x = \theta_2/10$, with the scaling factor of 10 employed simply to increase the range of motion over which the linkage is valid. The desired range of motion is from the initial position $x_0 = 1$ $(\theta_{2,i} = 10 \text{ deg})$ to the final position $x_f = 9$ $(\theta_{2,f} = 90 \text{ deg})$. The four precision positions of the input angle θ_2 are chosen to be the roots of a fourth-order Chebychev polynomial. These input angles

Table 4 The input and output angles for the four precision positions

Position i	Input angles $\theta_{2,i}$(deg.)	Output angles $\theta_{4,i}$(deg.)
1	13.0448	1.7017
2	34.6927	12.0358
3	65.3073	42.6505
4	86.9552	75.6120

and the corresponding output angles are listed in Table 4. The process of selecting precision positions can be automated with the use of a spreadsheet program, which in many cases can be directly linked with the parametric design software.

As shown in Fig. 9, the input and output angles are measured positive in a counterclockwise sense relative to reference lines that pass through the center points O_A and O_B. This differs from the traditional approach to function generation in which the input and output angles are measured relative to a fixed line that is typically horizontal. Allowing the orientation of the reference lines to vary introduces two additional parameters, denoted as ψ_2 and ψ_4, so there are a total of five design parameters: r_2, r_3, r_4, ψ_2, and ψ_4. With only four precision positions specified in the example, the problem is underconstrained.

The first step is to draw the ground link of arbitrary length between the center points O_A and O_B. The length is arbitrary because the entire four-bar linkage can be scaled. In this example, the length is chosen as 100. The second step is to draw the two reference lines, one through O_A and the other through O_B. The precision positions are laid out by drawing lines, also through O_A and O_B, at the appropriate angles $\theta_{2,i}$ and $\theta_{4,i}$ $(i=1,2,3, \text{ and } 4)$ relative to the two reference lines. This construction is shown in Fig. 9. The Dimension Lock constraint is applied to these angles to fix their values. Finally, four different four-bar linkages are drawn, one associated with the pair of angles of each precision position. In Fig. 9, the endpoints of links 2 coincide with the endpoints of the precision position lines on the input side, and the endpoints of links 4 are located in the middle of the precision position lines on the output side. The two different locations are shown here to emphasize that this aspect of the construction is inconsequential. For links 2, application of the Strong Connect constraint requires the endpoints to be coincident, whereas for links 4, application of the Weak Connect constraint simply requires the endpoints of the links to lie on the precision position lines. Once the Equality constraint is applied to the lengths of the corresponding links in each four-bar linkage, the effects of the two different constructions are identical because no lengths are ever explicitly dimensioned.

One degree of freedom remains after the corresponding link lengths have been constrained to be equal. The designer can rotate the reference line for the input link to change the angle ψ_2, and the linkage will automatically update to satisfy the constraints. Alter-

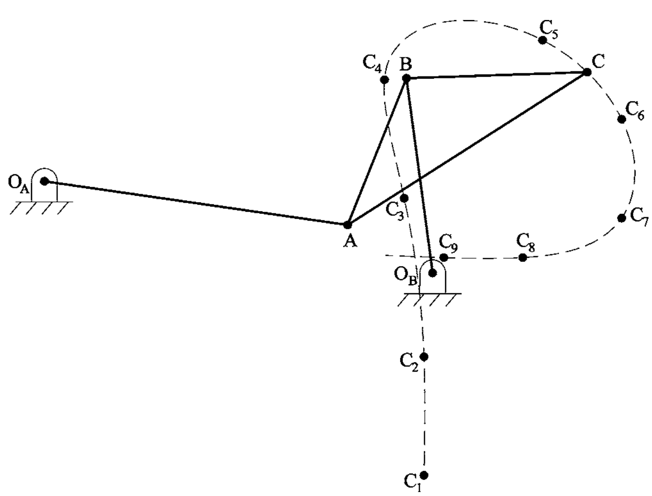

Fig. 8 The synthesized four-bar linkage and a portion of the coupler curve

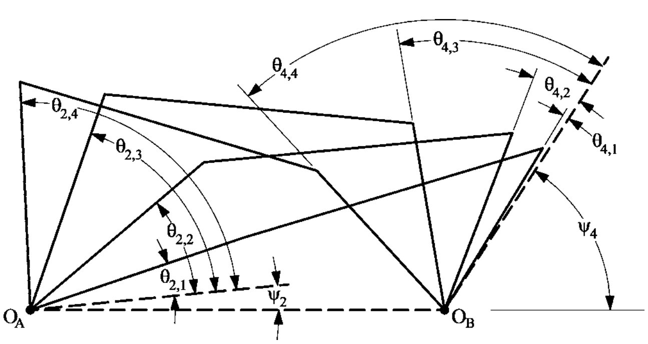

Fig. 9 Four finitely separated positions for function generation

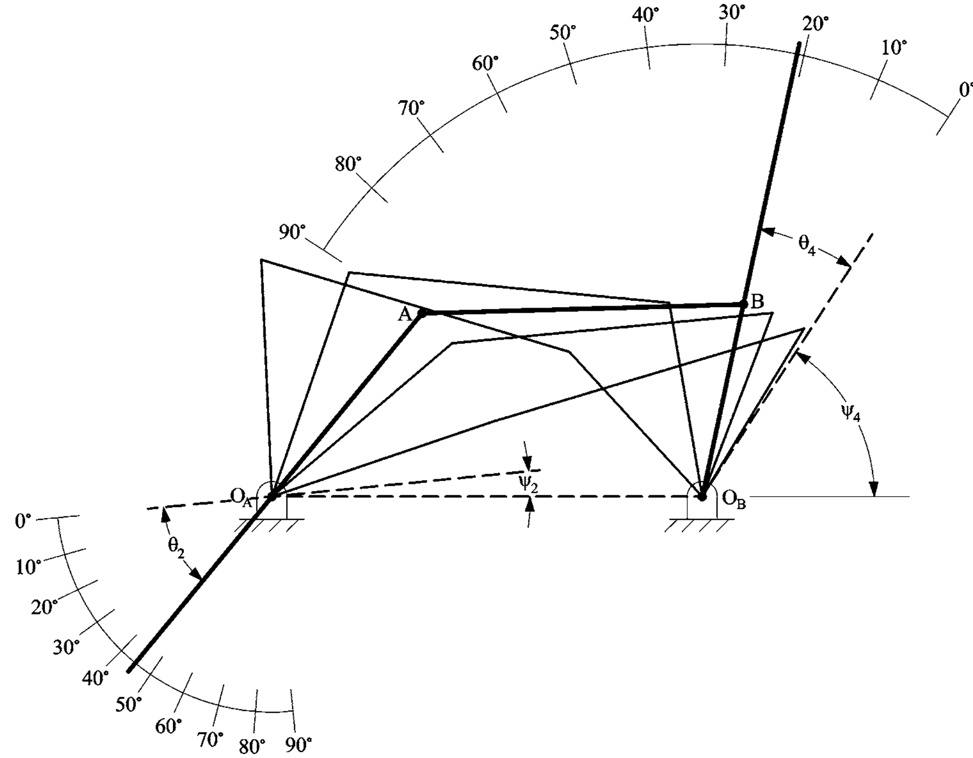

Fig. 10 The synthesized four-bar linkage in an intermediate position

natively, the designer can change the length of any individual link while the other variables dynamically adjust. This flexibility allows the designer to quickly observe the effects of changes in individual design parameters and optimize the design according to some visual criteria such as transmission angles. Once a satisfactory linkage is obtained, the design can be finalized by applying the Dimension Lock constraint to any of the link lengths, to ψ_2, or to ψ_4. As in the previous examples, the final step is to validate the linkage by moving it through the design range of motion to ensure that there are no order or branch defects.

For the example presented in this section, the Dimension Lock constraint was applied to make the length of the input link $O_AA=55$. The remaining dimensions of the synthesized four-bar linkage, shown in an intermediate position in Fig. 10, are $O_AO_B=100$, $AB=74.71$, and $O_BB=45.62$. The structural error of this linkage (i.e., the theoretical difference between the function produced by the synthesized linkage and the function originally prescribed) is less than 2 deg over the specified range of motion of the input link.

6 Complex Mechanisms and Atypical Constraints

In the motion and path generation examples presented in Secs. 3 and 4, the GCP technique exploits the fact that two points in the coupler of a four-bar linkage travel on circular arcs. This fact is also exploited by traditional graphical synthesis techniques for four-bar linkages. For many linkages with more than four links, such as the Stephenson six-bar linkage, though, the designer has inadequate a priori knowledge about the motion of individual coupler points. GCP is well-suited to problems of this nature because it does not rely upon any such a priori knowledge. To synthesize a Stephenson six-bar linkage, for example, a designer only needs to define a series of task requirements such as precision positions, precision points, or a combination of both, and draw a six-bar linkage satisfying each. Then, the dimensions of the numerous six-bar linkages are constrained to be equal, and the center points are constrained to be coincident. The synthesized six-bar linkage can be moved through its full range of motion to validate the design. As in the planar four-bar linkage synthesis examples, it may be necessary, prior to application of the final constraints, to move the six-bar linkages to positions that enable the numerical solver to successfully converge. Developing a viable design may also require iteration since circuit and branch defects are common in six-bar synthesis problems. One strategy might be to divide a complex problem into two sub-problems, each including only a subset of the task requirements. Two separate mechanisms can be more easily generated to satisfy the reduced sets of constraints, and then these two mechanisms can be constrained to be identical in the final step, such that the final mechanism satisfies the full set of task requirements.

Another useful feature of GCP is the ability to solve kinematic synthesis problems with nonstandard constraints. An example is the design of three-point hitches on the back of most agricultural tractors. In this problem, the center points and the range of motion of the crank may be pre-defined by the tractor itself. The length of the rocker is variable, and the designer can select the length of the crank to produce the desired path or orientations for the implement. In this problem, the desired performance is not defined by traditional motion generation task requirements. Rather, the design goals are typically to locate the instant center (also called the pitch point for three-point hitches) in a desirable location, to provide large ground clearance for the implement when it is not in use, and to allow for some elevation of the implement without significant change in its orientation. The designer using the GCP technique can employ the Dimension Lock constraint to specify the pre-defined dimensions and then impose nonstandard constraints to meet the appropriate task requirement. The ease with which a designer can mix constraints to perform a hybrid synthesis in highly constrained problems is a significant advantage of GCP over other techniques that require more rigid problem definitions.

It is worth noting that the kinematic synthesis of extremely complex linkages by the GCP technique may currently not be practical for most desktop computers due to computational limi-

tations. However, as memory and processing power become increasingly accessible, it will be possible to directly extend the GCP technique to complex linkages.

7 Summary

The primary contribution of this paper is the first published description of an approach to kinematic synthesis using the constraint-based sketching mode available in most modern parametric computer-aided design software. This technique, referred to as geometric constraint programming (and denoted as GCP), allows the designer to synthesize planar mechanisms in a new, intuitive manner by creating a "program" in the form of a kinematic diagram. Instead of explicitly formulating the underlying nonlinear equations, the designer imposes geometric constraints using lines, circles and other fundamental geometric constructions through the graphical interface of a CAD package. These constraints provide a mapping from the input task requirements of the problem to the synthesized mechanism. The results provided by geometric constraint programming exhibit the accuracy and repeatability achieved with analytical synthesis techniques, while simultaneously providing the geometric insight developed through graphical synthesis techniques. The advantages of the technique are that it is broadly applicable to a wide range of synthesis problem, it is sufficiently user friendly so as not to require an advanced background in kinematics, and it is highly accessible to engineers in both academe and industry. In addition, this technique can be integrated in a straightforward manner into a closed-loop design process. The conclusion is that the geometric constraint programming technique provides a very effective method to design mechanisms. For the purpose of illustration, the technique is applied in this paper to the kinematic synthesis of planar four-bar linkages for motion generation through five finitely separated positions, path generation through nine finitely separated points, and function generation through four finitely separated points. Application of the GCP technique to more general synthesis problems is also addressed.

Finally, future enhancements of parametric design software will only broaden and improve the application of GCP to the kinematic synthesis of mechanisms. As processing power continues to improve, the iterative equation solvers in the software will be able to converge more quickly to solutions to large systems of nonlinear equations. This will enable increasingly complex mechanisms to be synthesized using GCP. In addition, none of the CAD packages with which the authors are familiar currently offer an automated means of drawing a locus of points that is not a line or a conventional arc. This creates some difficulty in plotting center point curves, circle point curves, and coupler curves. The figures presented in this paper were drawn in SOLID EDGE Version 15 [24] using the techniques that are described in the text. Note that in Figs. 3–5 and 8, the center point curves and the coupler curves were traced, discrete points were fixed, and then a spline was drawn through these points. This is a time consuming process. While not fundamentally necessary for kinematic synthesis, the ability to quickly and easily plot such curves would be a significant advantage for visualization and would, therefore, further benefit the GCP technique.

References

[1] Beyer, R., 1963, *The Kinematic Synthesis of Mechanisms*, Trans. by H. Kuenzel, McGraw-Hill, New York.
[2] Hirschhorn, J., 1962, *Kinematics and Dynamics of Plane Mechanisms*, McGraw-Hill, New York.
[3] Uicker, J. J., Jr., Pennock, G. R., and Shigley, J. E., 2003, *Theory of Machines and Mechanisms*, 3rd ed., Oxford University Press, New York.
[4] Dijksman, E. A., 1976, *Motion Geometry of Mechanisms*, Cambridge University Press, Cambridge, England.
[5] Erdman, A. G., Sandor, G. N., and Kota, S., 2001, *Mechanism Design*, 4th ed., Prentice-Hall, Upper Saddle River, New Jersey, Vol. 1.
[6] Hain, K., 1967, *Applied Kinematics*, 2nd ed., McGraw-Hill, New York.
[7] Hartenberg, R. S., and Denavit, J., 1964, *Kinematic Synthesis of Linkages*, McGraw-Hill, New York.
[8] Hall, A. S., Jr., 1986, *Kinematics and Linkage Design*, Waveland Press, Inc., Prospect Heights, Illinois. (Originally published by Prentice-Hall, Inc., 1961).
[9] Burmester, L., 1888, *Lehrbuch der Kinematik*, Felix, A., ed., Verlag, Leipzig, Germany.
[10] Kaufman, R. E., 1971, "KINSYN: An Interactive Kinematic Design System," *Proceedings of the Third World Congress on the Theory of Machines and Mechanisms*, Dubrovnik, Yugoslavia.
[11] Reed, W. S., 1972, "MECHSYN - An Interactive Mechanism Synthesis Package," Computer Graphics in Computer-Aided Design, School of Mechanical Engineering, Purdue University, West Lafayette, Indiana.
[12] Erdman, A. G., and Gustafson, J. E., 1977, "LINCAGES: Linkage Interactive Computer Analysis and Graphically Enhanced Synthesis Package," *ASME Paper No. 77-DETC-5*.
[13] Chuang, J. C., Strong, R. T., and Waldron, K. J., 1981, "Implementation of Solution Rectification Techniques in an Interactive Linkage Synthesis Program," ASME J. Mech. Des., **103**(3), pp. 657–664.
[14] Barker, C. R., and Tso, P. L., 1988, "Three Position Path Generation of Planar Four-Bar Mechanisms Using SYNTRA," *ASME Trends and Developments in Mechanisms, Machines and Robotics, Proceedings of the 20th Biennial ASME Mechanisms Conference*, DE - Vol. 15-1, pp. 279–286, Kissimmee, Florida, September 25–28.
[15] Soni, A. H., Dado, M. H. F., and Weng, Y., 1988, "An Automated Procedure for Intelligent Mechanism Selection and Dimensional Synthesis," ASME J. Mech., Transm., Autom. Des., **110**, pp. 130–137.
[16] http://www.symech.com
[17] http://www.softintegration.com/webservices/mechanism/
[18] http://www.artas.nl
[19] http://www.heron-technologies.com/watt/
[20] Erdman, A. G., 1995, "Computer-Aided Mechanism Design: Now and the Future," ASME 50th Anniversary of Design Engineering Division, Special Combined Issue of Journal of Mechanical Design and Journal of Vibration and Acoustics, 117(B), pp. 93–100.
[21] Waldron, K. J., and Kinzel, G. L., 2004, *Kinematics, Dynamics, and Design of Machinery*, 2nd ed., Wiley, New York.
[22] Norton, R. L., 2004, *Design of Machinery*, McGraw-Hill, New York.
[23] Sridhar, N., Agrawal, R., and Kinzel, G. L., 1996, "Algorithms for the Structural Diagnosis and Decomposition of Sparse, Underconstrained Design Systems," Comput.-Aided Des., **28**(4), pp. 237–249.
[24] http://www.solidedge.com
[25] http://www.ugs.com/
[26] http://www.ptc.com
[27] http://www.solidworks.com
[28] http://www.autodesk.com