Order Number 9322439

Kinematic synthesis of adjustable four-bar mechanisms for multi-phase motion generation

Wang, Shao Jie, Ph.D.

New Jersey Institute of Technology, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

KINEMATIC SYNTHESIS
OF ADJUSTABLE FOUR-BAR MECHANISMS
FOR MULTI-PHASE MOTION GENERATION

by
Shao Jie Wang

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Mechanical Engineering

January, 1993

APPROVAL PAGE

Kinematic Synthesis of Adjustable Four-Bar Mechanisms for Multi-Phase Motion Generation

Shao Jie Wang

Dr. Raj S. Sodhi, Thesis Advisor
Director of Manufacturing Engineering Program and Associate Professor of Mechanical Engineering, NJIT

Dr. Bernard Koplik, Committee Member
Chairperson of Mechanical and Industrial Engineering and Professor of Mechanical Engineering, NJIT

Dr. Harry Herman, Committee Member
Professor of Mechanical Engineering, NJIT

Dr. Charles E. Wilson, Committee Member
Professor of Mechanical Engineering, NJIT

Dr. Zhiming Ji, Committee Member
Assistant Professor of Mechanical Engineering, NJIT

Dr. Sanchoy K. Das, Committee Member
Assistant Professor of Industrial and Management Engineering, NJIT

ABSTRACT

Kinematic Synthesis of Adjustable Four-Bar Mechanisms for Multi-Phase Motion Generation

by Shao Jie Wang

A four-bar linkage can satisfy up to five prescribed positions for the motion generation problem. The adjustable four-bar linkage, on the other hand, can satisfy more than five given positions by making some of the parameters adjustable.

Limited work had been done in the area of motion generation problems of kinematic synthesis of adjustable four-bar linkages until Wilhelm introduced the concept of multiple adjustments.

This study considers for the first time, the adjustment of a moving pivot, and the problems of three phases of motion. Various combinations of the number of prescribed positions for the motion generation problems are solved here until the prescribed positions reach the maximum permissible number. These solutions are developed for two and three phase adjustable moving pivot problems, two phase adjustable moving pivot and crank length problems, three phase adjustable crank length problems, and three phase adjustable fixed pivot problems. Equations are also developed for the most complicated cases, which are two phase adjustable moving pivot problems with three positions in each of the two phases, and three phase adjustable crank length problem with two positions in each of the three-phases.

Six synthesis example problems are presented which represent various topics covered in this study. Several Turbo Pascal programs are developed for solving the synthesis problems. Many user-defined AutoLISP functions and commands are specially designed for this work.

BIOGRAPHICAL SKETCH

Author: Shao Jie Wang

Degree: Doctor of Philosophy in Mechanical Engineering

Date: January, 1993

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 1993
- Master of Science in Computer and Information Science,
  New Jersey Institute of Technology, Newark, NJ, 1987
- Bachelor of Engineering in Mechanical Engineering,
  Shanghai Jiao–Tong University, Shanghai, China, 1961

Major: Mechanical Engineering

Presentations and Publications:

Wang, Shao Jie et al. "Fundamentals of Machine Design", People's Education Publishing House, Beijing, China, 1979.

Wilson C. E., T. K. Vaidyanathan, W. R. Cinotti, S. M. Cohen, and S. J. Wang. "Hearing Damage Risk and Communication Interference in Dental Practice", *Journal of Dental Research*, February 1990.

This thesis is dedicated to
Dr. Charles E. Wilson, Dr. James A.M. McHugh,
Mr. James Whitescarver, Dr. Harry Herman,
Dr. Bernard Koplik and Dr. Raj S. Sodhi

ACKNOWLEDGEMENT

The author wishes to express his sincere gratitude to his advisor, Dr. Raj S. Sodhi, for his guidance, friendship, and moral support throughout this research.

Special thanks to Doctors Bernard Koplik, Harry Herman, Charles E. Wilson, Zhiming. Ji, and Sanchoy K. Das for serving as members of the committee, and to Dr. Andrew J. Wilhelm for his help. The author is also grateful to the Mechanical Engineering Department of New Jersey Institute of Technology for the funding of this research.

TABLE OF CONTENTS

Chapter **Page**

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

Figure	Page

Figure		Page

Figure **Page**

Chapter 1
Introduction

1.1 Tasks of Kinematic Synthesis

In kinematic analysis, a given mechanism is investigated for kinematic characteristics, such as degrees of freedom, velocity, acceleration, etc. for a given position and an input motion. Kinematic synthesis, on the other hand, is to choose the type and determine the dimensions of a mechanism in order to accomplish a prescribed task. In other words, kinematic synthesis is to design a new mechanism.

There are three kinds of problems in kinematic synthesis: function generation, path generation, and motion generation problems.

In function generation, the motions of driving and driven links need to be correlated by a function $y = f(x)$. An example ideal function $y = f(x)$ is shown in Figure 1.1 for the given range $a \leq x \leq b$. However, a four-bar linkage can only satisfy a given function at a limited number of prescribed precision points. As shown in Figure 1.1, the curve representing the actual motion intercepts the curve of ideal motion at a limited points P_1, P_2, P_3, and P_4. The actual motion approximates the ideal motion at other points.

A path generation task needs a point on a coupler to trace a prescribed path with respect to the fixed reference. The prescribed precision points on the path may need to be correlated with either driving link positions or time. In this case, the task is called path generation with prescribed timing.

Motion generation or rigid-body guidance requires that a moving body be guided through a series of prescribed positions. The body to be guided usually is a part of a coupler.

Since a linkage has only a finite number of significant dimensions, the designer may only prescribe a finite number of precision points. That is, for all three kinds of synthesis tasks, only a finite number of precision points could be satisfied.

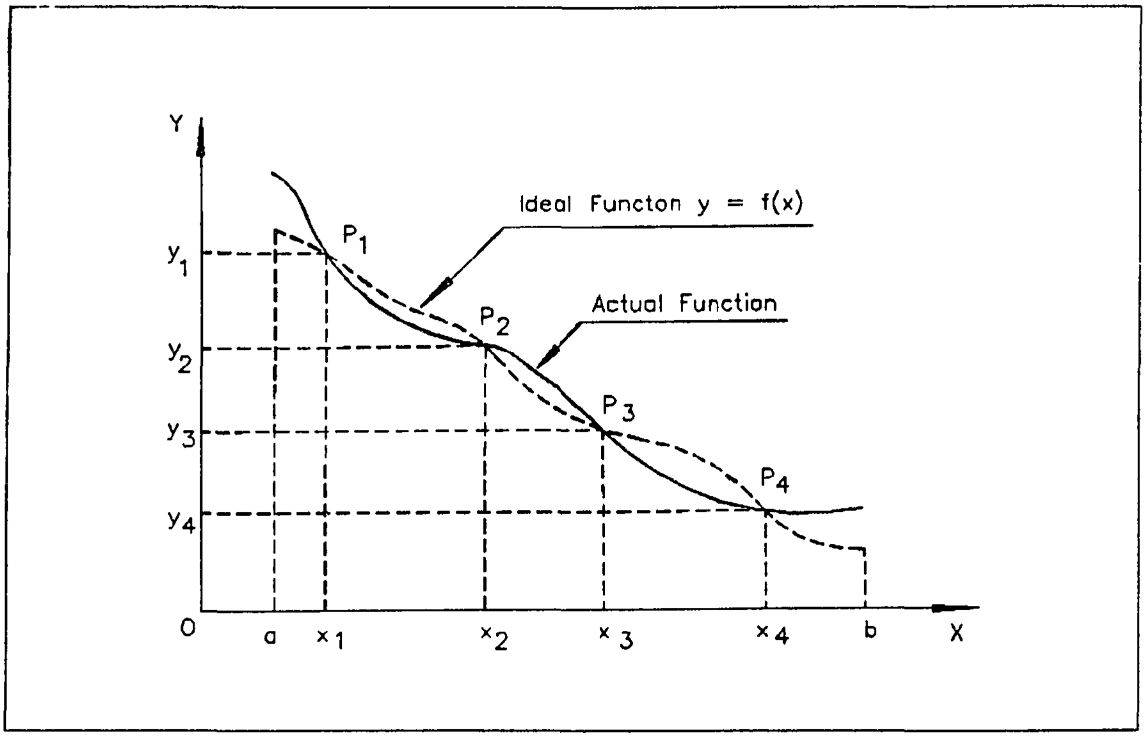

Figure 1.1 Function generation problem

1.2 Review of Adjustable Linkages

Adjustable linkages are important in varying the behavior of a linkage. Achieving varied outputs from one set of "hardware" with simple adjustments is a definite advantage in engineering design.

In the area of path generation, Tao [1, 2, 3] has published a considerable amount of work. Tao and Amos [1] developed a technique which provides a flexible means of synthesizing an adjustable four–bar linkage to satisfy a specified change in the direction of a straight line segment traced by a coupler point. They presented a simple and straightforward graphics technique to change the direction of the straight line motion through a specified angle by simply adjusting a fixed pivot. Tao and Krishnamoorthy [2] investigated the adjustable four–bar linkage with a double point, and developed a graphical synthesis method for tracing "figure–eight" with approximate straight line segments at double point by adjusting the fixed pivot.

Tao and Yan [3] developed a synthesis technique of adjustable linkages to generate symmetrical coupler curves with tangential circular arcs and concentric circular arcs by adjusting a fixed pivot or a circle pivot.

Beaudrot [4] presented his synthesis technique to meet the following design requirements: (1) Design a four–bar linkage such that two points on the coupler plane each trace two straight–line motions, but at different specified angular displacements, for an adjustment of a fixed pivot. (2) Design a four–bar linkage such that one point on the coupler plane will trace three straight–line motions for three specified adjustments of a fixed pivot.

Bonnell and Cofer [5] extended complex–number method of plane kinematic synthesis developed by Sandor for adjustable linkages by adjusting either of the fixed pivot on a circular arc or adjusting the crank length.

McGovern and Sandor [8, 9] extended the work of Bonnell and Cofer [5] to geared linkages and high order synthesis. They developed a technique to obtain analytical and closed form solutions for arbitrary adjustable paths of a coupler point by means of adjusting a fixed pivot. The complex–number method could also be applied to function generation problems. The linkage considered are a four–bar, a geared five–bar, and a geared six–bar mechanism.

In the case of adjustable linkages for motion generation, the published work is limited. Ahmad and Waldron [10, 11] developed a technique for a four–bar linkage with adjustable driven fixed pivot. They solved two phase problems with a maximum total number of five positions. Wilhelm [12] developed synthesis techniques for two phase motion generation problems of adjustable four–bar linkages. He solved various combination of positions for adjustable fixed pivot problems, adjustable crank length problems, and multiple adjustment of the two. Also, the total number of positions of the synthesis tasks reached their maximum possible value.

1.3 Objective of This Research

As mentioned in the last section, not much work had been done in the area of motion generation problems of kinematic synthesis of adjustable four–bar linkages. Ahmad and Wilhelm made significant contributions, but their research did not cover the adjustment of moving pivot and the problems of more than two phases.

This research develops solutions for the first time to the following problems: two and three phase adjustable moving pivot, two phase adjustable moving pivot and crank length, three phase adjustable crank length, and various three phase motion generation synthesis problems.

The synthesis problems of various combinations of the prescribed position numbers for all different kind problems mentioned above are going to be solved in this study until the total number of positions reaches the maximum permissible number.

A motion generation problem is to guide a rigid body, which is usually part of the coupler to take a series of prescribed positions in a prescribed order by means of a linkage. The maximum number of prescribed positions for synthesizing a four–bar linkage is five. This maximum number is increased when an adjustment is made to a four–bar linkage. The bigger the maximum number the more prescribed positions could be satisfied by an adjustable linkage. One of the objective is to determine the maximum allowable number of prescribed positions for each of the cases.

The final goal of this research is to code this work into programs, so that the user can solve their synthesis problems on an IBM compatible personal computer by means of Turbo Pascal, AutoCAD, and AutoLISP. Many user–defined AutoLISP functions and commands need to be developed for this research.

Chapter 2
Motion Generation Problems

2.1 Basic Equations

This chapter covers the motion generation problems for a normal four–bar linkage.

As shown in Figure 2.1, the position of a moving body (coupler) in general plane motion could be specified by a point (A) and an angle (θ) of a line (AX') passing through it. In the figure, the moving reference X'AY' is fixed to the coupler and moves along with it wherever it goes. A crank with a center pivot S (P,Q) and a circle pivot C (X,Y) is also shown in the figure. The following equations are valid for all positions of the crank if the fixed pivot, the crank length are not made adjustable:

$$(X_i - P)^2 + (Y_i - Q)^2 = R^2 \qquad i = 1, \ldots, n \quad (2.1)$$

where X_i and Y_i are coordinates of circle pivot C with respect to the fixed frame, P and Q are the coordinates of the fixed pivot with respect to the fixed frame, R is the crank length, and n is the number of precision positions.

Also,

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1, \ldots, n \quad (2.2)$$

where p and q are coordinates of the circle pivot C with respect to the moving reference X'AY', a_i and b_i are coordinates of the origin of the moving coordinate system with respect to the fixed reference.

As mentioned before, a motion generation problem is to guide a rigid body, which is usually part of the coupler to take a series of prescribed positions in a specified sequence by means of a linkage. With equations (2.1) and (2.2), the prescribed positions could thus be represented by a_i, b_i and θ_i, where i = 1, ... ,n. n is the number of prescribed positions. Substitution of equations (2.2) into (2.1) to eliminate X_i and Y_i leaves us with five unknowns: P, Q, p, q, and R. Hence, the maximum number of prescribed positions for synthesizing a normal four-bar linkage is five.

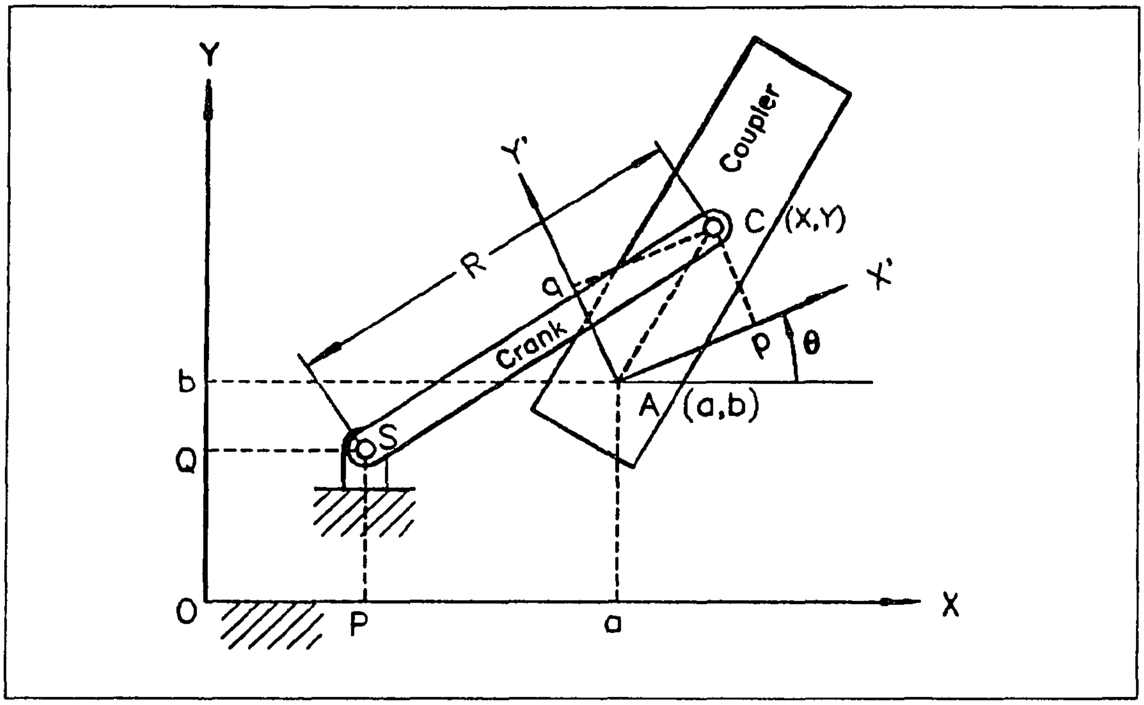

Figure 2.1 The moving and fixed references

An ideal motion of the coupler can only be approximated by several discrete precision positions. The linkage can create the motion precisely at these positions and will approximate the ideal motion at other positions. The more precision positions used, the closer to the ideal motion is the actual motion of the coupler. But the problem is more difficult to solve as the number of precision positions is increased. Fortunately, many real world problems only need several critical positions to be satisfied precisely. Tolerance is usually allowed at other positions.

2.2 Synthesis Problems of Three Finitely Separated Positions

For a problem of three prescribed precision positions, equation (2.1) becomes

$$(X_i - P)^2 + (Y_i - Q)^2 = R^2 \qquad i = 1, 2, 3 \qquad (2.3)$$

Eliminate R, we get

$$(X_2 - P)^2 + (Y_2 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \qquad (2.4)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \qquad (2.5)$$

Substitute equation (2.2) into (2.4) and (2.5), we have

$$L_2P + M_2Q + N_2 = 0 \qquad (2.6)$$

$$L_3P + M_3Q + N_3 = 0 \qquad (2.7)$$

where L, M, and N are functions of a, b, θ, p, and θ.

We have two free choices of parameters. P and Q can be solved by means of equations (2.6) through (2.7) after choosing p and q. R can be solved by equation (2.3).

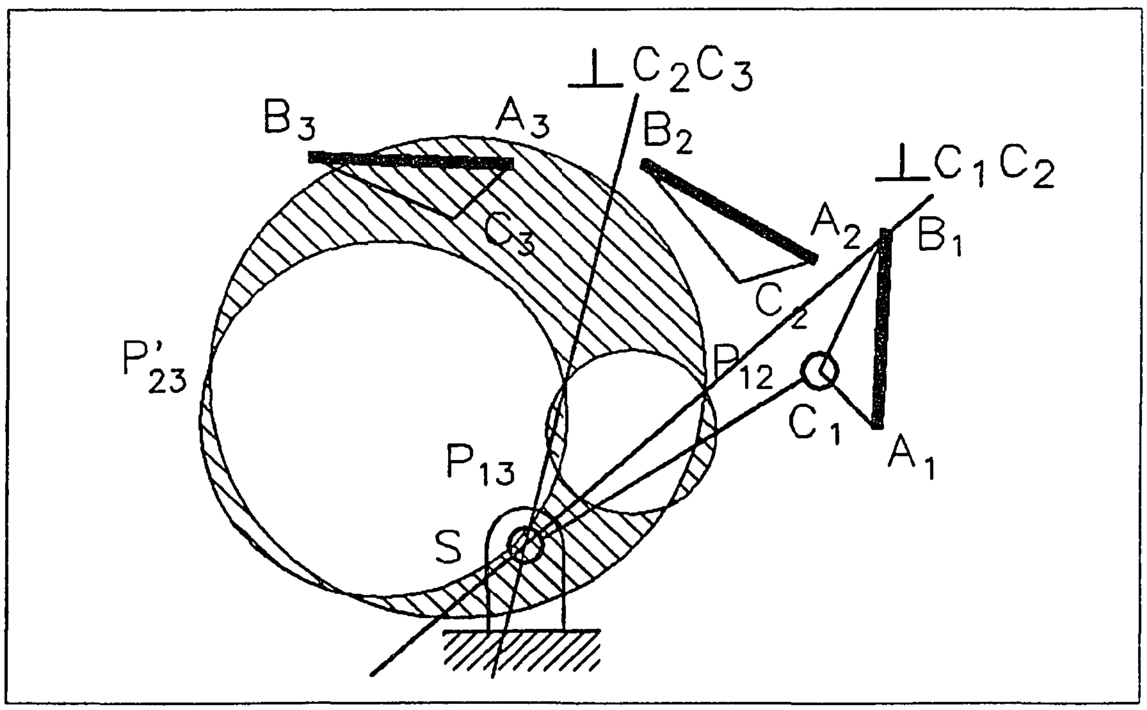

Figure 2.2 Three prescribed positions and the Waldron Image Pole Circles

Graphically, equations (2.6) and (2.7) represents right bisectors. The center point S would be at the intersection of the right bisectors of the segments C_1C_2 and C_2C_3 (Figure 2.2). Similarly, choose another circle point D_1 and plot bisectors to intersect another center point T as shown in Figure 2.4. The resulting linkage is also shown in Figure 2.4.

2.3 The Branch Defect for Problems of Three Prescribed Positions

The solution procedure in the last section does not guarantee that the resulting linkage can be moved through all three prescribed positions. It might be necessary to disconnect it and reassemble it in a different configuration to reach one of the positions. This behavior is associated with the appearance of branches in the coupler curves of the linkage. Hence, the defect is called a branch problem. The two different ways in which it is possible to connect the coupler and driven crank for a given driving crank angle are called geometric inversions. The branch defect arises whenever it is not possible to move the linkage from one geometric inversion to the other without disconnecting.

In order to avoid a branch defect for a three finitely separated position problem, a two steps graphical procedure was developed by Waldron [13, 14, 15] and Filemon [16].

For the driven circle point C_1, the alternate hatched area of the Waldron Image Pole Circles shown in Figure 2.2 gives the non–permissible area. For the driving circle point D_1, on the other hand, the hatched region of the Filemon construction shown in Figure 2.3 represents the forbidden region.

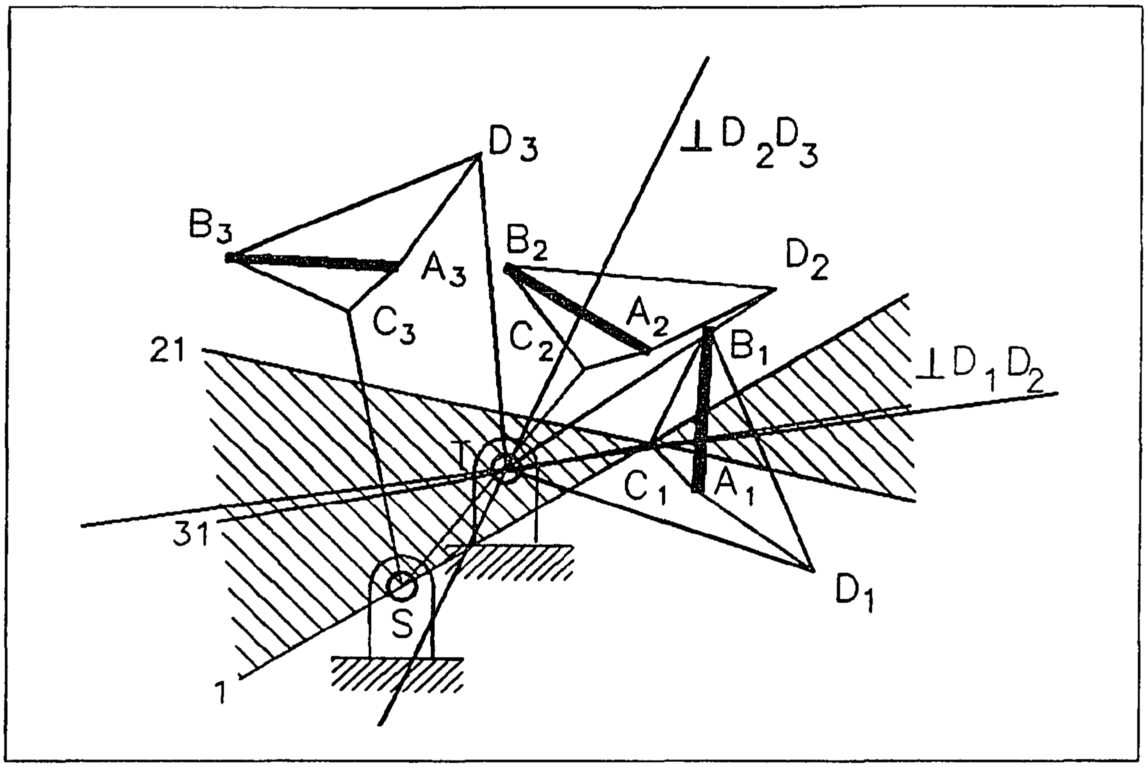

Figure 2.3 The Filemon Construction Lines

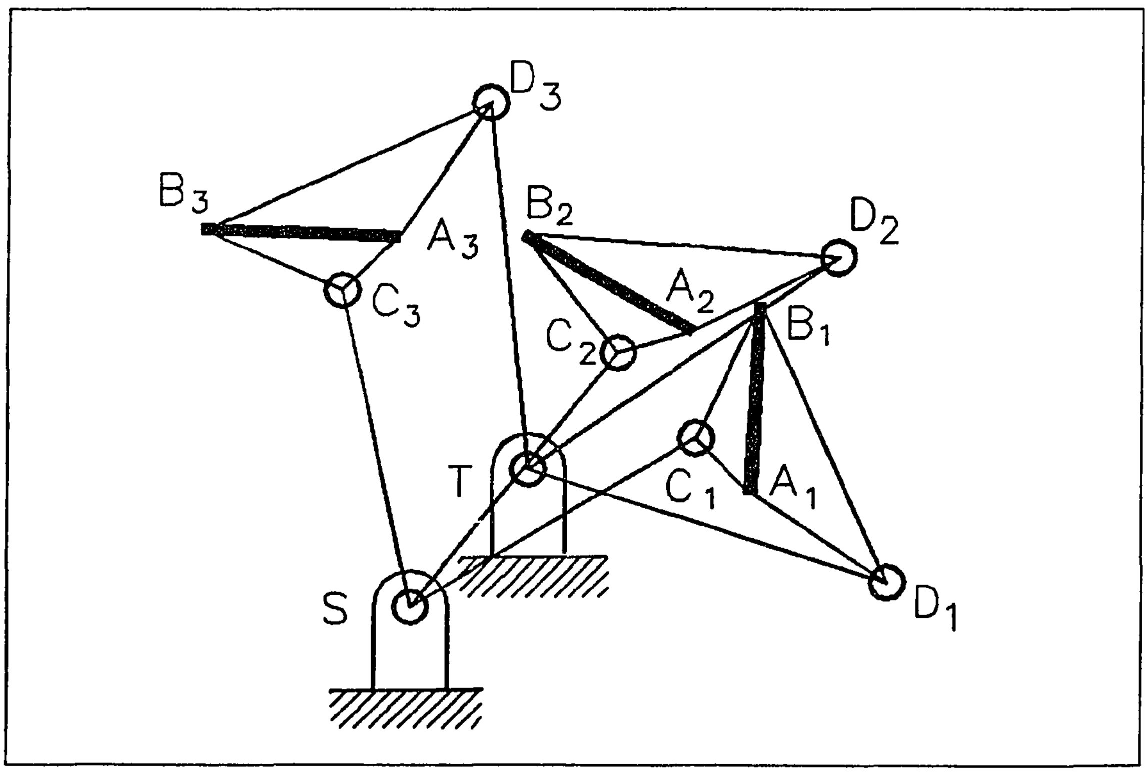

Figure 2.4 The resulting linkage

The restriction of the technique is that the resulting linkage must be a crank–rocker or a drag–link. A solution for the sample problem is shown in Figure 2.4, which is a drag–link.

2.4 Synthesis Problems of Four Finitely Separated Positions

For four positions, equation (2.1) becomes

$$(X_i - P)^2 + (Y_i - Q)^2 = R^2 \qquad i = 1, 2, 3, 4 \quad (2.8)$$

Eliminate R, we get

$$(X_2 - P)^2 + (Y_2 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \quad (2.9)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \quad (2.10)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \quad (2.11)$$

substitute equation (2.2) into (2.9) through (2.11), collect term in P and Q, we have

$$L_2P + M_2Q + N_2 = 0 \quad (2.12)$$

$$L_3P + M_3Q + N_3 = 0 \quad (2.13)$$

$$L_4P + M_4Q + N_4 = 0 \quad (2.14)$$

where L, M, and N are functions of a, b, θ, p, and θ.

For nontrivial solutions, the determinant of the coefficient matrix must be equal to zero.

$$\begin{vmatrix} L_2 & M_2 & N_2 \\ L_3 & M_3 & N_3 \\ L_4 & M_4 & N_4 \end{vmatrix} = 0 \quad (2.15)$$

This leads to the following circle point curve equation.

$$(Ap + Bq)(p_2 + q_2) + Cpq + Dp_2 + Eq_2 + Fp + Gq + H = 0 \quad (2.16)$$

where A, B, ... ,H are functions of a, b, and θ. Figure 2.5 shows the circle point curve for a sample problem. Every point on the curve should satisfy equations (2.8) and (2.2). That is, a circle point on the curve and its positions for prescribed positions 2, 3, and 4 should all lie on the same circle. The center of the circle is the center point.

Four prescribed positions of an example problem are shown in Figure 2.5. The circle point curve is plotted by running Turbo Pascal program CIRC_PT.PAS [12]. Pick two circle points C_1 and D_1 for the driving and driven cranks respectively. Find two center points S and T at the center of the circles passing through the circle points. The resulting linkage is shown in Figure 2.5. Neither a branch problem nor an order defect is found for the resulting linkage by inspection. It is a drag–link mechanism by its dimensions.

Similarly, collect term in p and q after substituting equation (2.2) into (2.9) through (2.11), we get center point equation and the center point curve.

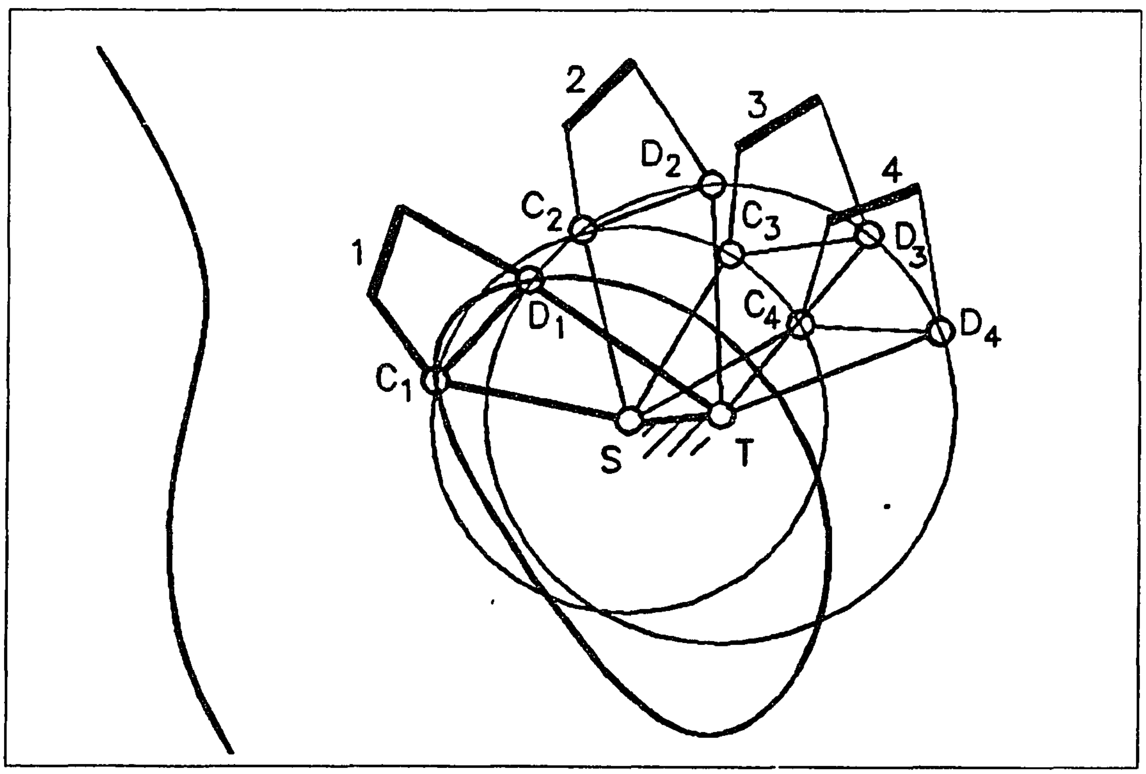

Figure 2.5 Synthesis problem of four prescribed positions

We are using only one free choice of parameter while picking a point on the curve. It is usually good to have one free choice, since the designer will have the flexibility to choose the parameters of the linkage in order to avoid the possible defects, and optimize the design to meet the overall requirement from the engineering point of view.

The branch defect is also possible in four prescribed position problems. Besides, an order problem may also occur for the four given position problems. In order to avoid the defects, some special points T_{ij}, U_{ij}, Q_{ij}, and Image poles must be found on the circle point curve [13, 14, 15, 16, 17].

Chapter 3

Two Phase Adjustable Moving Pivot Problems

3.1 Introduction

In the previous two chapters the background for the synthesis of a four–bar linkage and the motion generation problems has been described. The technique for synthesizing an adjustable four–bar linkage is going to be discussed in this chapter.

Chapter 3 deals with the adjustable moving pivot problem. The given data for a motion generation problem are represented by a consistent method in this research. That is, position i of the coupler is represented by a_i, b_i, and θ_i, where a_i and b_i are X and Y coordinates of a point A on the coupler, and θ_i is the directional angle of a line starting at A.

The synthesis of one crank of an adjustable four–bar linkage will be covered in the next four chapters, which means one side of the complete adjustable linkage only. The technique for adding another crank to complete a linkage design will be shown in chapter 8.

Eight adjustable moving pivot problems listed in Table 3.1 are going to be solved in this chapter. For a two phase problem, at least two positions are included in one phase. The last two problems in the table deal with seven positions, which is the maximum possible value for the problem. The number of shared positions is zero or one.

The problem with shared position 12–23 is considered the same as the problem 12–13. Similarly, the problem 123–345 is the same as the problem 123–145.

Table 3.1 Adjustable moving pivot problems

positions		number of	number of	number of
ph.1	ph.2	shared pos.	unknowns	free choices
1,2	3,4	0	4	3
1,2	2,3	1	4	3
1,2,3	4,5	0	5	2
1,2,3	3,4	1	5	2
1,2,3	4,5,6	0	6	1
1,2,3	3,4,5	1	6	1
1,2,3,4	5,6,7	0	7	0
1,2,3,4	4,5,6	1	7	0

3.2 Positions 12–34

For the case of two positions in each of the two phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.1}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.2}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.3}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{3.4}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \qquad i = 1,2 \tag{3.5}$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2 \tag{3.6}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \qquad i = 3,4 \tag{3.7}$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 3,4 \tag{3.8}$$

Seven parameters, P, Q, p_1, q_1, p_2, q_2, and R, are involved in four equations. Thus, this problem can be solved with three free choices of parameters, and has infinite solutions. Either an algebraic method or a graphic method can solve this problem.

The graphic method used with computer and software packages is straightforward, simple, fast, and with high precision, because it is an algebraic method internally.

Figure 3.1 represents a good solution of this problem. Choose p_1 and q_1 to locate C_1. Locate C_2 by plotting a similar triangle. Draw a right bisector for line segment C_1C_2. Choose crank length R. Draw a circle with center C_1 and radius R; this circle intersects the bisector at center point S. Invert triangle A_4B_4S into position 3 to get point S_4. Draw a circle with center S and radius R; this circle intersects the right bisector for line segment SS_4 at point D_3, which is the circle point for the second phase at position 3. D_4 can be found by plotting a similar triangle $A_4B_4D_4$.

Figure 3.2 represents another good solution of this problem. P, Q, and R are chosen as the three free choices at this time. Locate center point S after choosing P and Q. Invert center point S of position 2 into position 1 to get S_2. Choose R. Draw a circle with center S and radius R; this circle intersects the right bisector for line segment SS_2 at point C_1, which is the circle point of phase 1 at position 1. Locate point C_2 by plotting a similar triangle. Invert center point S of position 4 into position 3 to get point S_4. Plot a right bisector for line segment SS_4; this bisector intersects the circle at point D_3, which is the circle point of phase 2 at position 3. Finally, D_4 can be found by plotting a similar triangle.

Similarly, this problem could also be solved by choosing p_2, q_2, and R as the three free choices.

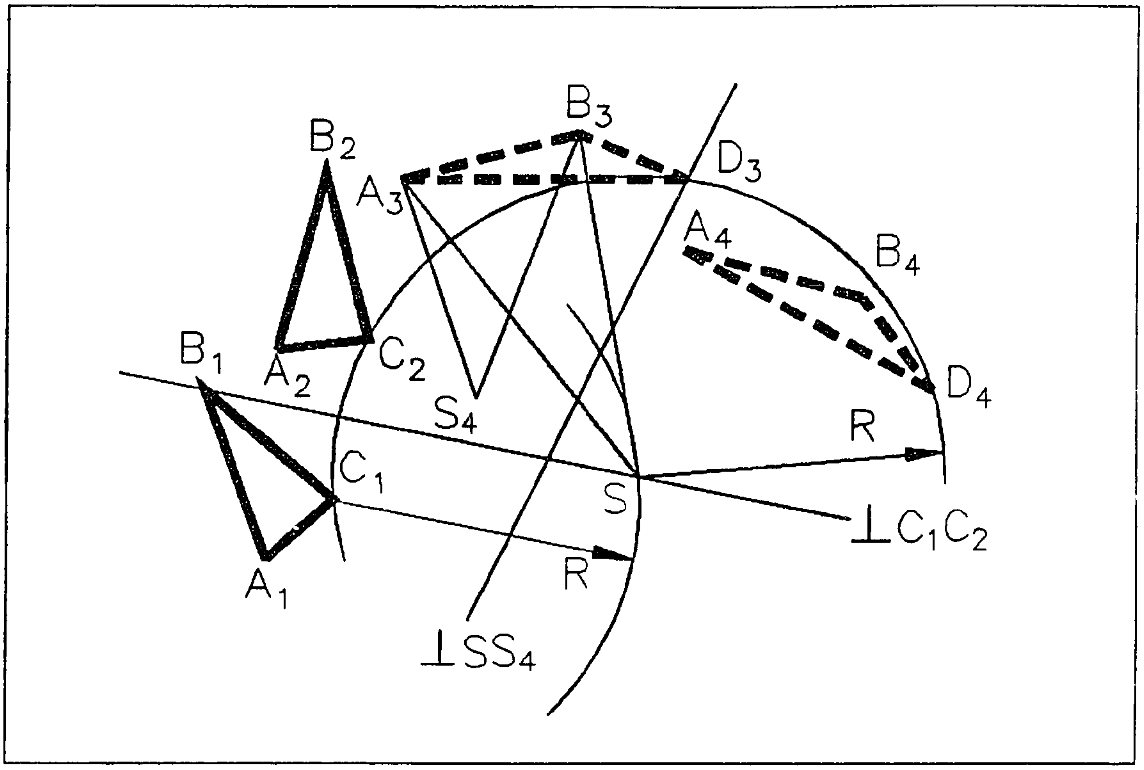

Figure 3.1 Adjustable moving pivot 12–34

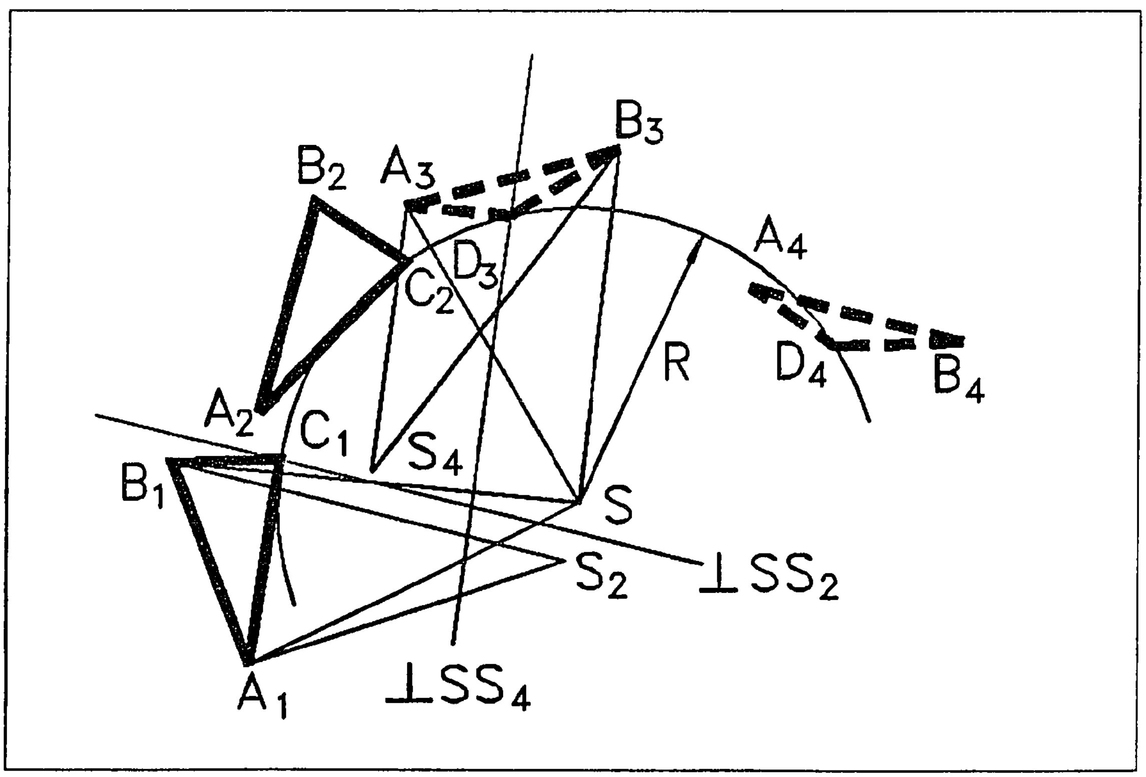

Figure 3.2 Another good solution for adjustable moving pivot 12–34

3.3 Positions 12–23

For the case of two positions in each of the two phases with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.9}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.10}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.11}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \qquad i = 1,2 \tag{3.12}$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2 \tag{3.13}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \qquad i = 2,3 \tag{3.14}$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 2,3 \tag{3.15}$$

The number of unknowns is still seven: P, Q, p_1, q_1, p_2, q_2, and R. The number of equations is still four, because at the shared position, X_2 and Y_2 are relating to p_1 and q_1 in equations (3.12) and (3.13), and to p_2 and q_2 in equations (3.14) and (3.15). Thus, this problem still can be solved with three free choices of parameters, and has infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Figure 3.3 represents a good solution of this problem. Choose p_1 and q_1 to locate C_1. Locate C_2 by plotting a similar triangle. Draw a right bisector for line segment C_1C_2. Choose crank length R. Draw a circle with center C_1 and radius R; this circle intersects the bisector at center point S. Invert triangle A_3B_3S to position 2 to get point S_3. Draw a circle with center S and radius R; this circle intersects the right bisector for the line segment SS_3 at point D_2, which is the circle point for phase 2 at position 2. D_3 can be found by plotting a similar triangle.

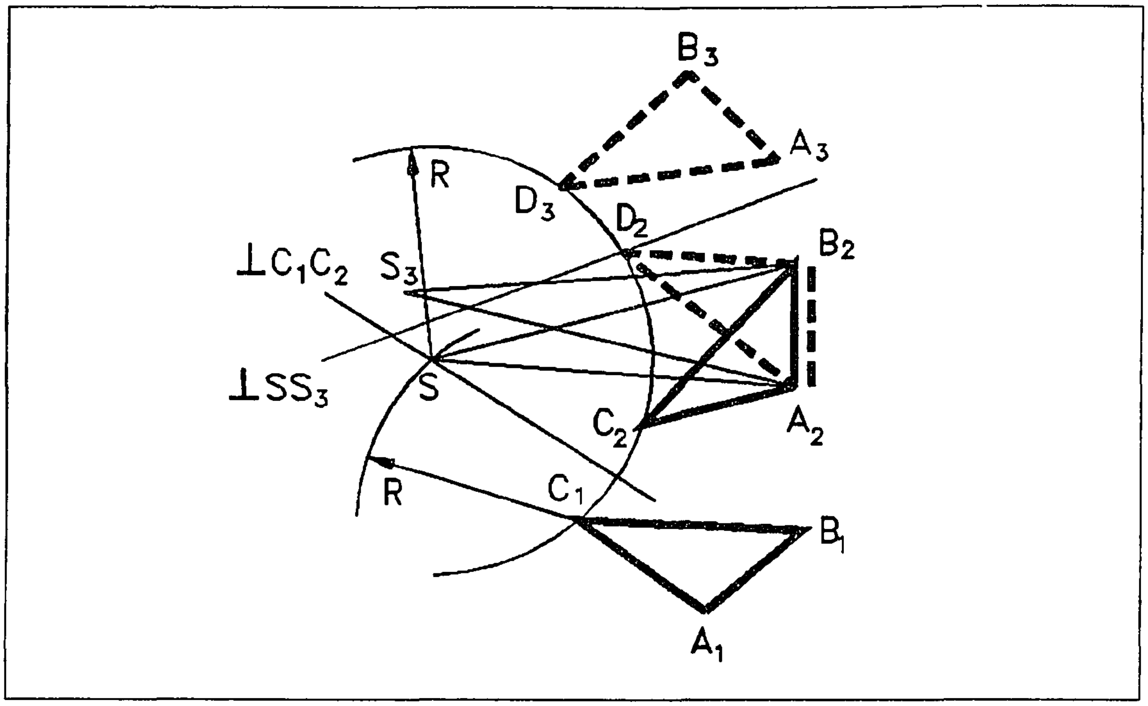

Figure 3.3 Adjustable moving pivot 12–23

Similar to the problem in the last section, the three free choices of parameters could be either P, Q, and R or p_2, q_2, and R.

3.4 Positions 123–45

For the case of three positions in one phase, two positions in the other phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.16}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.17}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.18}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{3.19}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{3.20}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \qquad i = 1,2,3 \tag{3.21}$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \qquad (3.22)$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \qquad i = 4,5 \qquad (3.23)$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5 \qquad (3.24)$$

Seven parameters, P, Q, p_1, q_1, p_2, q_2, and R, are involved in five equations. Thus, this problem can be solved with two free choices of parameters, and has infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Figure 3.4 represents a good solution of this problem. Choose p_1 and q_1 to locate C_1. Locate C_2 and C_3 by plotting similar triangles. Intersect two bisectors for line segments C_1C_2 and C_2C_3 at center point S.

Invert point S from position 5 into position 4 to get point S_5. Draw a circle with center S passing through points C_1, C_2, and C_3. Draw a right bisector for the line segment SS_5; this bisector intersects the circle at point D_4, which is the circle point for phase 2 at position 4. D_5 can be found by plotting a similar triangle $A_5B_5D_5$.

P and Q could also be chosen as the two free choices. As shown in Figure 3.5, the center point S is inverted from position 2 and 3 into position 1 to get points S_2 and S_3. Intersect bisectors for line segments SS_2 and S_2S_3 at point C_1, which is the circle point at position 1 of phase 1. Locate C_2 and C_3 by plotting similar triangles.

In order to find a circle point for phase 2, draw a circle passing through C_1, C_2, and C_3 with the center point S. Invert point S from position 5 into position 4 to get point S_5. Draw a right bisector for the line segment SS_5; this bisector intersects the circle at D_4, which is the circle point at position 4 of phase 2. D_5 can be found by plotting a similar triangle.

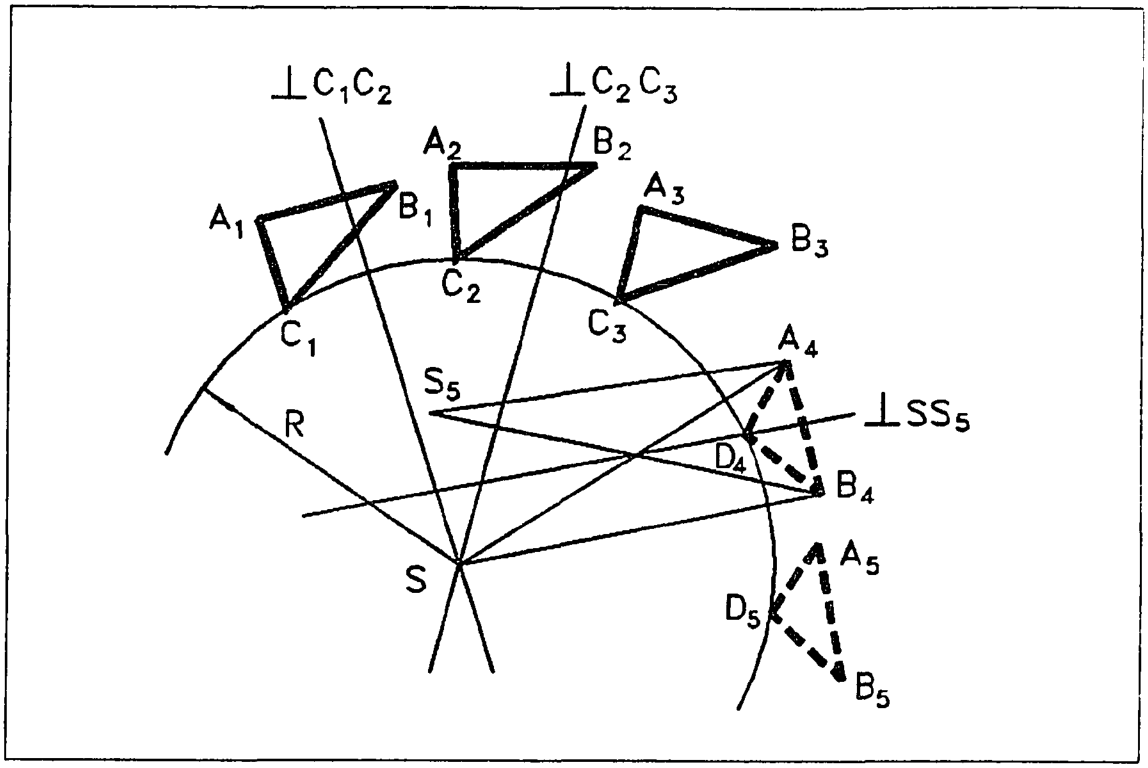

Figure 3.4 Adjustable moving pivot 123–45

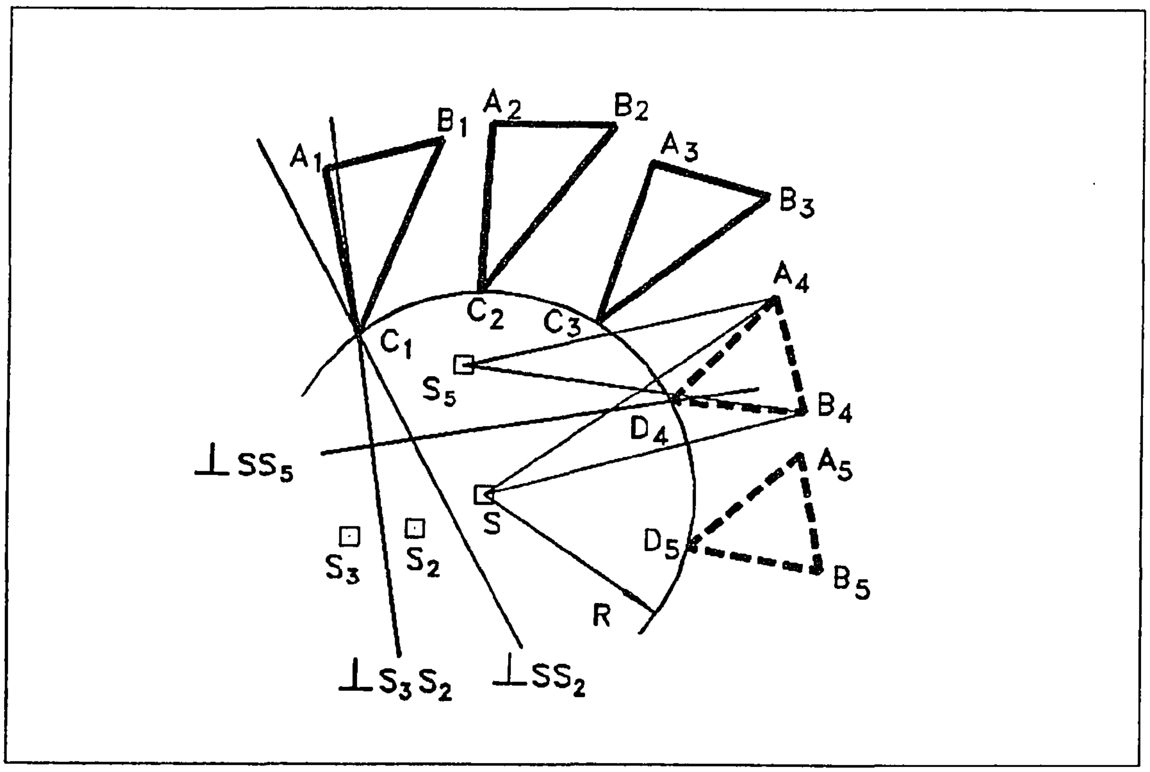

Figure 3.5 Another good solution for adjustable moving pivot 123–45

3.5 Positions 123–34

For the case of three positions in one phase, two positions in the other phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.25}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.26}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.27}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{3.28}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \qquad i = 1,2,3 \tag{3.29}$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \tag{3.30}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \qquad i = 3,4 \tag{3.31}$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 3,4 \tag{3.32}$$

Seven parameters, P, Q, p_1, q_1, p_2, q_2, and R, are involved in five equations. Thus, this problem can be solved with two free choices of parameters, and has infinite solutions. Either an algebraic method or a graphic method can to solve this problem.

Figure 3.6 represents a good solution of this problem. Choose p_1 and q_1 to locate C_1. Locate C_2 and C_3 by plotting similar triangles. Intersect two bisectors for line segments C_1C_2 and C_2C_3 at center point S. Invert point S from position 4 into position 3 to get point S_4. Draw a circle passing through C_1, C_2, and C_3 with center S; this circle intersects the right bisector for the line segment SS_4 at point D_3, which is the circle point at position 3 of phase 2. D_4 can be found by plotting a similar triangle $A_4B_4D_4$.

P and Q could also be chosen as the two free choices of parameters (Figure 3.7). Invert center point S from positions 2 and 3 into position 1 to get points S_2 and S_3. Draw bisectors for line segments SS_2 and S_2S_3 to get their intersection point C_1, which is the circle point at position 1 of phase 1. Locate C_2 and C_3 by plotting similar triangles.

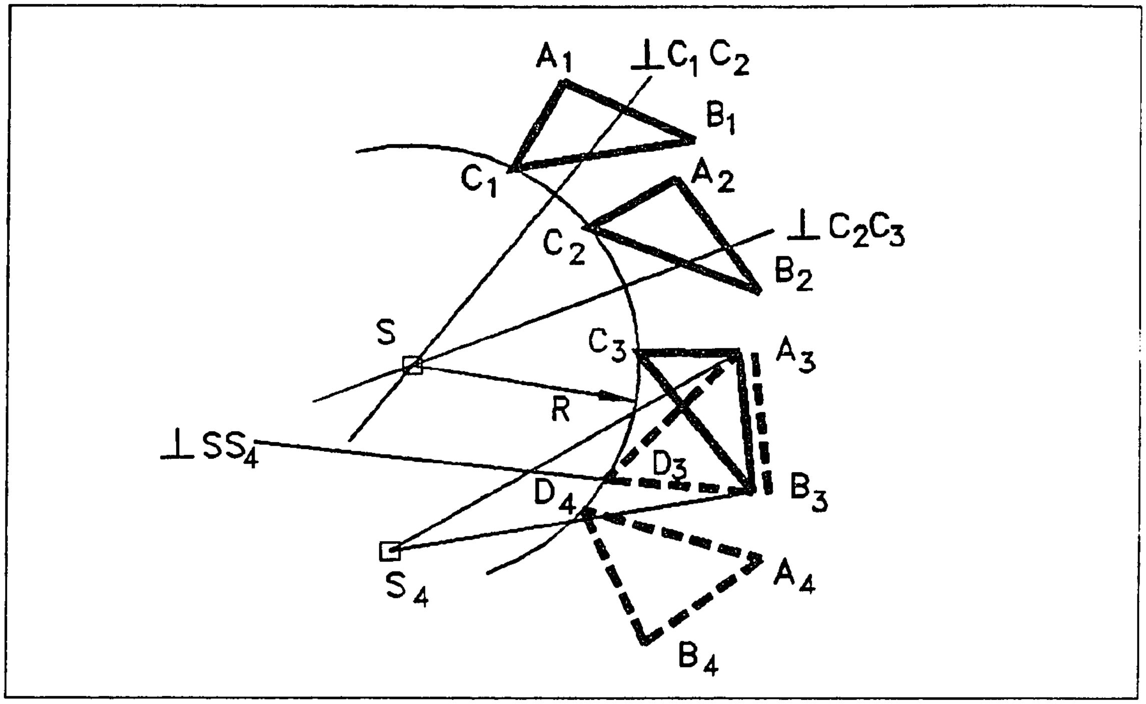

Figure 3.6 Adjustable moving pivot 123–34

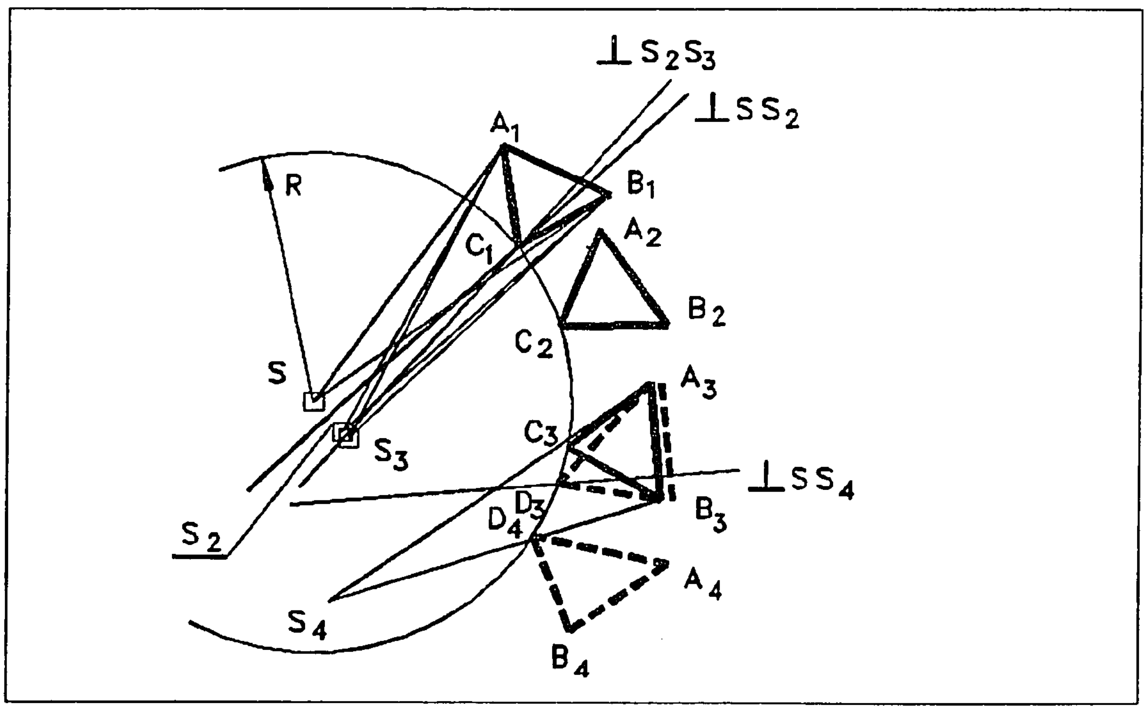

Figure 3.7 Another solution for adjustable moving pivot 123–34

In order to find a circle point for phase 2, invert point S from position 4 into position 3 to get point S_4. Draw a right bisector for the line segment SS_4; this bisector intersects the circle passing through C_1 and C_2 at D_3, which is the circle point at position 3 of phase 2.

3.6 Positions 123–456

3.6.1 Basic Equations

For the case of three positions in each of the two phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.33}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.34}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.35}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{3.36}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{3.37}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{3.38}$$

where

$$X_1 = a_1 + p_1 \cos \theta_1 - q_1 \sin \theta_1 \tag{3.39}$$

$$X_2 = a_2 + p_1 \cos \theta_2 - q_1 \sin \theta_2 \tag{3.40}$$

$$X_3 = a_3 + p_1 \cos \theta_3 - q_1 \sin \theta_3 \tag{3.41}$$

$$Y_1 = b_1 + p_1 \sin \theta_1 + q_1 \cos \theta_1 \tag{3.42}$$

$$Y_2 = b_2 + p_1 \sin \theta_2 + q_1 \cos \theta_2 \tag{3.43}$$

$$Y_3 = b_3 + p_1 \sin \theta_3 + q_1 \cos \theta_3 \tag{3.44}$$

and

$$X_4 = a_4 + p_2 \cos \theta_4 - q_2 \sin \theta_4 \tag{3.45}$$

$$X_5 = a_5 + p_2 \cos \theta_5 - q_2 \sin \theta_5 \tag{3.46}$$

$$X_6 = a_6 + p_2 \cos \theta_6 - q_2 \sin \theta_6 \tag{3.47}$$

$$Y_4 = b_4 + p_2 \sin \theta_4 + q_2 \cos \theta_4 \tag{3.48}$$

$$Y_5 = b_5 + p_2 \sin \theta_5 + q_2 \cos \theta_5 \quad (3.49)$$

$$Y_6 = b_6 + p_2 \sin \theta_6 + q_2 \cos \theta_6 \quad (3.50)$$

Seven unknowns, P, Q, p_1, q_1, p_2, q_2, and R, are included in six equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

3.6.2 Solutions at Poles

There are six rotation poles for six given positions in two phases, that is P_{12}, P_{13}, P_{23}, P_{45}, P_{46}, and P_{56}. Generally, any one of them is a good center point which satisfies the basic equations for the adjustable moving pivot problem 123–456.

Suppose six given positions are shown in Figure 3.8, and the rotation pole P_{12} (point S in Figure 3.8) is picked as the center point. Invert the center point S from positions 5 and 6 into position 4 to get points S_5 and S_6. Draw two bisectors for line segments SS_5 and S_5S_6 and intersect them at point D_4, which is the circle point at position 4 of phase 2. D_5 and D_6 can be found by plotting similar triangles.

Draw a circle passing through points D_4, D_5, and D_6 with center S. The radius of the circle is the crank length R. Invert center point S from positions 2 and 3 into position 1 to get points S_2 and S_3. Notice that S_2 coincides with S, because S is the rotation pole for positions 1 and 2. Draw a right bisector for line segment SS_3; this bisector intersects the circle at C_1, which is the circle point at position 1 of phase 1.

This indicates that Pole P_{12} satisfies the basic equations for the adjustable moving pivot 123–456 problem and so do the rest 5 rotation poles. Thus, in general, there is a solution at each of the six rotation poles.

In order to find the solution set for the adjustable moving pivot 123–456 problem, a numerical method will be used in the next section with the solutions at poles as its initial values.

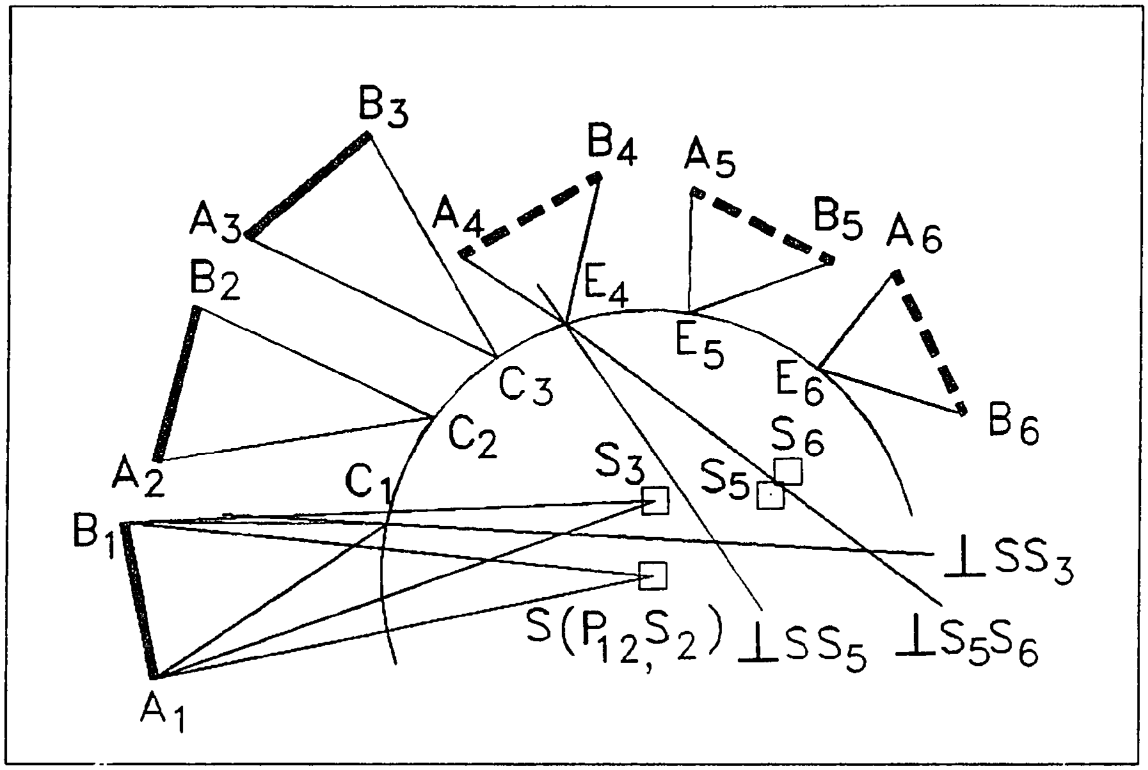

Figure 3.8 A solution at pole P_{12} for adjustable moving pivot 123–456

3.6.3 Derivation of Equations

This section derives equations for the solutions at poles. Let us find the solution at rotation pole P_{12}.

Eliminate R from equations (3.33) and (3.34), we have

$$(X_2 - P)^2 + (Y_2 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \qquad (3.51)$$

Similarly,

$$(X_3 - P)^2 + (Y_3 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \qquad (3.52)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = (X_4 - P)^2 + (Y_4 - Q)^2 \qquad (3.53)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = (X_4 - P)^2 + (Y_4 - Q)^2 \qquad (3.54)$$

Substitute equations (3.39), (3.40), (3.42), and (3.43) into equation (3.51), we have

$$
\begin{aligned}
&a_2^2 + b_2^2 + p_1^2 + q_1^2 + P^2 + Q^2 - 2Pa_2 - 2Qb_2 + \sin\theta_2\,(\,2b_2p_1 - \\
&- 2a_2q_1 + 2Pq_1 - 2Qp_1\,) + \cos\theta_2\,(\,2a_2p_1 + 2b_2q_1 - 2Pp_1 - 2Qq_1\,) \\
&= a_1^2 + b_1^2 + p_1^2 + q_1^2 + P^2 + Q^2 - 2Pa_1 - 2Qb_1 + \sin\theta_1\,(\,2b_1p_1 - \\
&- 2a_1q_1 + 2Pq_1 - 2Qp_1\,) + \cos\theta_1\,(\,2a_1p_1 + 2b_1q_1 - 2Pp_1 - 2Qq_1\,)
\end{aligned}
\tag{3.55}
$$

collect terms in p_1 and q_1, we get

$$L_{12}\,p_1 + M_{12}\,q_1 + N_{12} = 0 \tag{3.56}$$

Similarly,

$$L_{13}\,p_1 + M_{13}\,q_1 + N_{13} = 0 \tag{3.57}$$

$$L_{45}\,p_2 + M_{45}\,q_2 + N_{45} = 0 \tag{3.58}$$

$$L_{46}\,p_2 + M_{46}\,q_2 + N_{46} = 0 \tag{3.59}$$

where

$$
\begin{aligned}
L_{12} = {} & (-\cos\theta_1 + \cos\theta_2)\,P + (-\sin\theta_1 + \sin\theta_2)\,Q + \\
& + a_1\cos\theta_1 - a_2\cos\theta_2 + b_1\sin\theta_1 - b_2\sin\theta_2
\end{aligned}
\tag{3.60}
$$

$$
\begin{aligned}
M_{12} = {} & (\sin\theta_1 - \sin\theta_2)\,P + (-\cos\theta_1 + \cos\theta_2)\,Q - \\
& -a_1\sin\theta_1 + a_2\sin\theta_2 + b_1\cos\theta_1 - b_2\cos\theta_2
\end{aligned}
\tag{3.61}
$$

$$N_{12} = (-a_1 + a_2)\,P + (-b_1 + b_2)\,Q + (a_1^2 + b_1^2 - a_2^2 - b_2^2)\,/\,2 \tag{3.62}$$

$$
\begin{aligned}
L_{13} = {} & (-\cos\theta_1 + \cos\theta_3)\,P + (-\sin\theta_1 + \sin\theta_3)\,Q + \\
& + a_1\cos\theta_1 - a_3\cos\theta_3 + b_1\sin\theta_1 - b_3\sin\theta_3
\end{aligned}
\tag{3.63}
$$

$$
\begin{aligned}
M_{13} = {} & (\sin\theta_1 - \sin\theta_3)\,P + (-\cos\theta_1 + \cos\theta_3)\,Q - \\
& -a_1\sin\theta_1 + a_3\sin\theta_3 + b_1\cos\theta_1 - b_3\cos\theta_3
\end{aligned}
\tag{3.64}
$$

$$N_{13} = (-a_1 + a_3)\,P + (-b_1 + b_3)\,Q + (a_1^2 + b_1^2 - a_3^2 - b_3^2)\,/\,2 \tag{3.65}$$

$$
\begin{aligned}
L_{45} = {} & (-\cos\theta_4 + \cos\theta_5)\,P + (-\sin\theta_4 + \sin\theta_5)\,Q + \\
& + a_4\cos\theta_4 - a_5\cos\theta_5 + b_4\sin\theta_4 - b_5\sin\theta_5
\end{aligned}
\tag{3.66}
$$

$$M_{45} = (\sin\theta_4 - \sin\theta_5)\,P + (-\cos\theta_4 + \cos\theta_5)\,Q -$$

$$-a_4 \sin\theta_4 + a_5 \sin\theta_5 + b_4 \cos\theta_4 - b_5 \cos\theta_5 \quad (3.67)$$

$$N_{45} = (-a_4 + a_5) P + (-b_4 + b_5) Q + (a_4^2 + b_4^2 - a_5^2 - b_5^2) / 2 \quad (3.68)$$

$$L_{46} = (-\cos\theta_4 + \cos\theta_6) P + (-\sin\theta_4 + \sin\theta_6) Q +$$

$$+ a_4 \cos\theta_4 - a_6 \cos\theta_6 + b_4 \sin\theta_4 - b_6 \sin\theta_6 \quad (3.69)$$

$$M_{46} = (\sin\theta_4 - \sin\theta_6) P + (-\cos\theta_4 + \cos\theta_6) Q -$$

$$-a_4 \sin\theta_4 + a_6 \sin\theta_6 + b_4 \cos\theta_4 - b_6 \cos\theta_6 \quad (3.70)$$

$$N_{46} = (-a_4 + a_6) P + (-b_4 + b_6) Q + (a_4^2 + b_4^2 - a_6^2 - b_6^2) / 2 \quad (3.71)$$

Simultaneous equations (3.58) and (3.59) can be used to solve for p_2 and q_2, while any of the poles P_{12}, P_{13}, and P_{23} is picked as the center point at which an initial solution is calculated. Similarly, p_1 and q_1 can also be solved by simultaneous equations (3.56) and (3.57), while an initial solution is calculated at poles P_{45}, P_{46}, and P_{56}.

In the former case,

$$p_2 = (N_{46} M_{45} - N_{45} M_{46}) / (L_{45} M_{46} - L_{46} M_{45}) \quad (3.72)$$

$$q_2 = (L_{46} N_{45} - L_{45} N_{46}) / (L_{45} M_{46} - L_{46} M_{45}) \quad (3.73)$$

Solve for crank length R with equation (3.36)

$$R = [\, (X_4 - P)^2 + (Y_4 - Q)^2 \,]^{0.5} \quad (3.74)$$

where X_4 and Y_4 can be solved by equations (3.45) and (3.48).

Solve p_1 by equations (3.33), (3.39), (3.42), and (3.56)

$$p_{1,1} = [-B + (B^2 - 4\,A\,C)^{0.5}] / (2\,A) \quad (3.75)$$

$$p_{1,2} = [-B - (B^2 - 4\,A\,C)^{0.5}] / (2\,A) \quad (3.76)$$

where $p_{1,1}$ and $p_{1,2}$ are the first and second roots for p_1, and

$$A = G^2 + E^2 \quad (3.77)$$

$$B = [2\,G\,(D - P) + 2\,E\,(F - Q)\,] \quad (3.78)$$

$$C = (D - P)^2 + (F - Q)^2 - R^2 \quad (3.79)$$

$$D = a_1 + N_{12} \sin\theta_1 / M_{12} \quad (3.80)$$

$$E = \sin\theta_1 - L_{12} \cos\theta_1 / M_{12} \quad (3.81)$$

$$F = b_1 - N_{12} \cos\theta_1 / M_{12} \tag{3.82}$$

$$G = \cos\theta_1 + L_{12} \sin\theta_1 / M_{12} \tag{3.83}$$

Two corresponding values of q_1 can be solved by the following two equations:

$$q_{1,1} = (-L_{12}\, p_{1,1} - N_{12}) / M_{12} \tag{3.84}$$

$$q_{1,2} = (-L_{12}\, p_{1,2} - N_{12}) / M_{12} \tag{3.85}$$

Thus far, all seven unknown parameters have been determined. Note that we solve p_2 and q_2 prior to solving p_1 and q_1, because equations (3.56) and (3.57) can not be used to solve p_1 and q_1 while the solution is calculated at poles P_{12}, P_{13}, and P_{23}. The geometric explanation is simple. No right bisector can be constructed for two points which coincide with each other, and inverting a rotation pole results in coincident points. Geometrically, each one of equations (3.56) through (3.59) represents a right bisector.

Algebraically, the coefficients L_{12} and M_{12} in equation (3.56) equal to zero when P_{12} is picked as the calculation point, which makes this equation meaningless. Likewise, the coefficients L_{13} and M_{13} also equal to zero while solving at pole P_{13}.

Similarly, we should solve p_1 and q_1 by equations (3.56) and (3.57) prior to solving p_2 and q_2 while calculating at poles P_{45}, P_{46}, and P_{56}.

3.6.4 More Solutions

We have six solutions at six poles so far. In order to find more solutions for the adjustable moving pivot 123–456 problem, a numerical method similar to that developed by Wilhelm [12] can be used to solve this problem.

Let Functions F_i equal to the following equations derived from equations (3.33) through (3.38) and equations (3.39) through (3.50)

$$F_i = (a_i + p_1 \cos\theta_i - q_1 \sin\theta_i - P)^2 + (b_i + p_1 \sin\theta_i + \\ + q_1 \cos\theta_i - Q)^2 - R^2 = 0 \qquad i = 1,2,3 \tag{3.86}$$

$$F_i = (a_i + p_2 \cos \theta_i - q_2 \sin \theta_i - P)^2 + (b_i + p_2 \sin \theta_i + $$
$$+ q_2 \cos \theta_i - Q)^2 - R^2 = 0 \qquad i = 4,5,6 \qquad (3.87)$$

Seven parameters are included in the above six equations. The number of the parameters is reduced to six after assigning a value to R. Substitute a solution at pole into above equations. All six equations should be satisfied and equal to zero.

Suppose the value of R is increased or decreased by ΔR. The above six equations will no longer be equal to zero. In order to satisfy the basic equations of this problem, the values of the rest six parameters should be increased or decreased properly by an increment or decrement to make the six equations back to zero again. The modified values of the seven parameters constitute a new solution point other than the pole point, but pretty close to it.

The new solution point is used as a new initial point, and the new values of the seven parameters are treated as the new initial values in the next iteration of calculation to find another solution point which is close to it.

The solution points appear starting at each rotation poles in four different directions. As shown in Figure 3.9, two groups of values of p_1, q_1, p_2, and q_2 along with increment and decrement of R result in four branches of curves at each pole of the sample problem.

Equations (3.86) and (3.87) consists of six equations, but we can also think they are six functions to be solved for solutions every time R gets an increment or decrement ΔR.

The Newton-Raphson Method has been used to get the solutions numerically. The following simultaneous equations should be solved for the numerical solutions:

$$\begin{bmatrix} \frac{\partial F_1}{\partial P} & \frac{\partial F_1}{\partial Q} & \frac{\partial F_1}{\partial p_1} & \frac{\partial F_1}{\partial q_1} & \frac{\partial F_1}{\partial p_2} & \frac{\partial F_1}{\partial q_2} \\ \frac{\partial F_2}{\partial P} & \frac{\partial F_2}{\partial Q} & \frac{\partial F_2}{\partial p_1} & \frac{\partial F_2}{\partial q_1} & \frac{\partial F_2}{\partial p_2} & \frac{\partial F_2}{\partial q_2} \\ \frac{\partial F_3}{\partial P} & \frac{\partial F_3}{\partial Q} & \frac{\partial F_3}{\partial p_1} & \frac{\partial F_3}{\partial q_1} & \frac{\partial F_3}{\partial p_2} & \frac{\partial F_3}{\partial q_2} \\ \frac{\partial F_4}{\partial P} & \frac{\partial F_4}{\partial Q} & \frac{\partial F_4}{\partial p_1} & \frac{\partial F_4}{\partial q_1} & \frac{\partial F_4}{\partial p_2} & \frac{\partial F_4}{\partial q_2} \\ \frac{\partial F_5}{\partial P} & \frac{\partial F_5}{\partial Q} & \frac{\partial F_5}{\partial p_1} & \frac{\partial F_5}{\partial q_1} & \frac{\partial F_5}{\partial p_2} & \frac{\partial F_5}{\partial q_2} \\ \frac{\partial F_6}{\partial P} & \frac{\partial F_6}{\partial Q} & \frac{\partial F_6}{\partial p_1} & \frac{\partial F_6}{\partial q_1} & \frac{\partial F_6}{\partial p_2} & \frac{\partial F_6}{\partial q_2} \end{bmatrix} \begin{bmatrix} dP \\ dQ \\ dp_1 \\ dq_1 \\ dp_2 \\ dq_2 \end{bmatrix} = \begin{bmatrix} -F_1 \\ -F_2 \\ -F_3 \\ -F_4 \\ -F_5 \\ -F_6 \end{bmatrix} \qquad (3.88)$$

where

$$\frac{\partial F_i}{\partial P} = -2\,(a_i + p_1 \cos\theta_i - q_1 \sin\theta_i - P)$$

$$\frac{\partial F_i}{\partial Q} = -2\,(b_i + p_1 \sin\theta_i + q_1 \cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial p_1} = 2\cos\theta_i\,(a_i + p_1 \cos\theta_i - q_1 \sin\theta_i - P) + {}+ 2\sin\theta_i\,(b_i + p_1 \sin\theta_i + q_1 \cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial q_1} = -2\sin\theta_i\,(a_i + p_1 \cos\theta_i - q_1 \sin\theta_i - P) + {}+ 2\cos\theta_i\,(b_i + p_1 \sin\theta_i + q_1 \cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial p_2} = 0$$

$$\frac{\partial F_i}{\partial q_2} = 0 \qquad i = 1,2,3 \qquad (3.89)$$

$$\frac{\partial F_i}{\partial P} = -2\,(a_i + p_2 \cos\theta_i - q_2 \sin\theta_i - P)$$

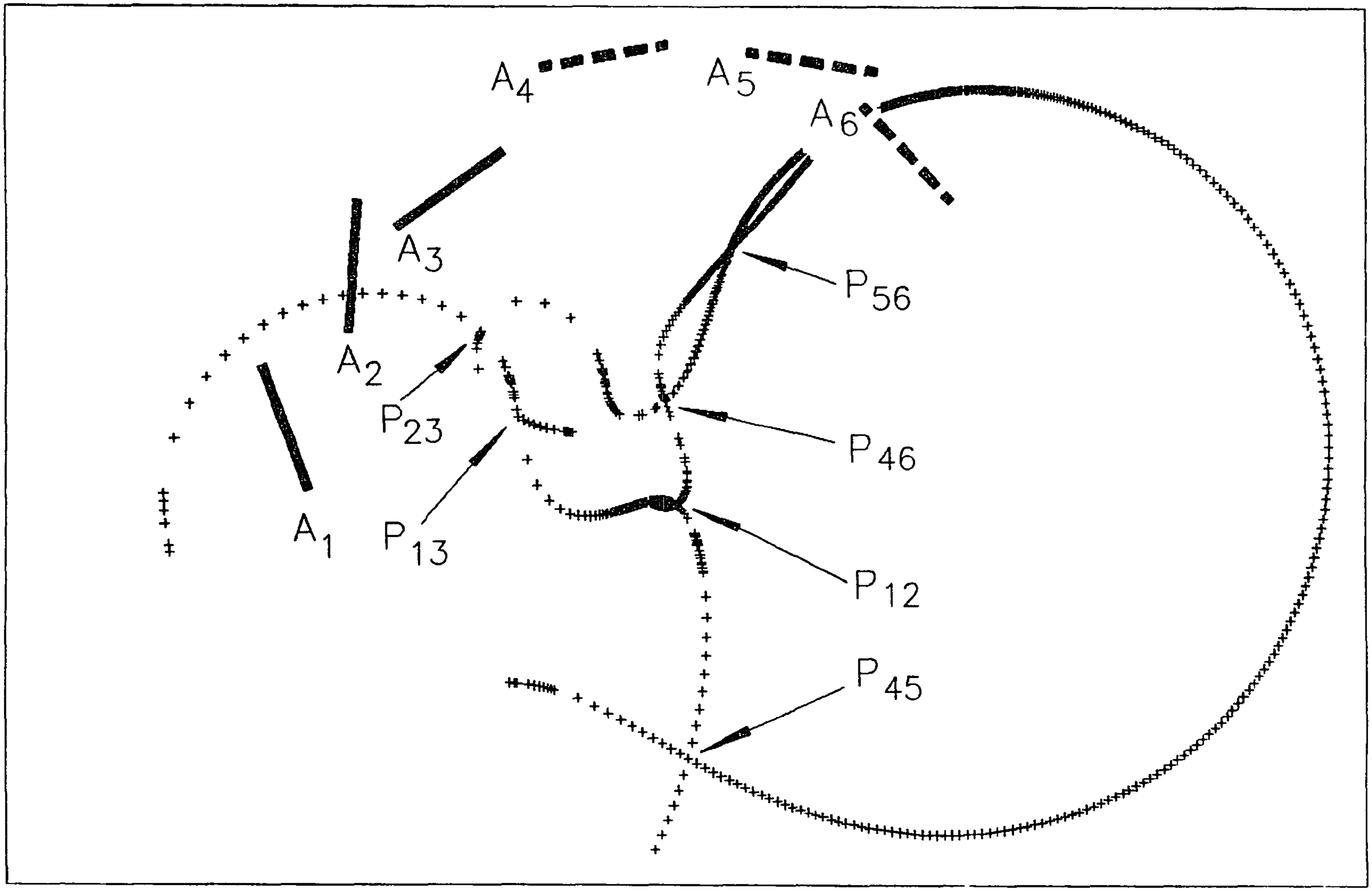

Figure 3.9 Center points for adjustable moving pivot 123–456

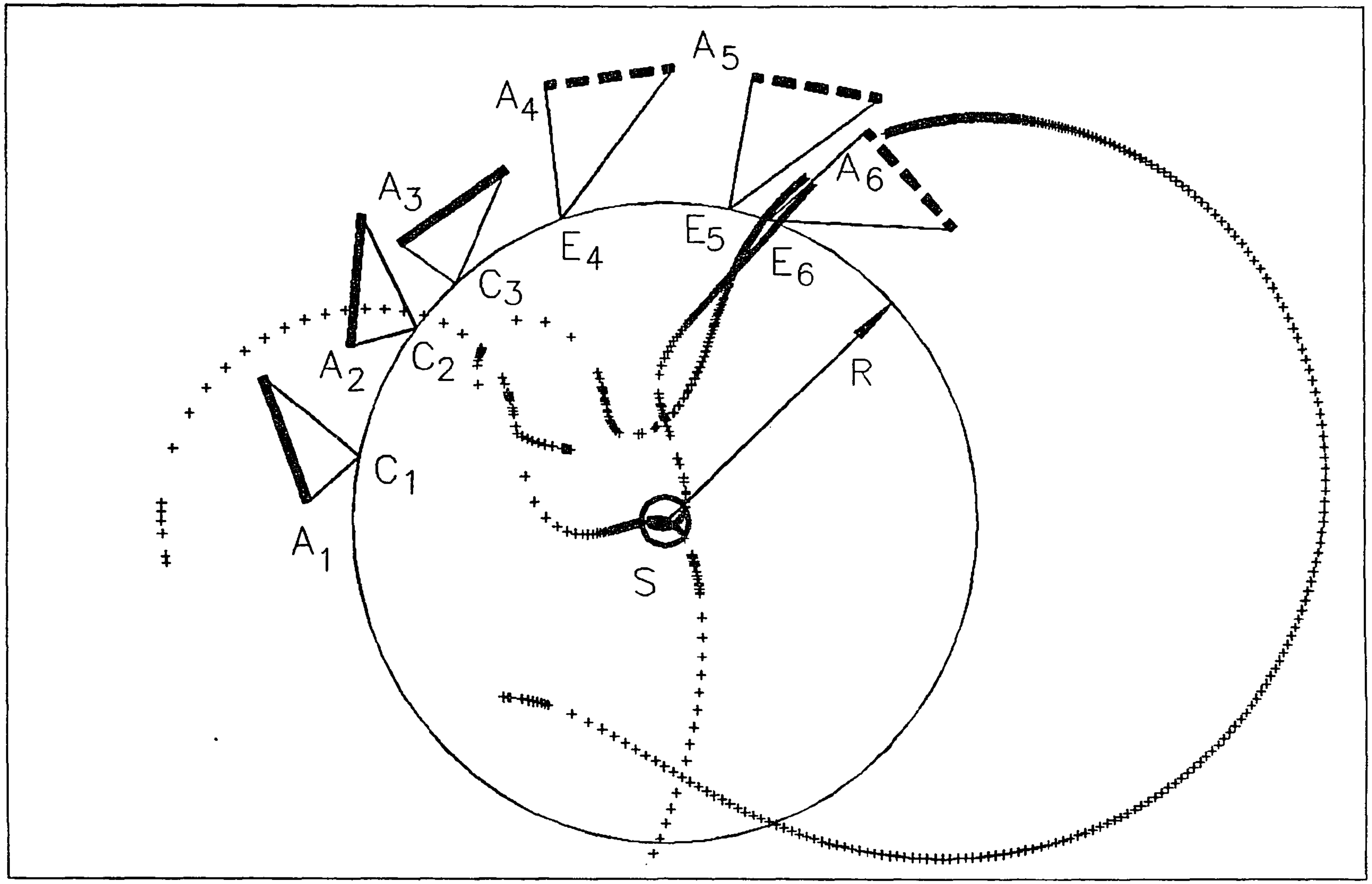

Figure 3.10 A solution for adjustable moving pivot 123–456

$$\frac{\partial F_i}{\partial Q} = -2\,(b_i + p_2 \sin\theta_i + q_2 \cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial p_2} = 2\cos\theta_i\,(a_i + p_2\cos\theta_i - q_2\sin\theta_i - P) + $$
$$+\,2\sin\theta_i\,(b_i + p_2\sin\theta_i + q_2\cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial q_2} = -2\sin\theta_i\,(a_i + p_2\cos\theta_i - q_2\sin\theta_i - P) + $$
$$+\,2\cos\theta_i\,(b_i + p_2\sin\theta_i + q_2\cos\theta_i - Q)$$

$$\frac{\partial F_i}{\partial p_1} = 0$$

$$\frac{\partial F_i}{\partial q_1} = 0 \qquad i = 4,5,6 \qquad (3.90)$$

Program MP_3_3.PAS is designed to find center points numerically with solutions at all six poles as the initial solutions.

Figure 3.10 represents a good solution of a sample problem. A center point S is chosen on the center point curve in the figure. The circle points C_1 and E_4 can be found by kinematic inversion. Circle points C_2, C_3, E_5, and E_6 can be found by geometric similarity.

As shown in Figure 3.10, circle points C and E for phases 1 and 2 are two distinct points, and their positions C_1, C_2, C_3, E_4, E_5, and E_6 lie on the same circle with a unique center point S and radius R, which satisfies the given requirement. This indicates the validity of both the method and the program MP_3_3.PAS.

It is also found by inspection that Figure 3.10 is not only a solution of the equations but also a good solution without order defect.

3.7 Positions 123–345

For the case of three positions in each of the two phases with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \quad (3.91)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \quad (3.92)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \quad (3.93)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \quad (3.94)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \quad (3.95)$$

where

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \quad (3.96)$$

and

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 3,4,5 \quad (3.97)$$

Seven unknowns, P, Q, p_1, q_1, p_2, q_2, and R, are included in six equations. Thus, the equation set can be solved with one free choice of parameter, and has infinite solutions.

The discussion in section 3.6 also applies for this case with a little modification. Similar to the program MP_3_3.PAS, the Turbo Pascal program MP_3_3_1.PAS is designed to find good center points which satisfy the basic equations above.

An example problem is shown in Figure 3.11, in which five prescribed positions are drawn. The center points for problem MP_3_3_1 are displayed by running program MP_3_3_1.PAS and calling user-defined AutoLISP function PTS_+. Every point represented by a cross sign in the figure is a good point which satisfies the basic equations.

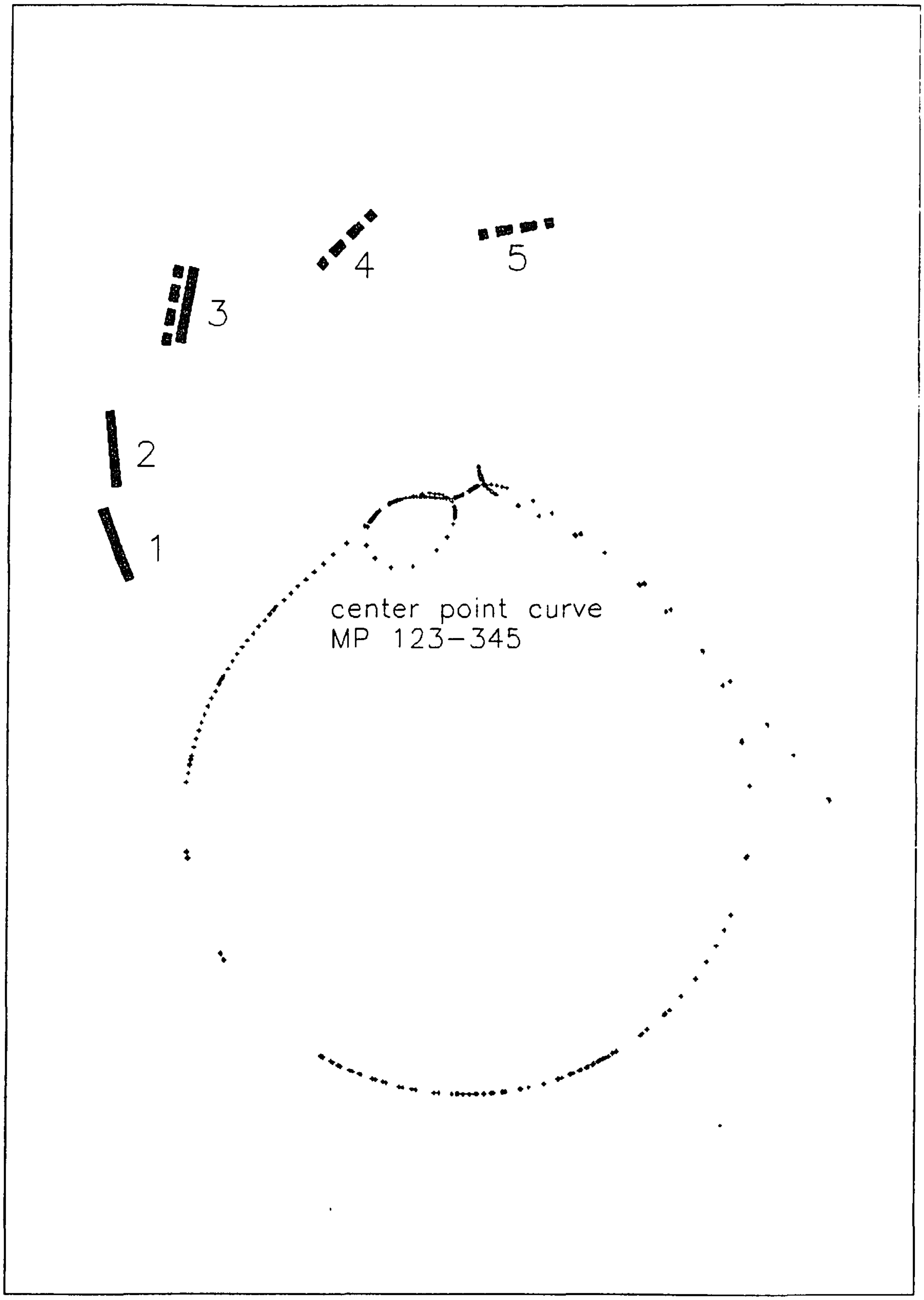

Figure 3.11 Center points for adjustable moving pivot 123–345

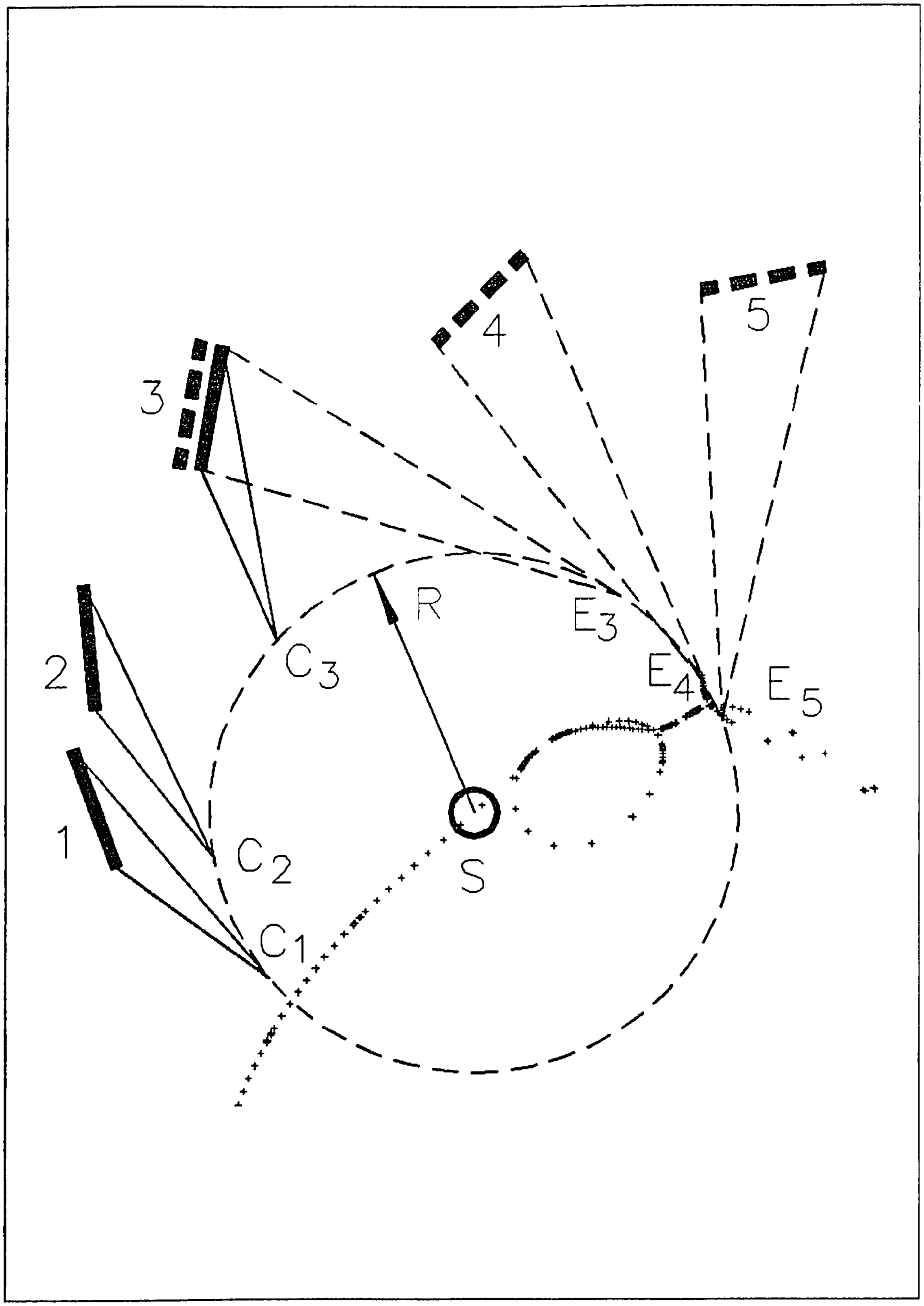

Figure 3.12 A solution for the problem of adjustable moving pivot 123–345

A center point S is picked along the curve in Figure 3.12. The circle points C_1 and E_3 can be found by kinematic inversion for phase 1 and 2 respectively. The circle points for other positions C_2, C_3, E_4, and E_5 are found by means of geometric similarity.

As shown in the figure, circle points C and E for phases 1 and 2 are two distinct points, and their positions C_1, C_2, C_3, E_3, E_4, and E_5 lie on the same circle with a unique center point S and radius R, which satisfies the given requirement. This indicates the validity of both the method and the program MP_3_3_1.PAS.

It is also found by inspection that Figure 3.12 is not only a solution of the equations but also a good solution without order defect.

3.8 Positions 1234–567

For the case of four positions in the first phase and three positions in the second phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{3.98}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{3.99}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{3.100}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{3.101}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{3.102}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{3.103}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{3.104}$$

where

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4 \tag{3.105}$$

and

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$
$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 5,6,7 \qquad (3.106)$$

Seven unknowns, P, Q, p_1, q_1, p_2, q_2, and R, are involved in seven equations. The number of positions reaches its maximum value. Thus, the equation set has no free choice of parameter.

Suppose seven given positions are shown in Figure 3.13. Plot center points in the figure for positions 123–567 by means of Turbo Pascal program MP_3_3.PAS and user–defined AutoLISP function PTS_+. Every center point in the figure satisfies the basic equations of the MP 123–567 problem.

Plot center point curve in Figure 3.14 for positions 1234 by using the Turbo Pascal Program CENT_PT.PAS. Figure 3.15 overlays Figure 3.13 with Figure 3.14 to get the intersection points. A good center point for the problem is found at the intersection point S of the two curves. Figure 3.16 is an enlarged view in the vicinity of the intersection point.

Call user–defined AutoLISP function INVERT for both phases 1 and 2 to display the solution. As shown in Figure 3.15, C and E are distinct circle points for phases 1 and 2 respectively. The circle with a center at the unique center point S and radius R precisely passes through seven circle points C_1, C_2, C_3, C_4, E_5, E_6, and E_7, which satisfies the given requirement. This indicates the validity of both the method and the programs.

Also, no order defect occurs in Figure 3.15 by inspection, which indicates that it is a good solution.

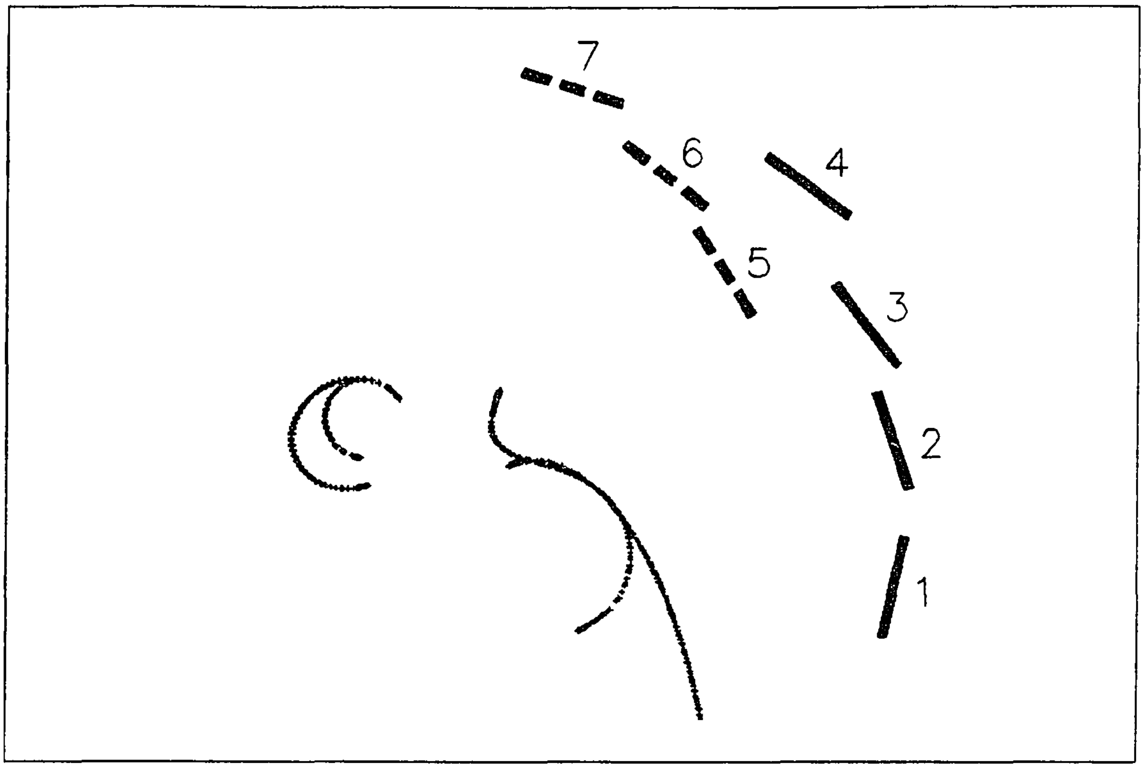

Figure 3.13 Seven given positions and the center point curve for MP 123–567

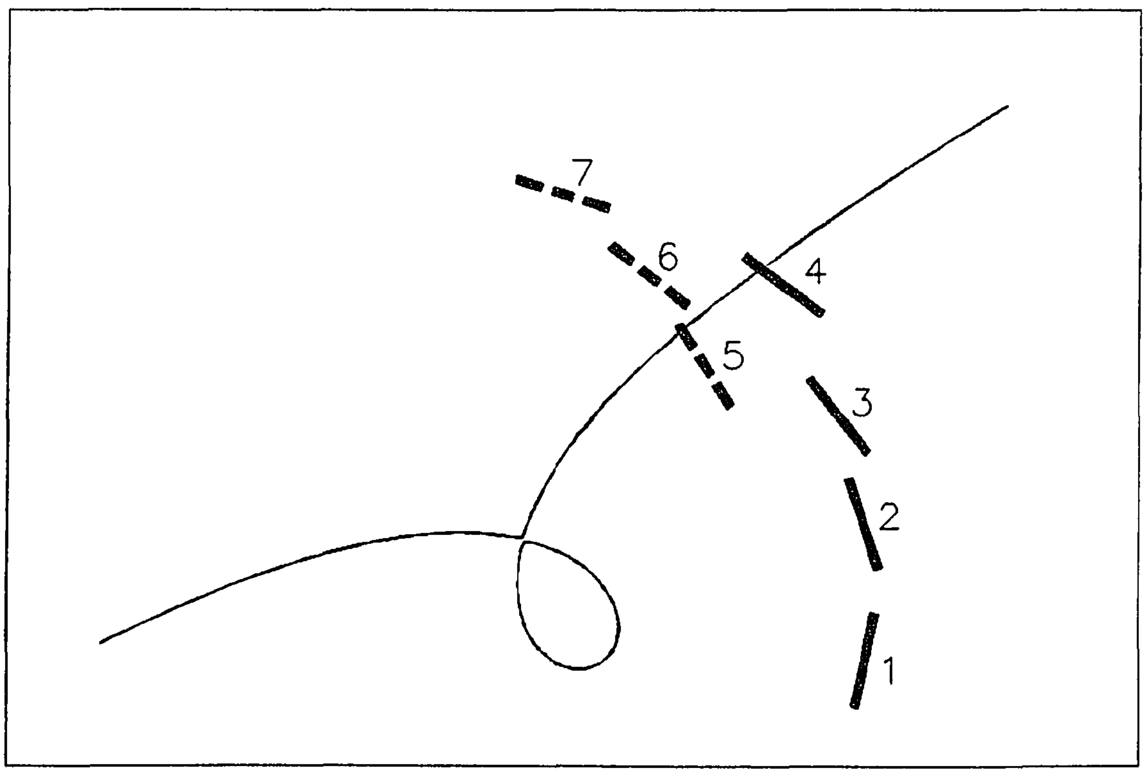

Figure 3.14 Center point curve for positions 1, 2, 3, and 4

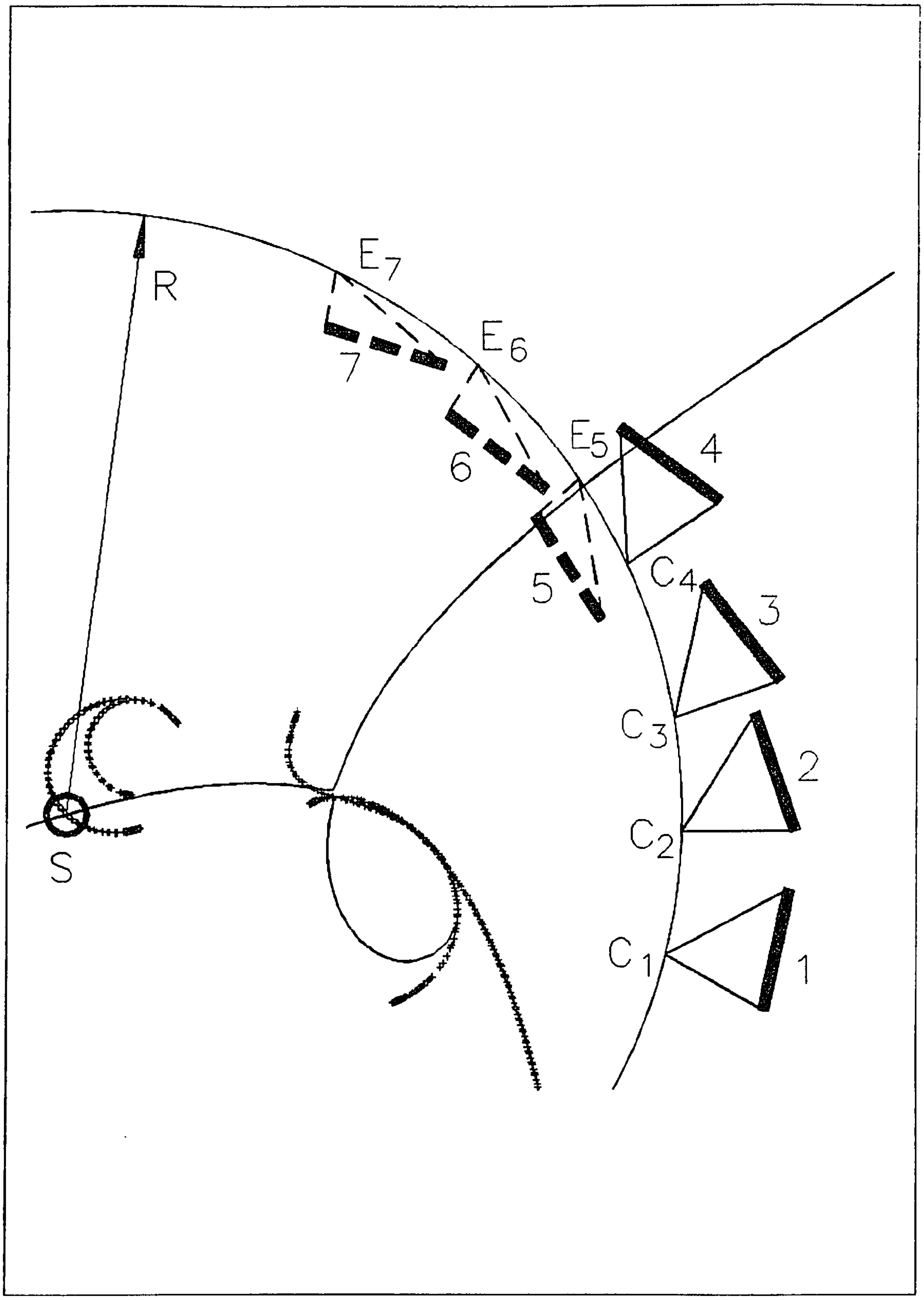

Figure 3.15 A solution for problem MP 1234–567

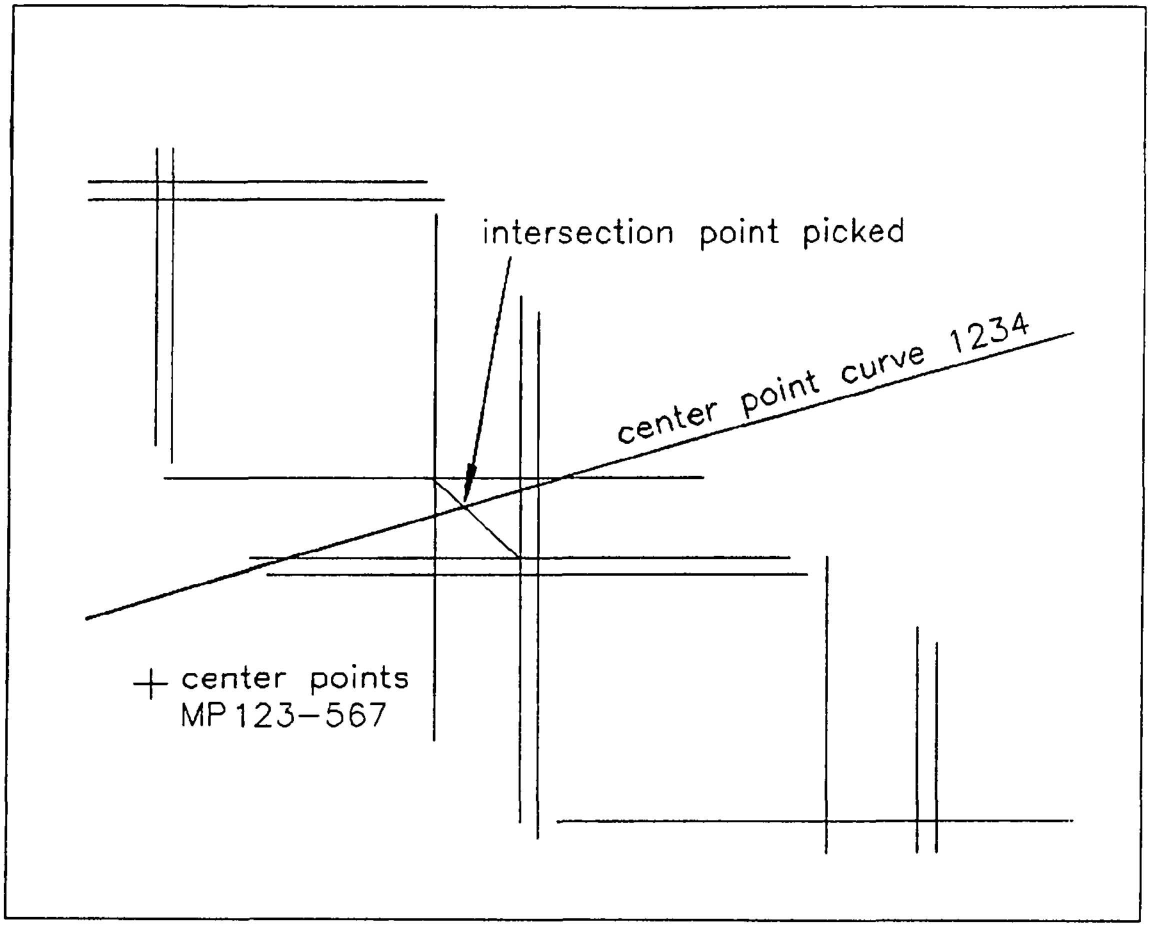

Figure 3.16 An enlarged view at an intersection point of center point curves MP 123–567 and CENT_PT 1234

3.9 Positions 1234–456

For the case of four positions in the first phase and three positions in the second phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \quad (3.107)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \quad (3.108)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \quad (3.109)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \quad (3.110)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \quad (3.111)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \qquad (3.112)$$

where

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$
$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3,4 \qquad (3.113)$$

and

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$
$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5,6 \qquad (3.114)$$

Seven unknowns, P, Q, p_1, q_1, p_2, q_2, and R, are involved in seven equations. The number of positions have reached its maximum value. Thus, the number of free choice of parameter is zero.

Suppose seven given positions are shown in Figure 3.17. Plot center points in the figure for positions 123–456 by means of Turbo Pascal program MP_3_3.PAS and user–defined AutoLISP function PTS_+. Every center point in the figure satisfies the basic equations of the MP 123–456 problem.

Plot center point curve in the same figure for positions 1234 by using the Turbo Pascal program CENT_PT.PAS. A good center point for the problem is found at the intersection point S of the two curves. Figure 3.18 is an enlarged view in the vicinity of the intersection point.

Call user–defined AutoLisp function INVERT for both phases 1 and 2 to display the solution at S. As shown in Figure 3.17, C and E are distinct circle points for phases 1 and 2 respectively. The circle with a center at the unique center point S and radius R precisely passes through seven circle points C_1, C_2, C_3, C_4, E_4, E_5, and E_6, which satisfies the given requirement. This indicates the validity of both the method and the programs.

Also, no order defect occurs in Figure 3.17 by inspection, which indicates that it is a good solution.

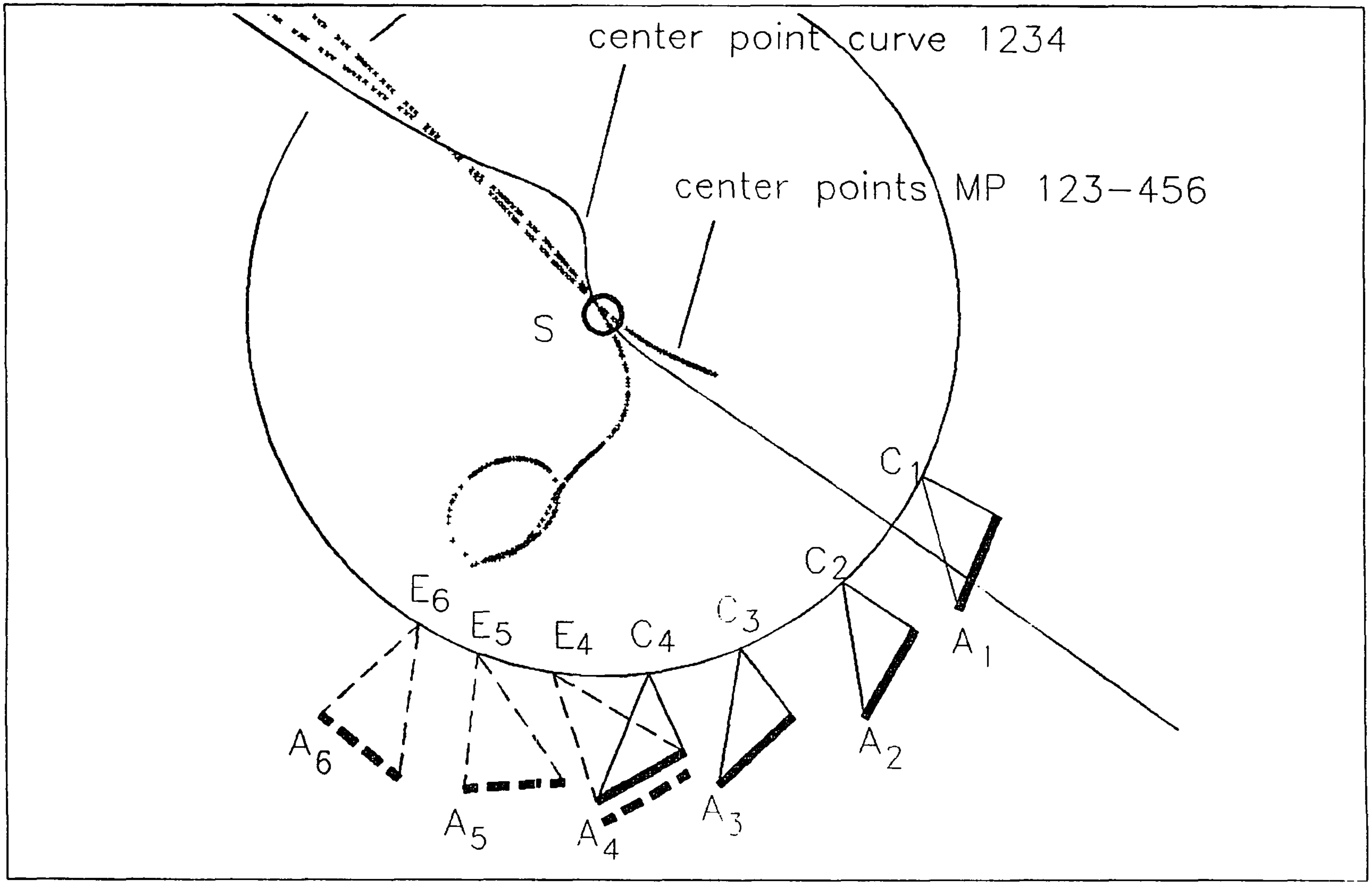

Figure 3.17 A solution for problem MP 1234–456

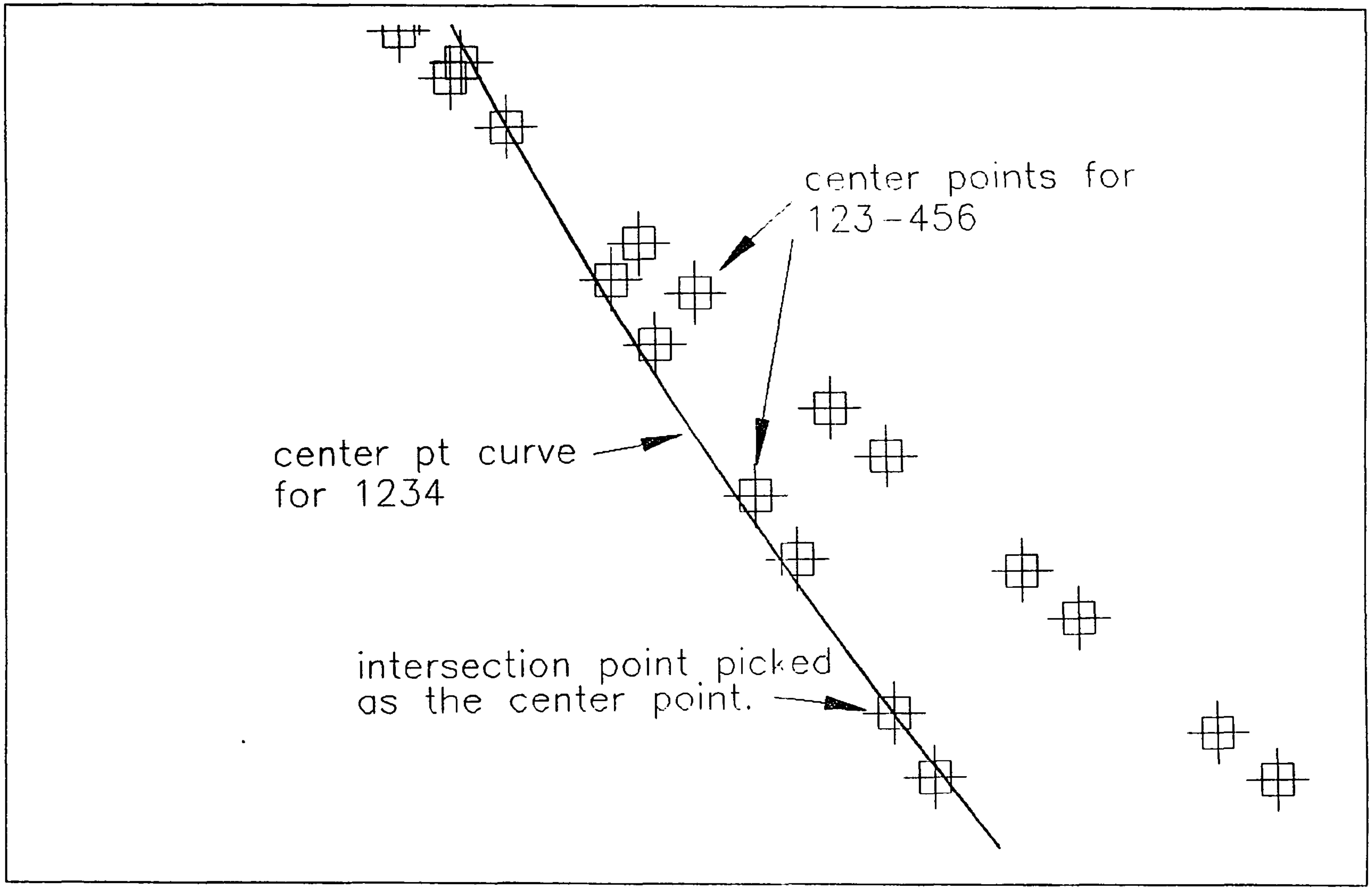

Figure 3.18 An enlarged view at a solution point for problem MP 1234-456

Chapter 4

Two Phase Adjustable Moving Pivot and Crank Length Problems

4.1 Introduction

This chapter deals with the problem of adjustable moving pivot and crank length. In the last chapter, the adjustable parameters are p and q, which are the relative coordinates of the circle point. In the problem of this chapter, one more adjustable parameter, the crank length R is added. Thus, eight parameters are needed to be determined, they are P, Q, p_1, q_1, p_2, q_2, R_1, and R_2.

Similar to the last chapter, only one side of an adjustable four–bar linkage will be considered in this chapter. The technique of adding one more crank to complete a linkage design will be shown in chapter 8.

Twelve adjustable moving pivot and crank length problems listed in Table 4.1 are solved in this chapter. The minimum number of prescribed positions included in one phase is considered to be two. The maximum number of prescribed positions included in one phase is five, which is the maximum allowable number. The last four problems in the table deal with eight prescribed positions, which is the maximum allowable value for the problem. The number of shared positions is zero or one.

4.2 Positions 123–456

For the case of three positions in each of the two phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \qquad (4.1)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.2}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.3}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{4.4}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.5}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.6}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \tag{4.7}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$

Table 4.1 Adjustable moving pivot and crank length

positions ph.1	positions ph.2	number of shared pos.	number of unknowns	number of free choices
1,2,3	4,5,6	0	6	2
1,2,3	3,4,5	1	6	2
1,2,3,4	5,6	0	6	2
1,2,3,4	4,5	1	6	2
1,2,3,4	5,6,7	0	7	1
1,2,3,4	4,5,6	1	7	1
1,2,3,4,5	6,7	0	7	1
1,2,3,4,5	5,6	1	7	1
1,2,3,4	5,6,7,8	0	8	0
1,2,3,4	4,5,6,7	1	8	0
1,2,3,4,5	6,7,8	0	8	0
1,2,3,4,5	5,6,7	1	8	0

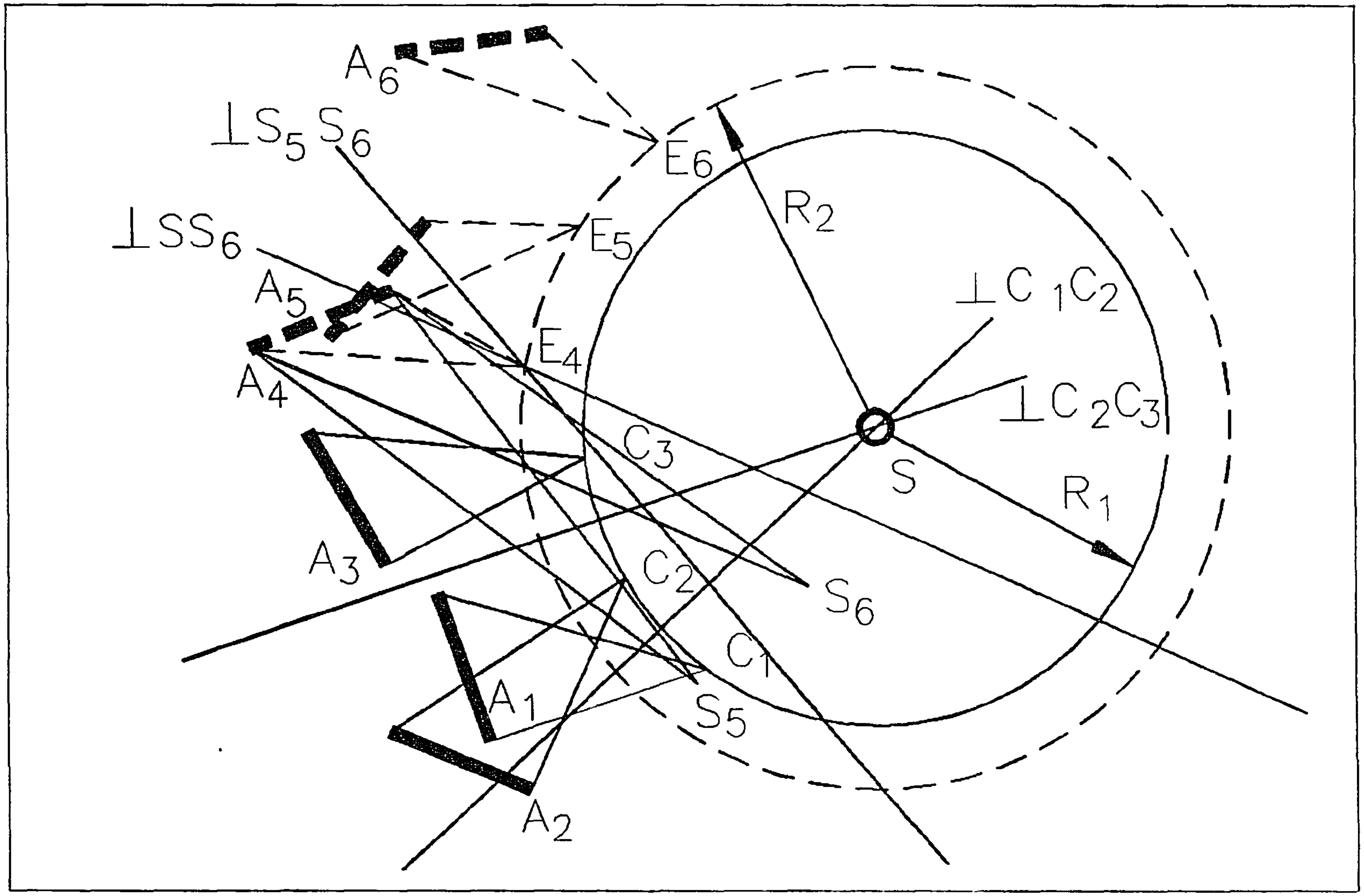

Figure 4.1 Adjustable moving pivot and crank length 123–456

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5,6 \tag{4.8}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in six equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

The two free choices could be the absolute coordinates of the center point (P, Q), the relative coordinates of the circle point of phase 1 (p_1, q_1), or that of phase 2 (p_2, q_2).

Suppose six prescribed positions are shown in Figure 4.1, and the relative coordinates of the circle point C_1 are chosen as the two free choices. Find C_2 and C_3 by geometric similarity after locating C_1. Intersect right bisectors for line segments C_1C_2 and C_2C_3 at point S, which is the center point. Invert point S from positions 5 and 6 into position 4 to get points S_5 and S_6. Intersect right bisectors for line segments SS_6 and S_5S_6 at E_4, which is the circle point at position 4 of phase 2. E_5 and E_6 can be found by geometric similarity.

4.3 Positions 123–345

For the case of three positions in each of the two phases with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.9}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.10}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.11}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_2^2 \tag{4.12}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{4.13}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.14}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$
$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \qquad (4.15)$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$
$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 3,4,5 \qquad (4.16)$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in six equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

The two free choices could be the absolute coordinates of the center point (P, Q), the relative coordinates of the circle point of phase 1 (p_1, q_1), or that of phase 2 (p_2, q_2).

Suppose five prescribed positions are shown in Figure 4.2. The coordinates of the center point S are chosen as the two free choices in this case, although it can be solved in the same way as that for the problem MC 123–456 in the previous section.

Choose center point S on the plane. Invert point S from positions 2 and 3 into position 1 to get points S_2 and S_3. Intersect right bisectors for line segments SS_2 and S_2S_3 at point C_1, which is the circle point at position 1. C_2 and C_3 can be found by geometric similarity.

Similarly, invert point S from positions 4 and 5 into position 3 to get points S_4 and S_5. Intersect right bisectors for line segments SS_5 and S_4S_5 at point E_3, which is the circle point at position 3 of phase 2. E_4 and E_5 can be found by geometric similarity.

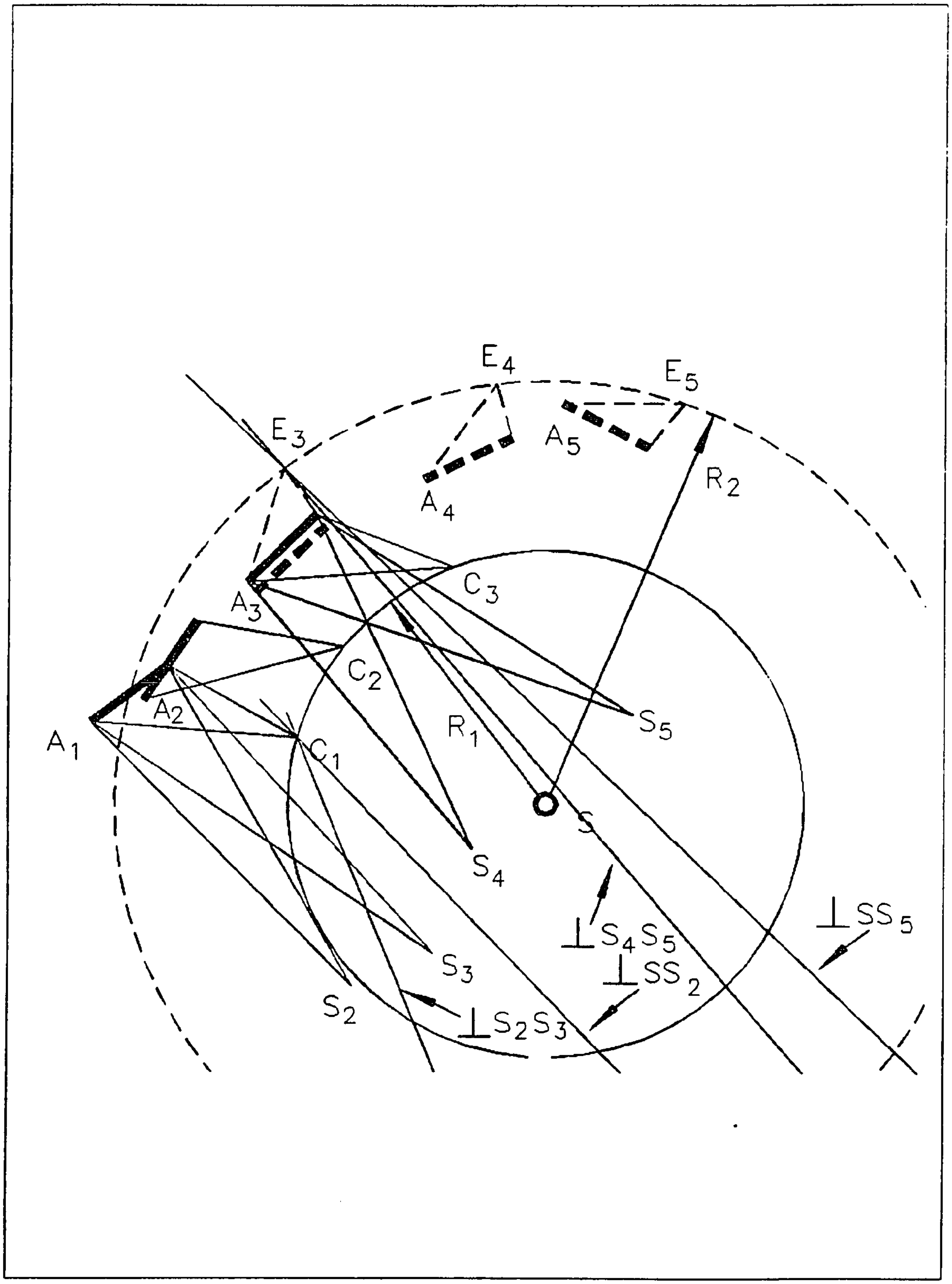

Figure 4.2 Adjustable moving pivot and crank length 123–345

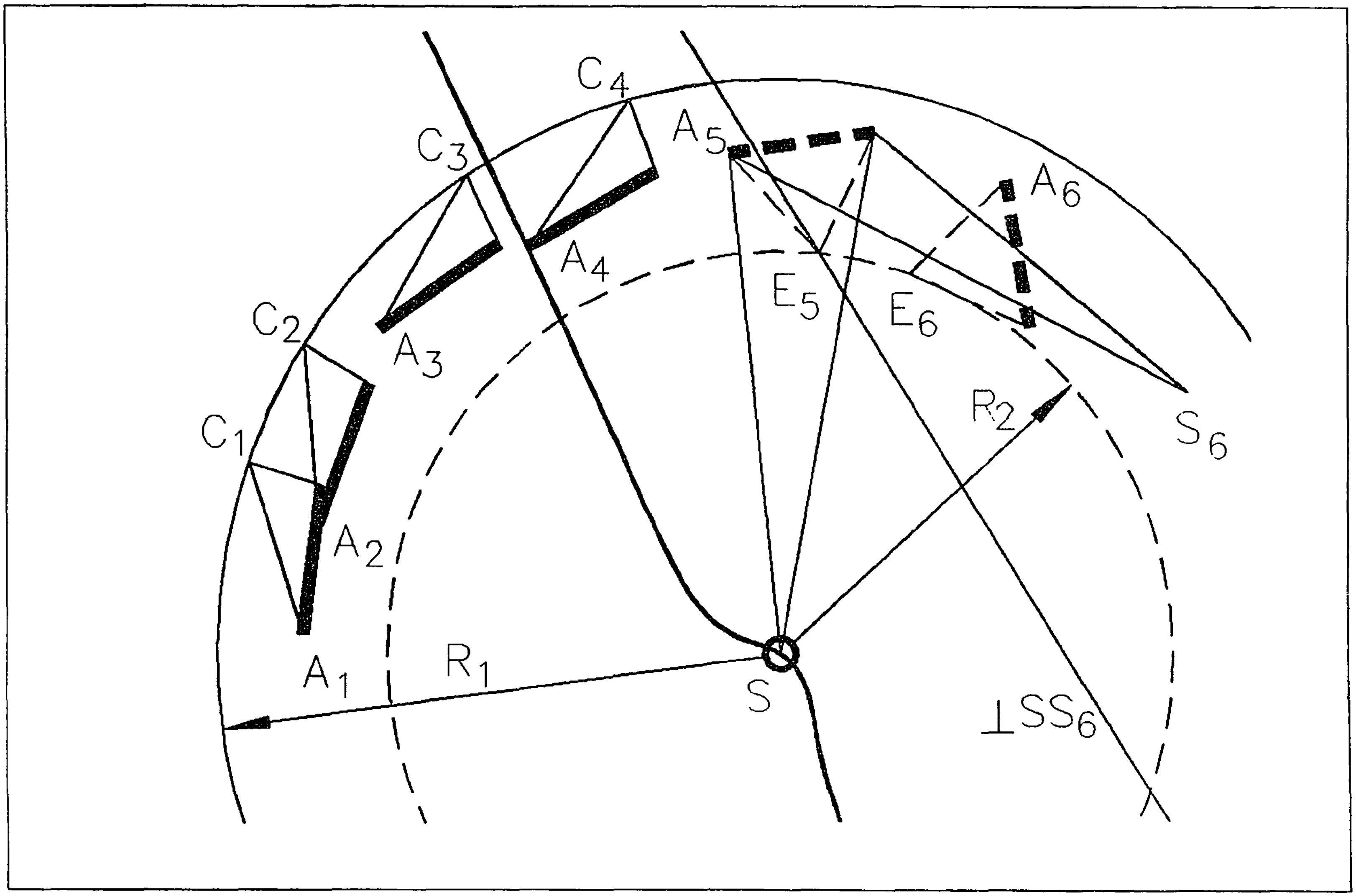

Figure 4.3 Adjustable moving pivot and crank length 1234–56

4.4 Positions 1234–56

For the case of four positions in the first phase and two positions in the second phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \quad (4.17)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \quad (4.18)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \quad (4.19)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \quad (4.20)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \quad (4.21)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \quad (4.22)$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4 \quad (4.23)$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 5,6 \quad (4.24)$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in six equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Suppose six prescribed positions are shown in Figure 4.3. Plot the center point curve for positions 1, 2, 3, and 4 by means of program CENT_PT.PAS [12], and choose a center point S on it. Locate the circle points C_1, C_2, C_3, and C_4 by kinematic inversion and geometric similarity. This is done by calling a user-defined AutoLISP function INVERT. Invert point S from position 6 into position 5 to get point S_6. Draw a right bisector for line segment SS_6. Pick a point E_5 on the bisector as the circle point for position 5. Find E_6 by geometric similarity.

The first free choice of parameter in the above solution steps is P, the X coordinate of center point S, and the second free choice is R_2, the crank length for phase 2.

4.5 Positions 1234–45

For the case of four positions in the first phase and two positions in the second phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.25}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.26}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.27}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.28}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{4.29}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.30}$$

Equation (2.2) for phase 1 takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \\ Y_i &= b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3,4 \end{aligned} \tag{4.31}$$

and that for phase 2 is

$$\begin{aligned} X_i &= a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \\ Y_i &= b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5 \end{aligned} \tag{4.32}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in six equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

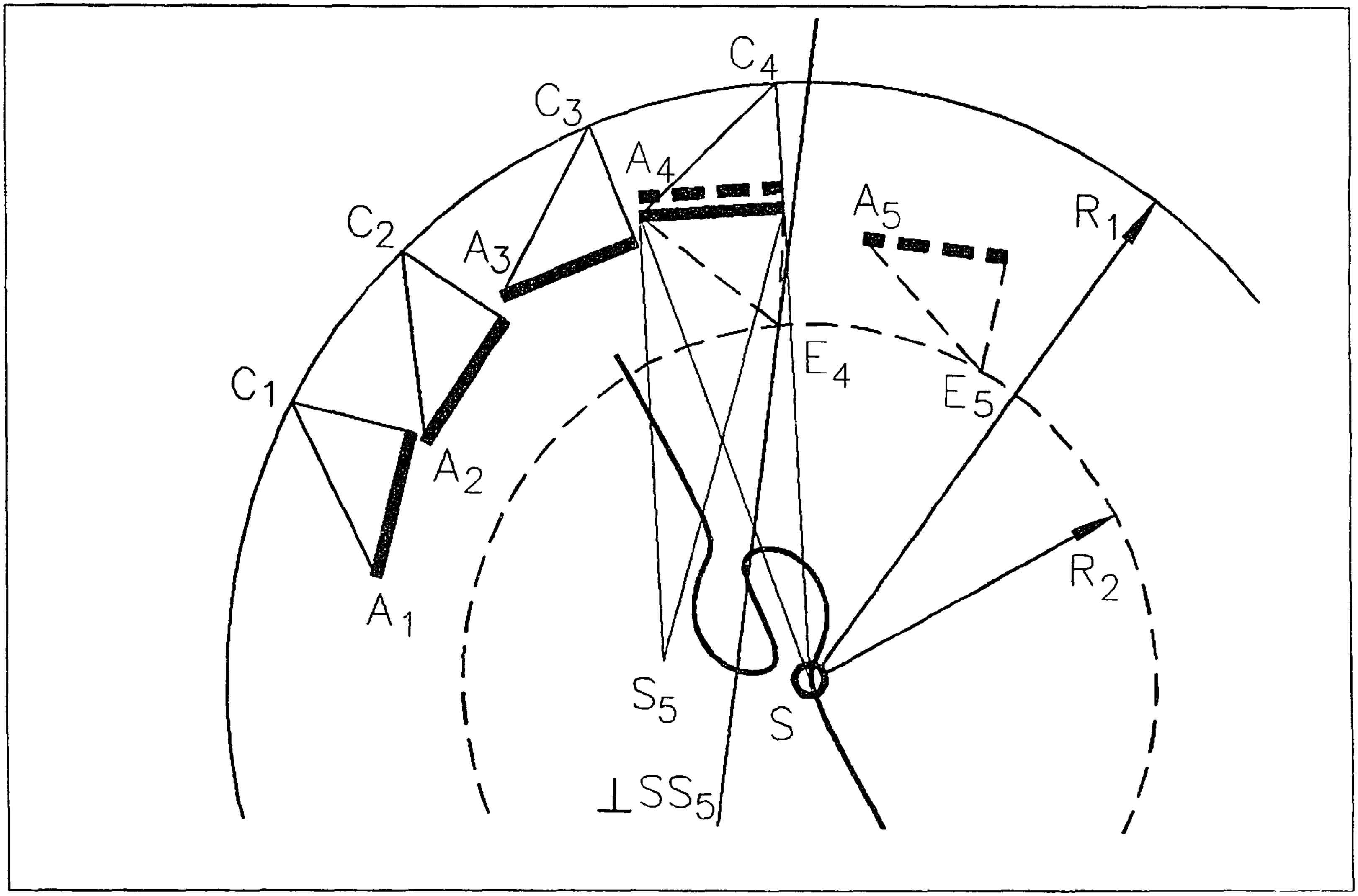

Figure 4.4 Adjustable moving pivot and crank length 1234–45

Five prescribed positions and a solution for this problem are shown in Figure 4.4. The solution steps for this problem are similar to that in the last section.

4.6 Positions 1234–567

For the case of four positions in the first phase and three positions in the second phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \qquad (4.33)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \qquad (4.34)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \qquad (4.35)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \qquad (4.36)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \qquad (4.37)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \qquad (4.38)$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \qquad (4.39)$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3,4 \qquad (4.40)$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 5,6,7 \qquad (4.41)$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in seven equations. Thus, the equations can be solved with one free choice of parameters, and have infinite solutions.

Suppose seven prescribed positions are shown in Figure 4.5. Plot the center point curve for positions 1, 2, 3, and 4 and choose a center point S on it. Locate the circle points C_1, C_2, C_3, and C_4 by kinematic inversion and geometric similarity. This is done by calling a user-defined AutoLISP function INVERT. Invert point S from positions 6 and 7 into position 5 to get points S_6 and S_7. Intersect right bisectors for line segments SS_6 and S_6S_7 at point E_5, which is the circle point at position 5. Circle points E_6 and E_7 can be found by geometric similarity.

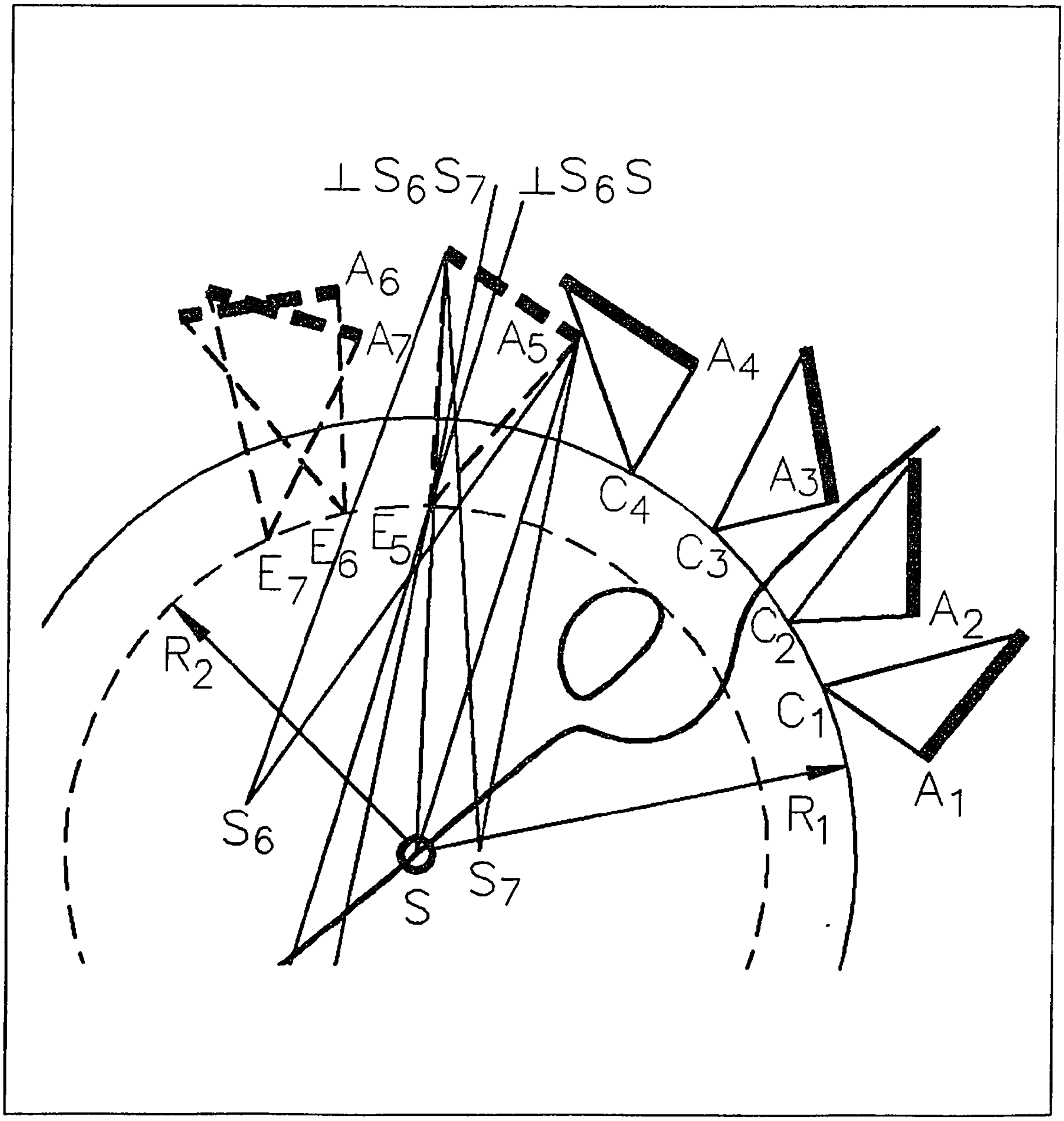

Figure 4.5 Adjustable moving pivot and crank length 1234–567

4.7 Positions 1234–456

For the case of four positions in the first phase and three positions in the second phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.42}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.43}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.44}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.45}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{4.46}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.47}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.48}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4 \tag{4.49}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 4,5,6 \tag{4.50}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in seven equations. Thus, the equations can be solved with one free choice of parameters, and have infinite solutions.

Suppose seven prescribed positions are shown in Figure 4.6. Plot the center point curve for positions 1, 2, 3, and 4 and choose a center point S on it. Locate the circle point C_1 by kinematic inversion. Find C_2, C_3, and C_4 by geometric similarity. Invert point S from positions 5 and 6 into position 4 to get points S_5 and S_6. Intersect right bisectors for line segments SS_5 and S_5S_6 at point E_4, which is the circle point at position 4 of phase 2. Circle points E_5 and E_6 can be found by geometric similarity.

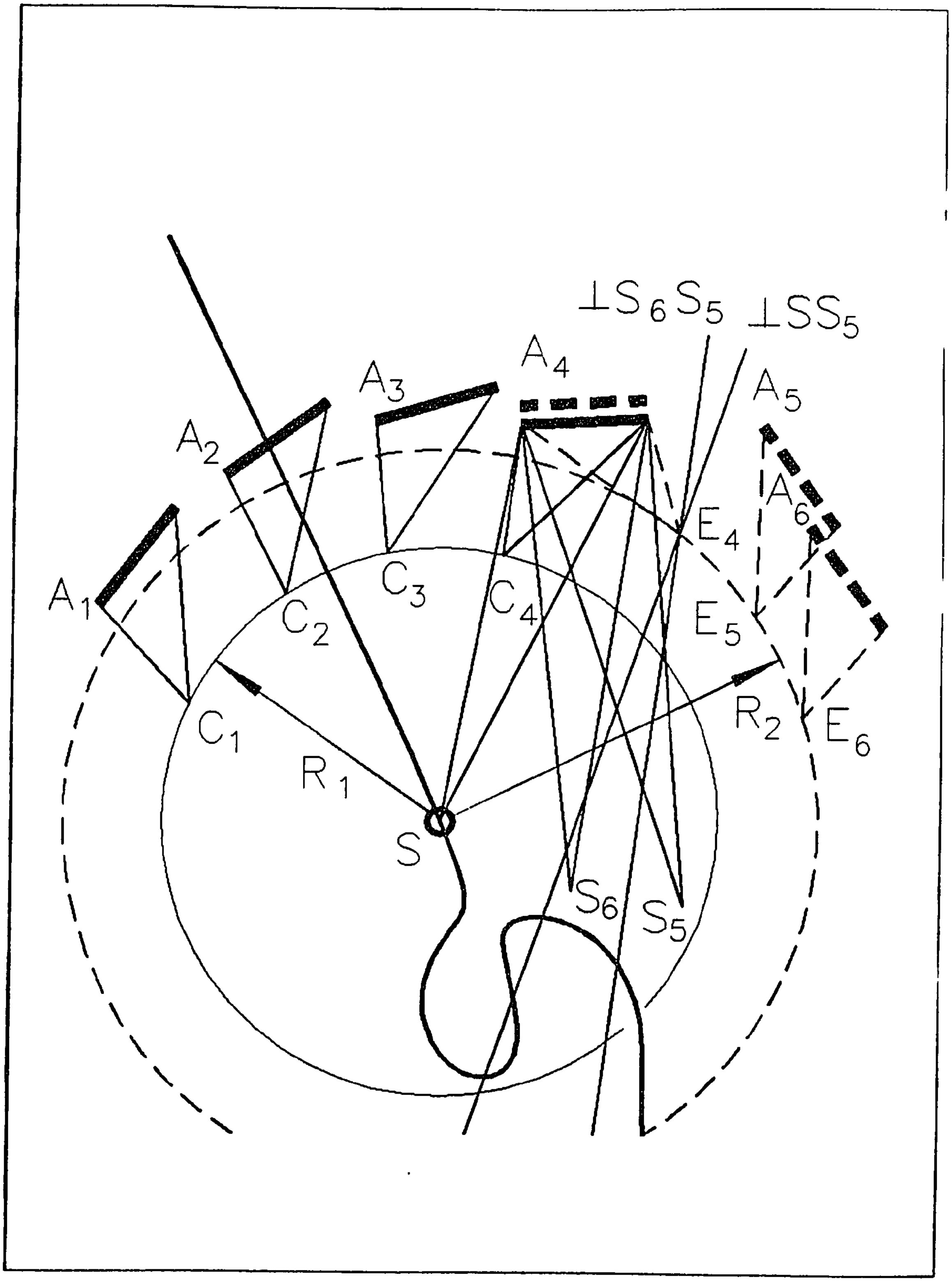

Figure 4.6 Adjustable moving pivot and crank length 1234–456

4.8 Positions 12345–67

For the case of five positions in the first phase and two positions in the second phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.51}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.52}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.53}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.54}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_1^2 \tag{4.55}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.56}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \tag{4.57}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$
$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4,5 \tag{4.58}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$
$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 6,7 \tag{4.59}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in seven equations. Thus, the equations have one free choice of parameter.

Suppose seven prescribed positions are shown in Figure 4.7. Plot two center point curves: one for positions 1, 2, 3, and 4, another one for positions 1, 2, 4, and 5 as shown in the figure. Try to pick a intersection point S of the two curves as the center point. Invert point S from positions 2, 3, 4, and 5 into position 1 by means of kinematic inversion to find point C_1, which is the circle point at position 1. Find circle points C_2, C_3, C_4, and C_5 by geometric similarity. As shown in the figure, no order defect occurs for the five positions of phase 1, which indicates that it is a good solution.

For the second phase, invert point S from position 7 into 6 to get point S_7. Plot the right bisector for the line segment SS_7. Any point on the bisector satisfies the basic equations of the problem. Now it is time to use the only free choice of parameter. Choose a crank length R_2 for phase 2. Draw a circle with center S and radius equal to R_2; this circle intersects the bisector at point E_6, which is the circle point at position 6 of phase 2. E_7 can be found by geometric similarity.

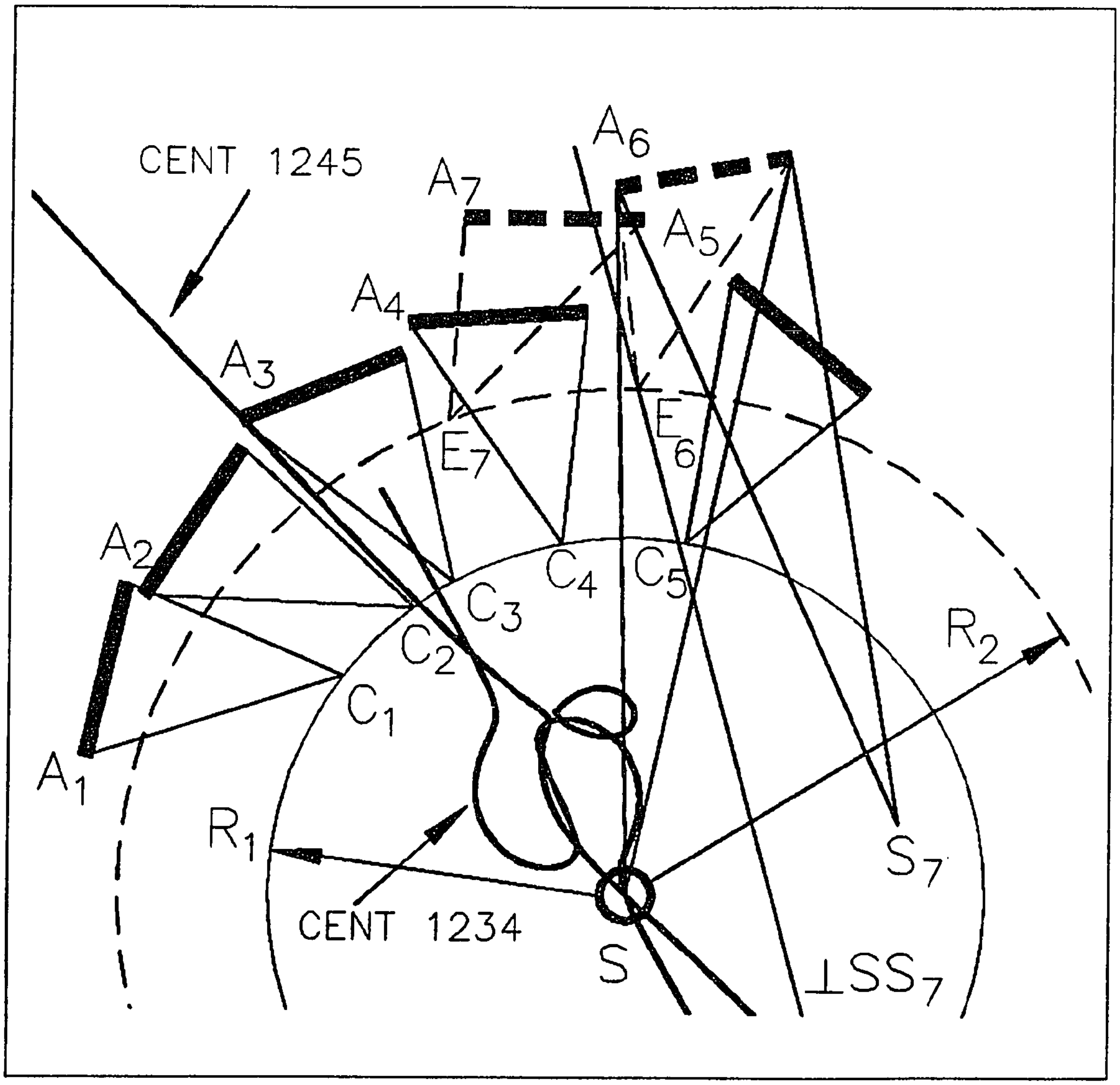

Figure 4.7 Adjustable moving pivot and crank length 12345–67

There are four different combinations of the position numbers in which the center point curves can be plotted. They are 1234, 1235, 1245, and 1345. Note that position number 1 should be included, because plane number 1 is our working plane, everything is inverted into position 1 and the circle point at position 1 is our goal of this solution step.

4.9 Positions 12345–56

For the case of five positions in the first phase and two positions in the second phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.60}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.61}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.62}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.63}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_1^2 \tag{4.64}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.65}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.66}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4,5 \tag{4.67}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 5,6 \tag{4.68}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in seven equations. Thus, the equations have one free choice of parameter.

Suppose six prescribed positions are shown in Figure 4.8. Since the five prescribed positions of phase 1 are the same as that for the example in the last section, the work for phase 1 is the same as that in Figure 4.7.

The work for the second phase is similar to that for the last example. Invert center point S from position 6 into position 5 to get S_6. Draw a right bisector for line segment SS_6. Pick a circle point E_5 for position 5 of phase 2 on the bisector. Find E_6 by geometric similarity.

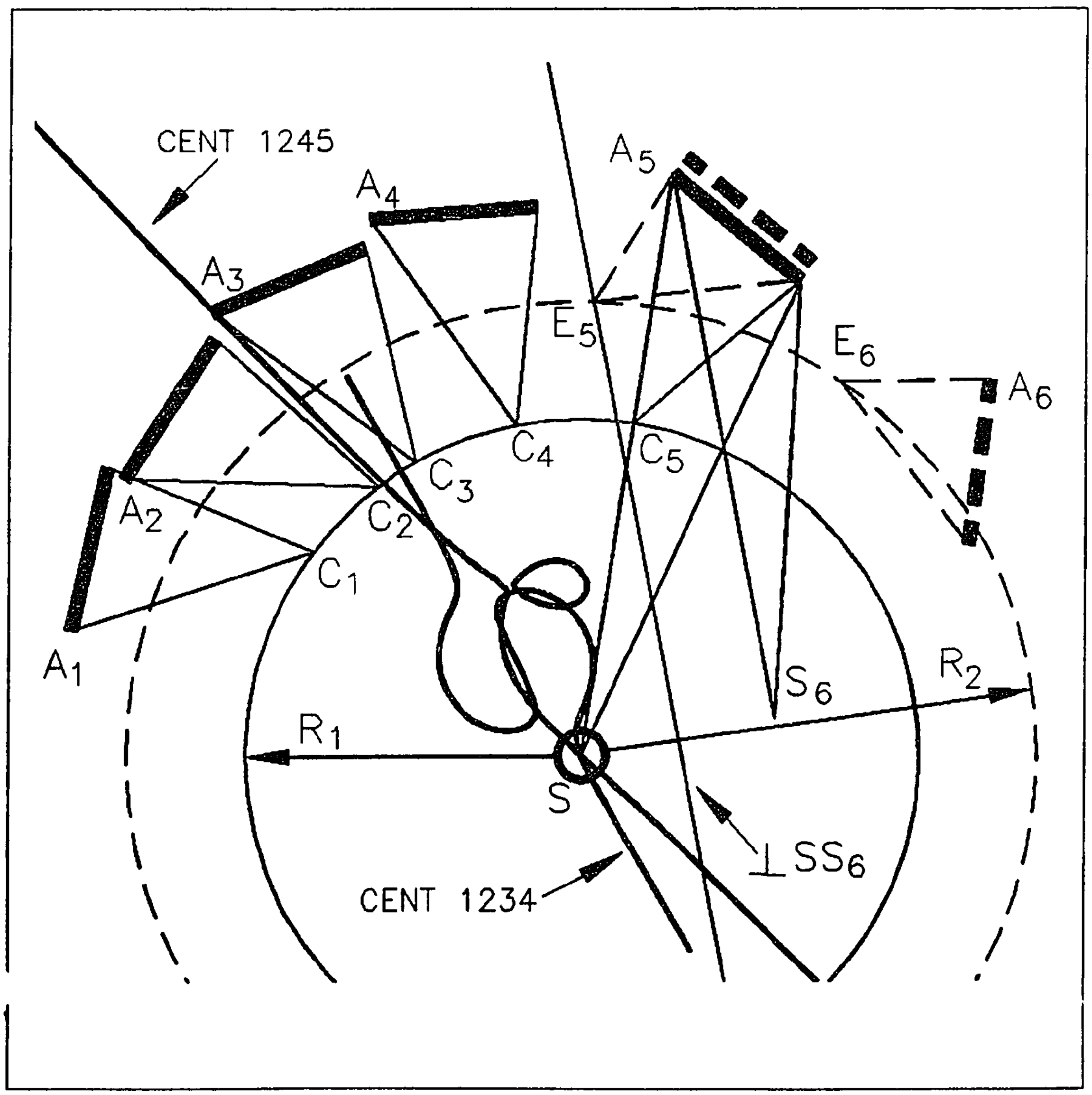

Figure 4.8 Adjustable moving pivot and crank length 12345–56

4.10 Positions 1234–5678

For the case of four positions in each of the two phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.69}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.70}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.71}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.72}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.73}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.74}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \tag{4.75}$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R_2^2 \tag{4.76}$$

Equation (2.2) for phase 1 takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos\theta_i - q_1 \sin\theta_i \\ Y_i &= b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4 \end{aligned} \tag{4.77}$$

and that for phase 2 is

$$\begin{aligned} X_i &= a_i + p_2 \cos\theta_i - q_2 \sin\theta_i \\ Y_i &= b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 5,6,7,8 \end{aligned} \tag{4.78}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in eight equations. There is no free choice of parameter, and the number of positions have reached the maximum value.

Suppose eight prescribed positions are shown in Figure 4.9. Plot two center point curves: one for positions 1, 2, 3, and 4, another one for positions 5, 6, 7, and 8. Pick a center point S at the intersection point of the two curves. Invert S for phase 1 to get circle point C_1, and for phase 2 to get circle point E_5. Find circle points C_2, C_3, C_4, E_6, E_7, and E_8 by geometric similarity. A good solution is found in the figure since no order defect occurs.

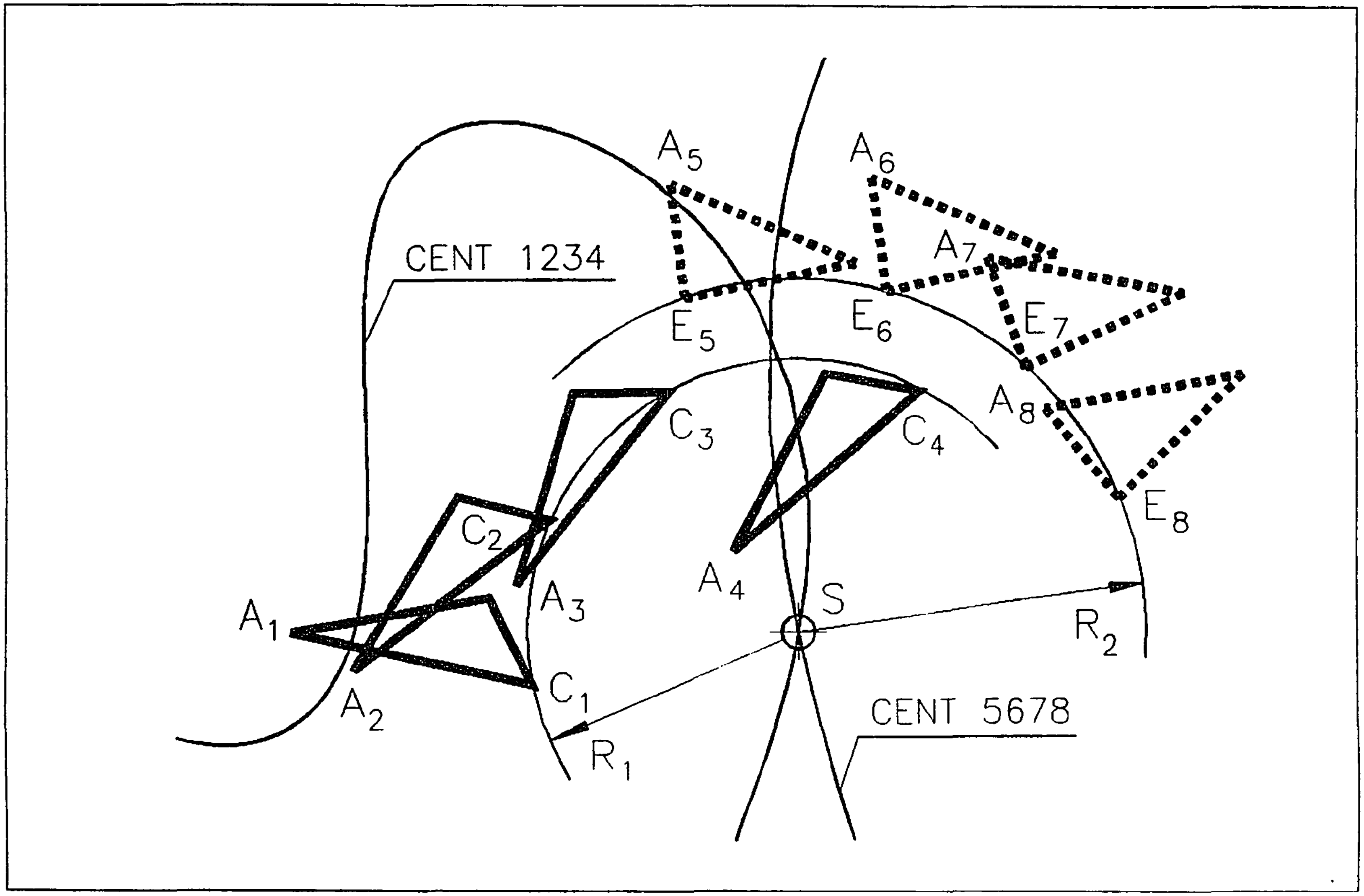

Figure 4.9 Adjustable moving pivot and crank length 1234–5678

4.11 Positions 1234–4567

For the case of four positions in each of the two phases with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.79}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.80}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.81}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.82}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{4.83}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{4.84}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.85}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \tag{4.86}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3,4 \tag{4.87}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5,6,7 \tag{4.88}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in eight equations. There is no free choice of parameter, and the number of positions reaches the maximum value.

Suppose eight prescribed positions are shown in Figure 4.10. Plot two center point curves: one for positions 1, 2, 3, and 4, another one for positions 4, 5, 6, and 7. Pick center point S at the intersection point of the two curves. Invert S for phase 1 to get circle point C_1, and for phase 2 to get circle point E_4. Find circle points C_2, C_3, C_4, E_5, E_6, and E_7 by geometric similarity. A good solution is found in the figure since no order defect occurs.

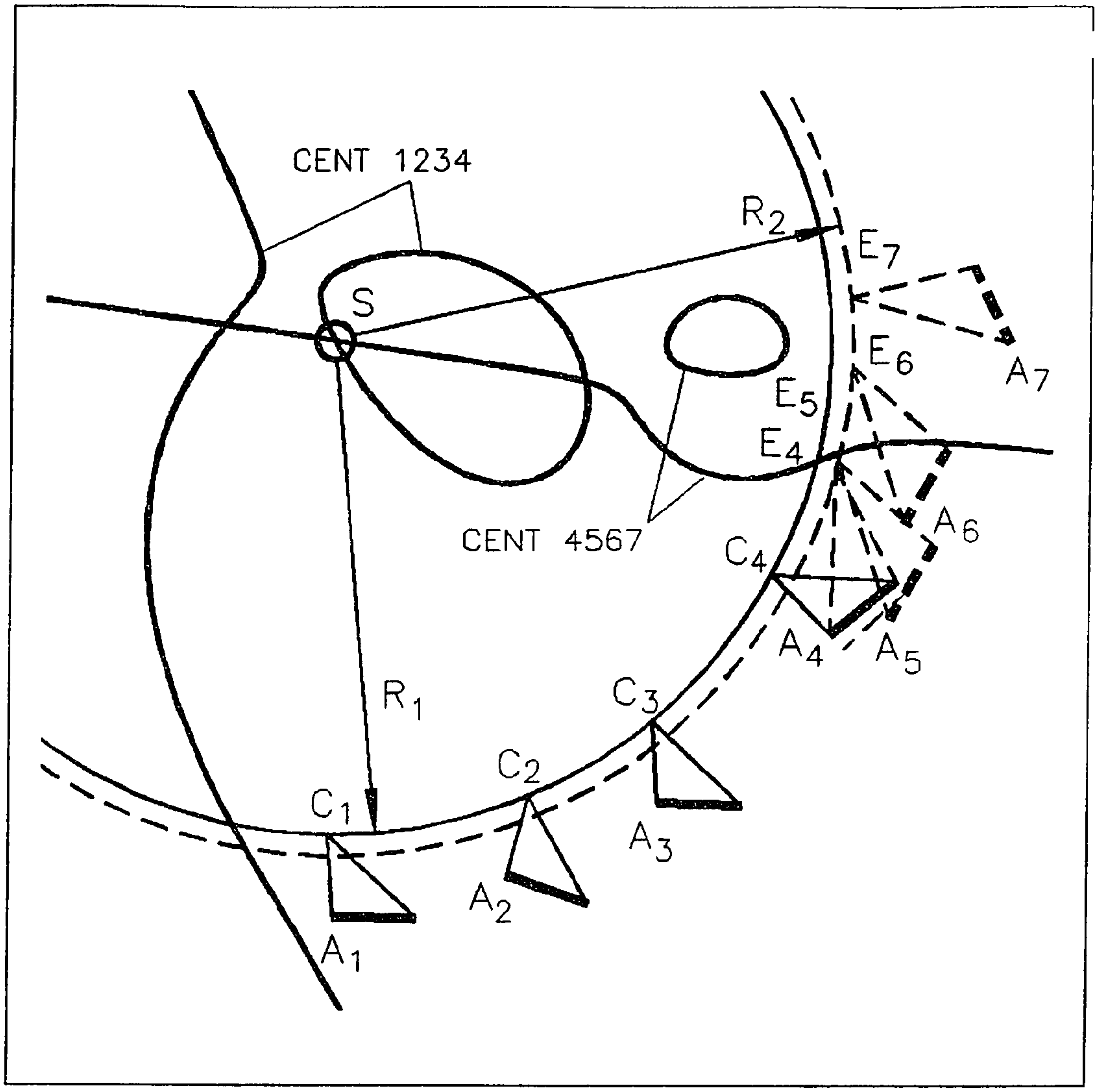

Figure 4.10 Adjustable moving pivot and crank length 1234-4567

4.12 Positions 12345–678

For the case of five positions in the first phase and three positions in the second phase with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{4.89}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{4.90}$$

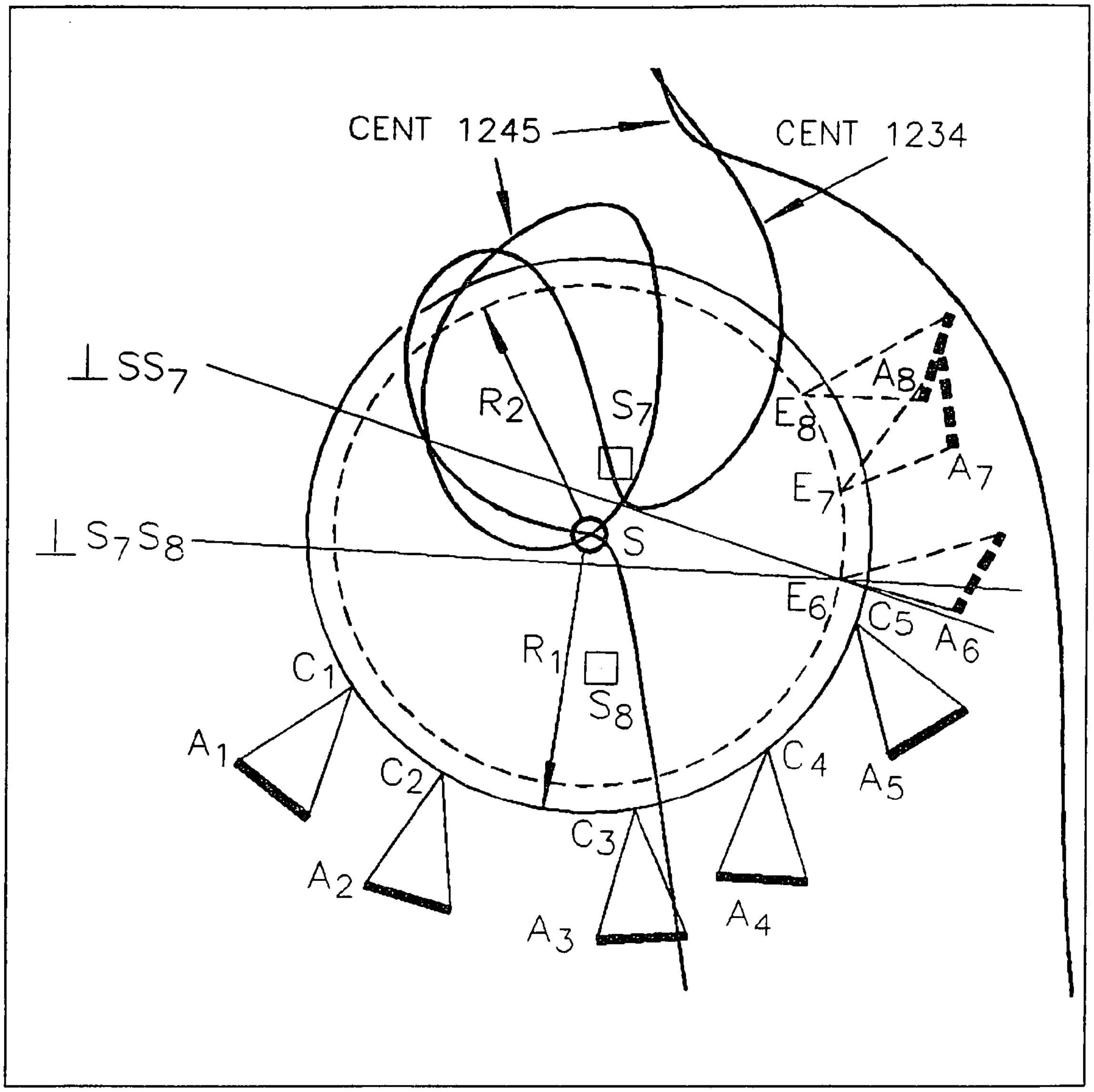

Figure 4.11 Adjustable moving pivot and crank length 12345–678

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{4.91}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \tag{4.92}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_1^2 \tag{4.93}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \tag{4.94}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \tag{4.95}$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R_2^2 \tag{4.96}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$
$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4,5 \quad (4.97)$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$
$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 6,7,8 \quad (4.98)$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in eight equations. There is no free choice of parameter. The number of positions reaches the maximum value, the number of positions in phase 1 also reaches its maximum value.

Suppose eight prescribed positions are shown in Figure 4.11. Plot two center point curves: one for positions 1, 2, 3, and 4, another one for positions 1, 2, 4, and 5. Pick center point S at the intersection point of the two curves. Invert S for phase 1 to get circle point C_1, and for phase 2 to get circle point E_6. Find circle points C_2, C_3, C_4, C_5, E_7, and E_8 by geometric similarity. A good solution is shown in the figure since no order defect occurs.

4.13 Positions 12345–567

For the case of five positions in the first phase and three positions in the second phase with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \quad (4.99)$$
$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \quad (4.100)$$
$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \quad (4.101)$$
$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_1^2 \quad (4.102)$$
$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_1^2 \quad (4.103)$$
$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \quad (4.104)$$
$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_2^2 \quad (4.105)$$

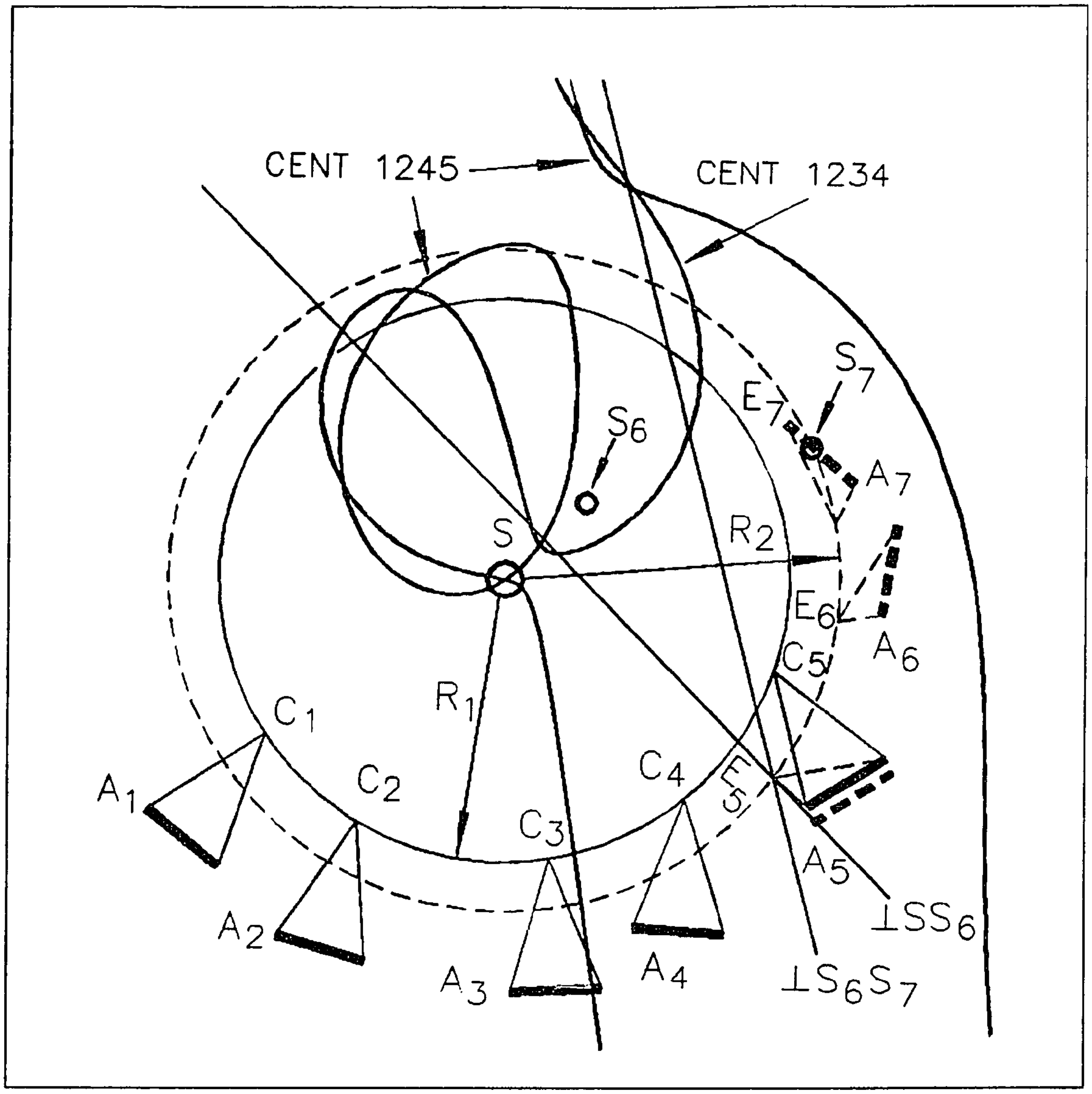

Figure 4.12 Adjustable moving pivot and crank length 12345–567

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_2^2 \tag{4.106}$$

Equation (2.2) for phase 1 takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4,5 \tag{4.107}$$

and that for phase 2 is

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 5,6,7 \tag{4.108}$$

Eight parameters, P, Q, p_1, q_1, p_2, q_2, R_1, and R_2, are involved in eight equations. There is no free choice of parameter. The number of positions reaches the maximum value, the number of positions in phase 1 also reaches its maximum value.

Suppose eight prescribed positions are shown in Figure 4.12. Plot two center point curves: one for positions 1, 2, 3, and 4, another one for positions 1, 2, 4, and 5. Pick center point S at the intersection point of the two curves. Invert S for phase 1 to get circle point C_1, and for phase 2 to get circle point E_5. Find circle points C_2, C_3, C_4, C_5, E_6 and E_7 by geometric similarity. A good solution is shown in the figure since no order defect occurs.

Chapter 5

Three Phase Adjustable Moving Pivot Problems

5.1 Introduction

Chapters 3 and 4 dealt with two phase problems. Three phase problems are discussed in chapters 5, 6, and 7. This chapter deals with the problem of three phase adjustable moving pivot. Nine parameters need to be determined for this group of problems, which are P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R. Thus, the maximum prescribed positions would be nine.

Table 5.1 Three phase adjustable moving pivot problems

ph.1	positions ph.2	ph.3	shared pos.	unknowns	free choices
1,2	3,4	5,6	0	6	3
1,2	3,4	4,5	1	6	3
1,2,3	4,5	6,7	0	7	2
1,2,3	4,5	5,6	1	7	2
1,2,3	4,5,6	7,8	0	8	1
1,2,3	4,5,6	6,7	1	8	1
1,2,3	3,4,5	6,7	1	8	1
1,2,3	3,4,5	5,6	2	8	1
1,2,3	4,5,6	7,8,9	0	9	0
1,2,3	3,4,5	6,7,8	1	9	0
1,2,3,4	5,6,7	8,9	0	9	0

Eleven problems listed in Table 5.1 are solved in this chapter. The minimum number of prescribed positions included in one phase is two, and the maximum number is four. The maximum total number of positions is nine, which is the maximum possible value. Some problems with shared positions are also included in the table.

The method for solving three phase adjustable moving pivot problems is based on the method for two phase problems in chapter 3. In other words, the method in chapter 3 could be extended for solving three phase problems of the same kind.

5.2 Positions 12–34–56

For the case of two positions in each of the three phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.1}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.2}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.3}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.4}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.5}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.6}$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2 \tag{5.7}$$

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 3,4 \tag{5.8}$$

$$X_i = a_i + p_3 \cos \theta_i - q_3 \sin \theta_i$$

$$Y_i = b_i + p_3 \sin \theta_i + q_3 \cos \theta_i \qquad i = 5,6 \tag{5.9}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in six equations. Thus, the equations can be solved with three free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Suppose six prescribed positions are shown in Figure 5.1, and the relative coordinates of the circle point C_1 are chosen as two free choices. Find C_2 by geometric similarity after locating C_1. Draw a circle with center C_1 and a chosen radius R; this circle intersects the right bisector for line segment C_1C_2 at S, which is the center point. Invert point S from position 4 into position 3 to get point S_4. Draw a circle passing through points C_1 and C_2 with center S; this circle intersects the right bisector for the line segment SS_4 at D_3, which is the circle point at position 3 of phase 2. Find D_4 by geometric similarity.

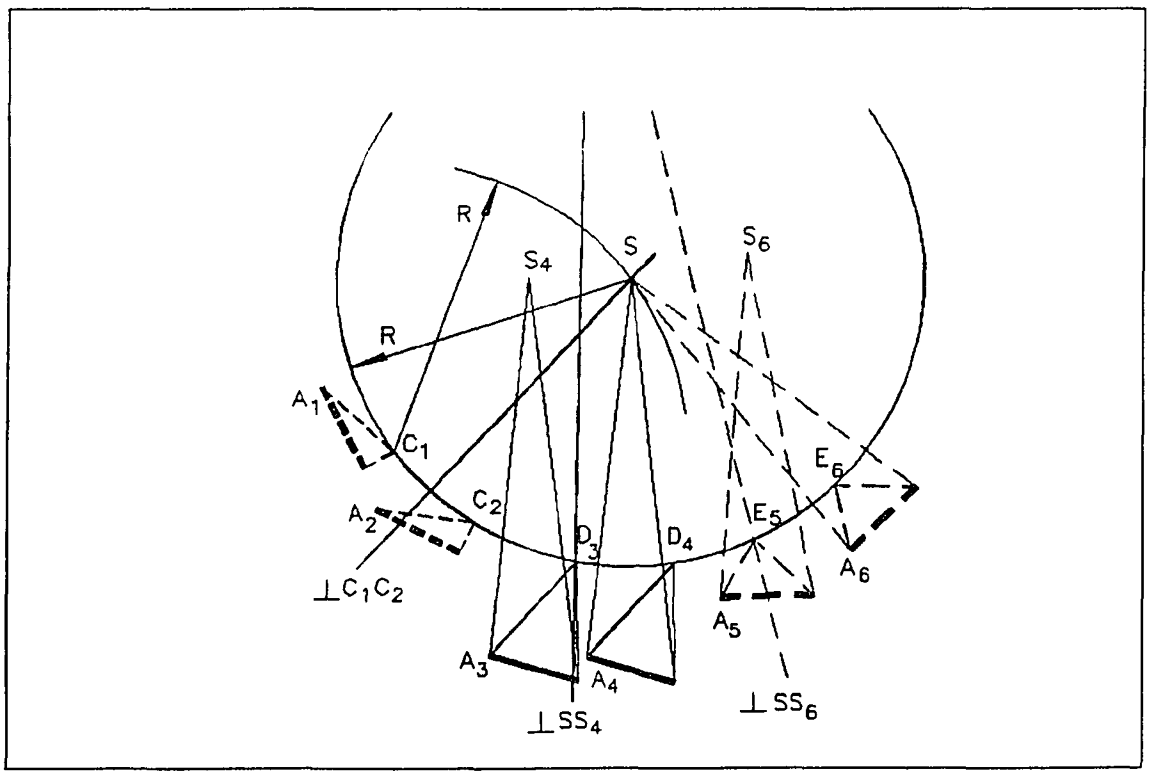

Figure 5.1 Adjustable moving pivot 12–34–56

Similarly, invert point S from position 6 into position 5 to get point S_6. Draw a right bisector for line segment SS_6; this bisector intersects the circle at point E_5, which is the circle point at position 5 of phase 3. Finally, E_6 can be found by geometric similarity.

5.3 Positions 12–34–45

For the case of two positions in each of the three phases with one position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \quad (5.10)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \quad (5.11)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \quad (5.12)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \quad (5.13)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \quad (5.14)$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2 \quad (5.15)$$

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 3,4 \quad (5.16)$$

$$X_i = a_i + p_3 \cos\theta_i - q_3 \sin\theta_i$$

$$Y_i = b_i + p_3 \sin\theta_i + q_3 \cos\theta_i \qquad i = 4,5 \quad (5.17)$$

The method in the last section could also be used to solve this problem. The position could be shared either by phases 1 and 2, or by phases 2 and 3. Figure 5.2 represents an example for the latter case.

In this example, position 4 is shared by phases 2 and 3. The solution steps for this problem are basically the same as that in the case of no shared position except the last few steps for phase 3. As shown in the figure, invert center point S from position 5 into position 4 to get point S_5. Plot the right bisector for line segment SS_5; this bisector intersects the circle passing through points C_1 and C_2 at E_4, which is the circle point at position 4 of phase 3.

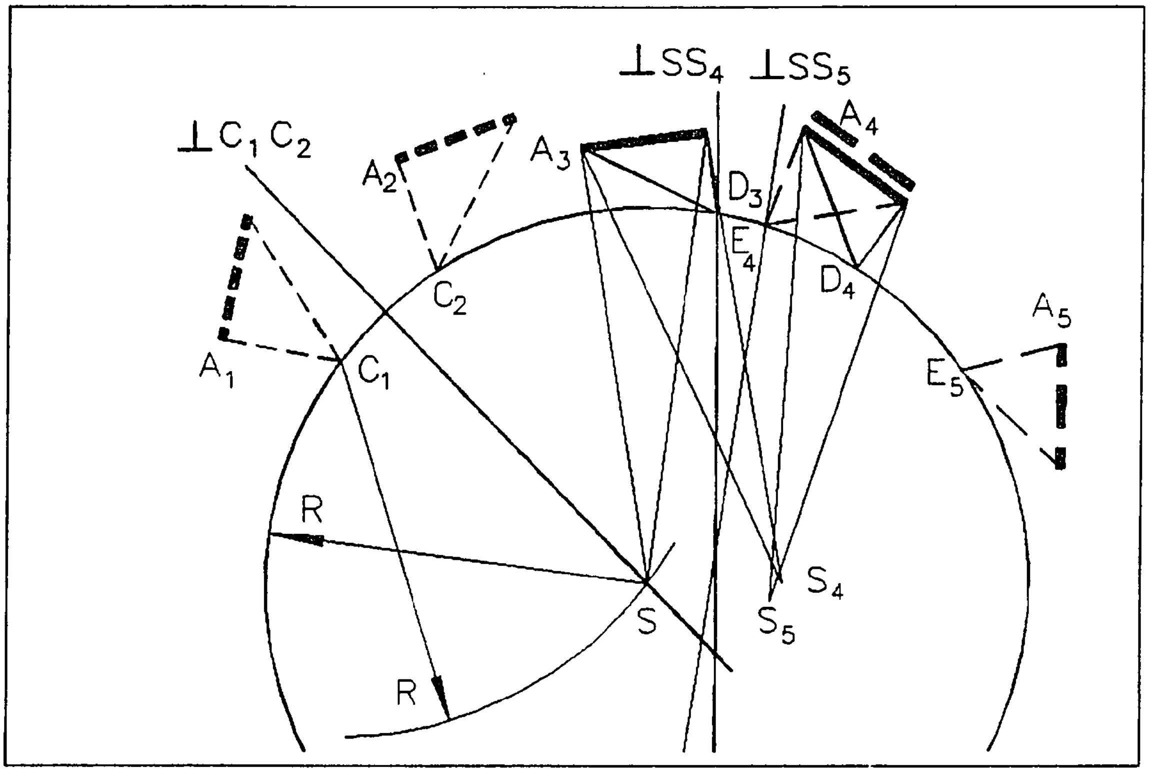

Figure 5.2 Adjustable moving pivot 12–34–45

5.4 Positions 123–45–67

This problem needs three positions in phase 1 and two positions in both phases 2 and 3 with no position shared. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \quad (5.18)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \quad (5.19)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \quad (5.20)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \quad (5.21)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \quad (5.22)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.23}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.24}$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \tag{5.25}$$

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 4,5 \tag{5.26}$$

$$X_i = a_i + p_3 \cos\theta_i - q_3 \sin\theta_i$$

$$Y_i = b_i + p_3 \sin\theta_i + q_3 \cos\theta_i \qquad i = 6,7 \tag{5.27}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in seven equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Suppose seven prescribed positions are shown in Figure 5.3, and the relative coordinates of the circle point C_1 are chosen as the two free choices. Find C_2 and C_3 by geometric similarity after locating C_1. Construct two right bisectors for line segments C_2C_3 and C_1C_2 to get their intersection point S, which is the center point. Construct a circle with center S passing through circle points C_1, C_2, and C_3. The radius of the circle is the crank length R.

Invert S from position 5 into position 4 to get point S_5. Construct a right bisector for line segment SS_5; this bisector intersects the circle at D_4, which is the circle point at position 4 of phase 2.

Similarly, Invert S from position 7 into position 6 to get point S_7. Construct a right bisector for line segment SS_7; this bisector intersects the circle at D_6, which is the circle point at position 6 of phase 3.

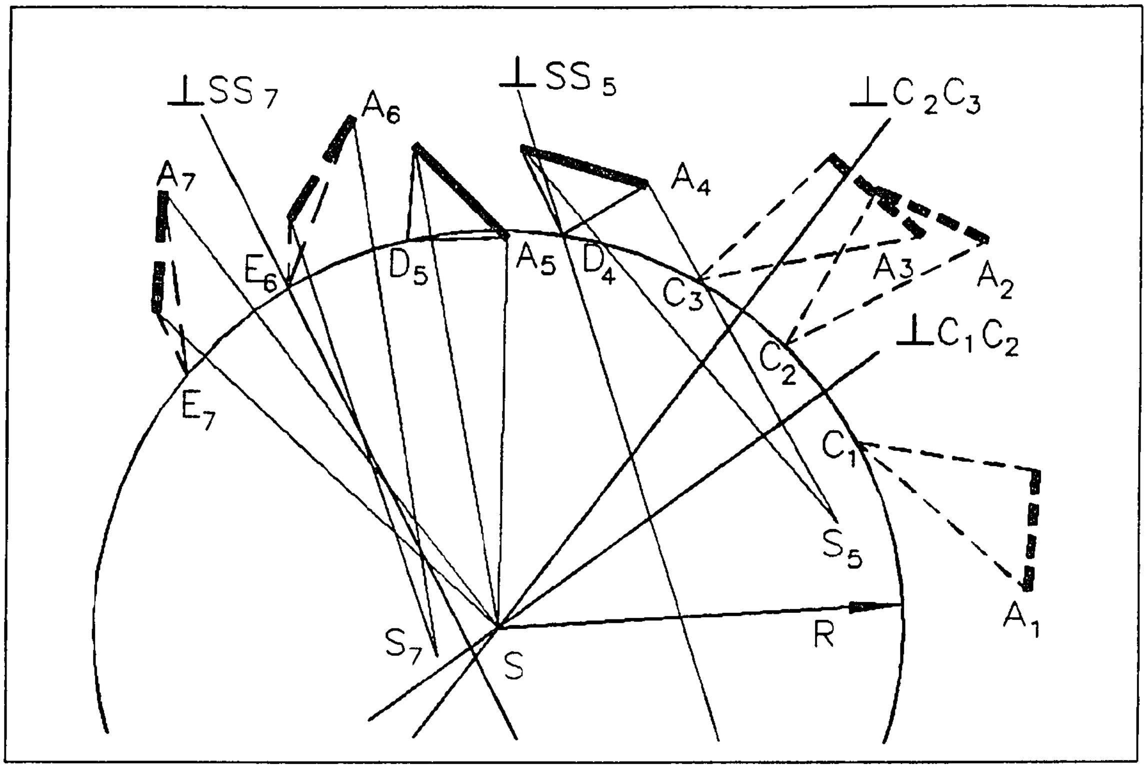

Figure 5.3 Adjustable moving pivot 123–45–67

5.5 Positions 123–45–56

This problem needs three positions in phase 1 and two positions in both phases 2 and 3 with one position shared by phases 2 and 3. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.28}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.29}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.30}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.31}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.32}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.33}$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \tag{5.34}$$

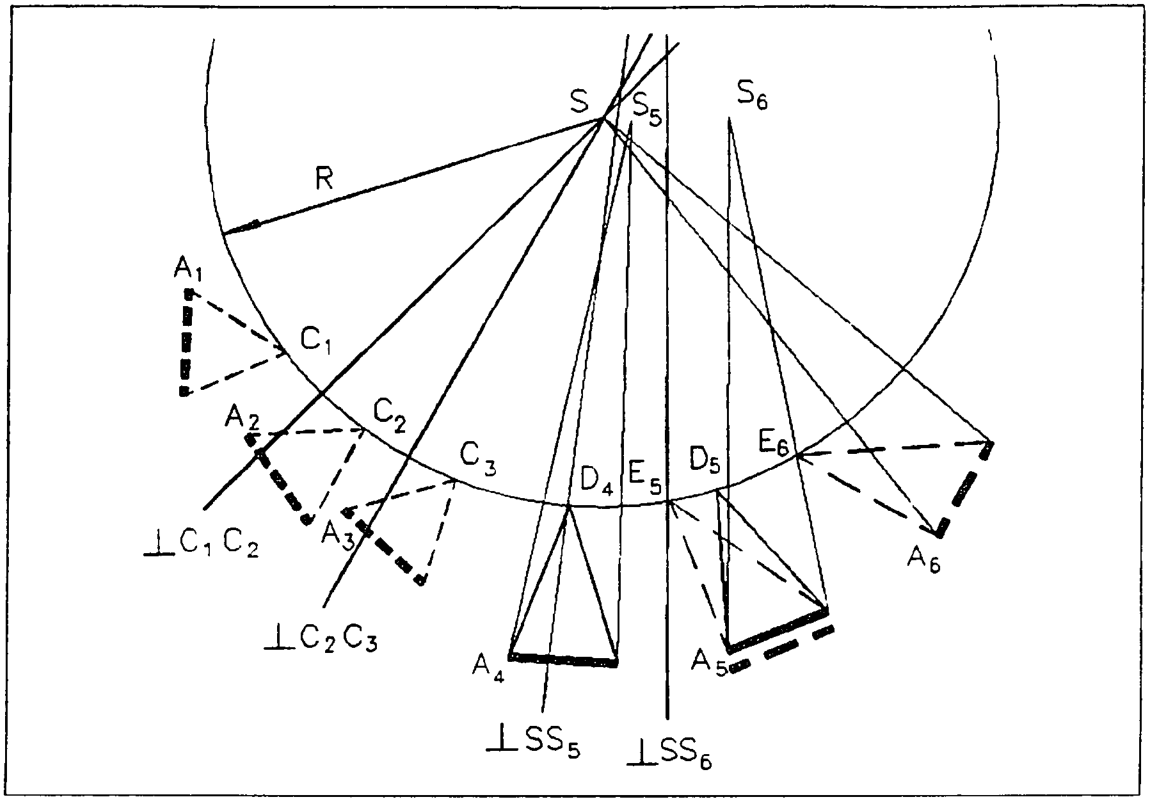

Figure 5.4 Adjustable moving pivot 123–45–56

$$X_i = a_i + p_2 \cos\theta_i - q_2 \sin\theta_i$$

$$Y_i = b_i + p_2 \sin\theta_i + q_2 \cos\theta \qquad i = 4,5 \quad (5.35)$$

$$X_i = a_i + p_3 \cos\theta_i - q_3 \sin\theta_i$$

$$Y_i = b_i + p_3 \sin\theta_i + q_3 \cos\theta \qquad i = 5,6 \quad (5.36)$$

In the example of Figure 5.4, position 5 is shared by phases 2 and 3. The solution steps for this problem are basically the same as that in the case of last section except the last few steps for phase 3. As shown in the figure, invert center point S from position 6 into position 5 to get point S_6. Plot a right bisector for line segment SS_6; this bisector intersects the circle passing through points C_1, C_2, and C_3 at E_5, which is the circle point at position 5 of phase 3.

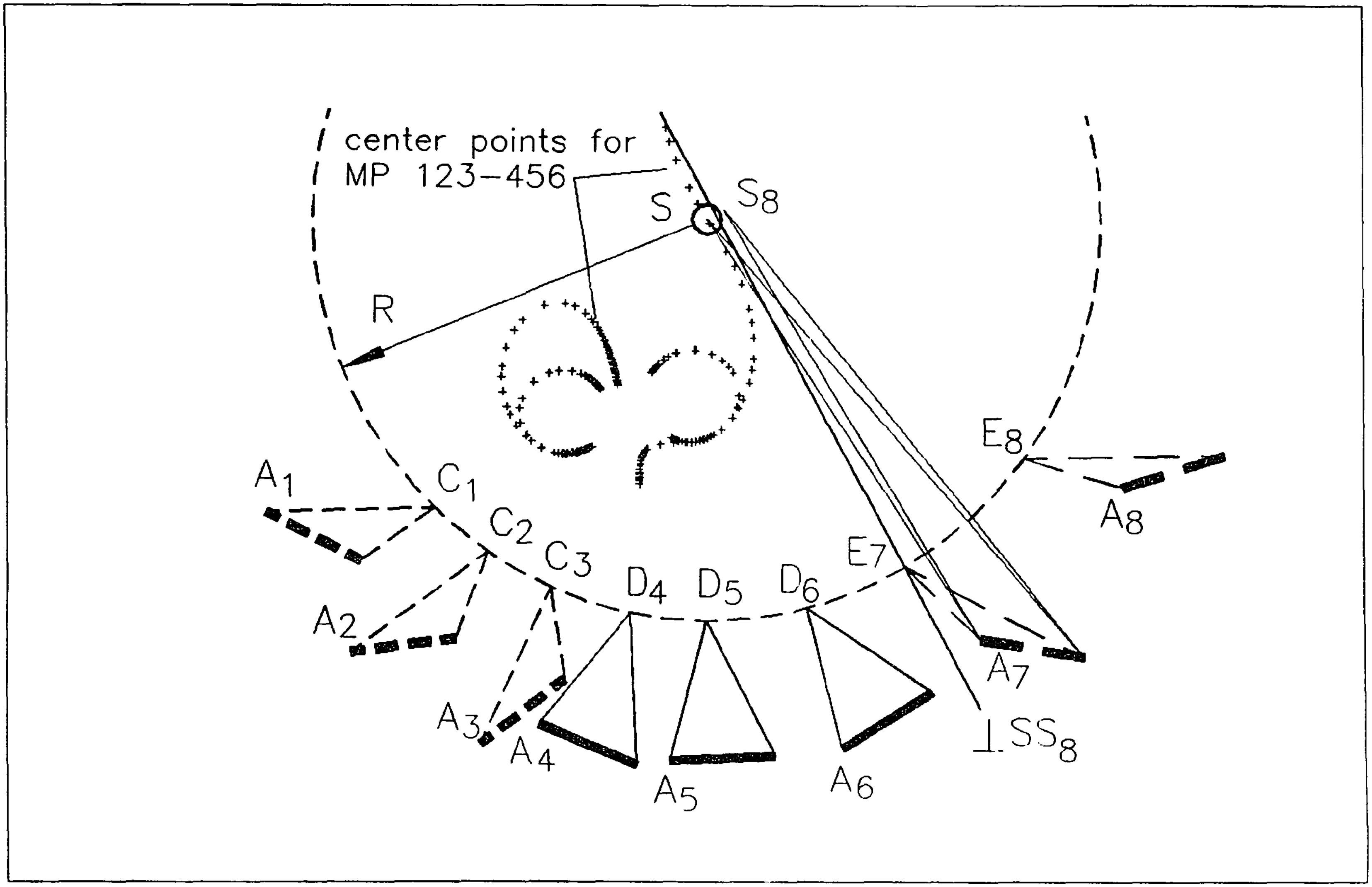

Figure 5.5 Adjustable moving pivot 123–456–78

5.6 Positions 123–456–78

This problem needs three positions in both phases 1 and 2, and two positions in phase 3 with no position shared. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.37}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.38}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.39}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.40}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.41}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.42}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.43}$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R^2 \tag{5.44}$$

Equation (2.2) takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos\theta_i - q_1 \sin\theta_i \\ Y_i &= b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \end{aligned} \tag{5.45}$$

$$\begin{aligned} X_i &= a_i + p_2 \cos\theta_i - q_2 \sin\theta_i \\ Y_i &= b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 4,5,6 \end{aligned} \tag{5.46}$$

$$\begin{aligned} X_i &= a_i + p_3 \cos\theta_i - q_3 \sin\theta_i \\ Y_i &= b_i + p_3 \sin\theta_i + q_3 \cos\theta_i \qquad i = 7,8 \end{aligned} \tag{5.47}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Any solution for the two phase adjustable moving pivot problem MP 123–456, which has been solved in chapter 3 is a solution for the problem 123–456–78.

Suppose eight prescribed positions and the center points for the two phase problem MP 123–456 are shown in Figure 5.5. A good center point S is picked on the curve, and the circle points C_1, C_2, C_3, D_4, D_5, and D_6 are found as shown.

The next goal is to find a circle point for positions 7 and 8, so that the center point S and the crank length R remain the same as that for phases 1 and 2. That is, the circle points E_7 and E_8 should lie on the circle passing through circle points C_1, C_2, C_3, D_4, D_5, and D_6.

Invert center point S from position 8 into position 7 to get point S_8. Plot a right bisector for line segment SS_8; this bisector intersects the circle at point E_7, which is the circle point at position 7 of phase 3. Finally, E_8 can be found by geometric similarity.

5.7 Positions 123–456–67

This problem needs three positions in both phases 1 and 2, and two positions in phase 3 with one position shared by phases 2 and 3. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.48}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.49}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.50}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.51}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.52}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.53}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.54}$$

Equation (2.2) for phase 1 takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos\theta_i - q_1 \sin\theta_i \\ Y_i &= b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \end{aligned} \tag{5.55}$$

$$\begin{aligned} X_i &= a_i + p_2 \cos\theta_i - q_2 \sin\theta_i \\ Y_i &= b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 4,5,6 \end{aligned} \tag{5.56}$$

..

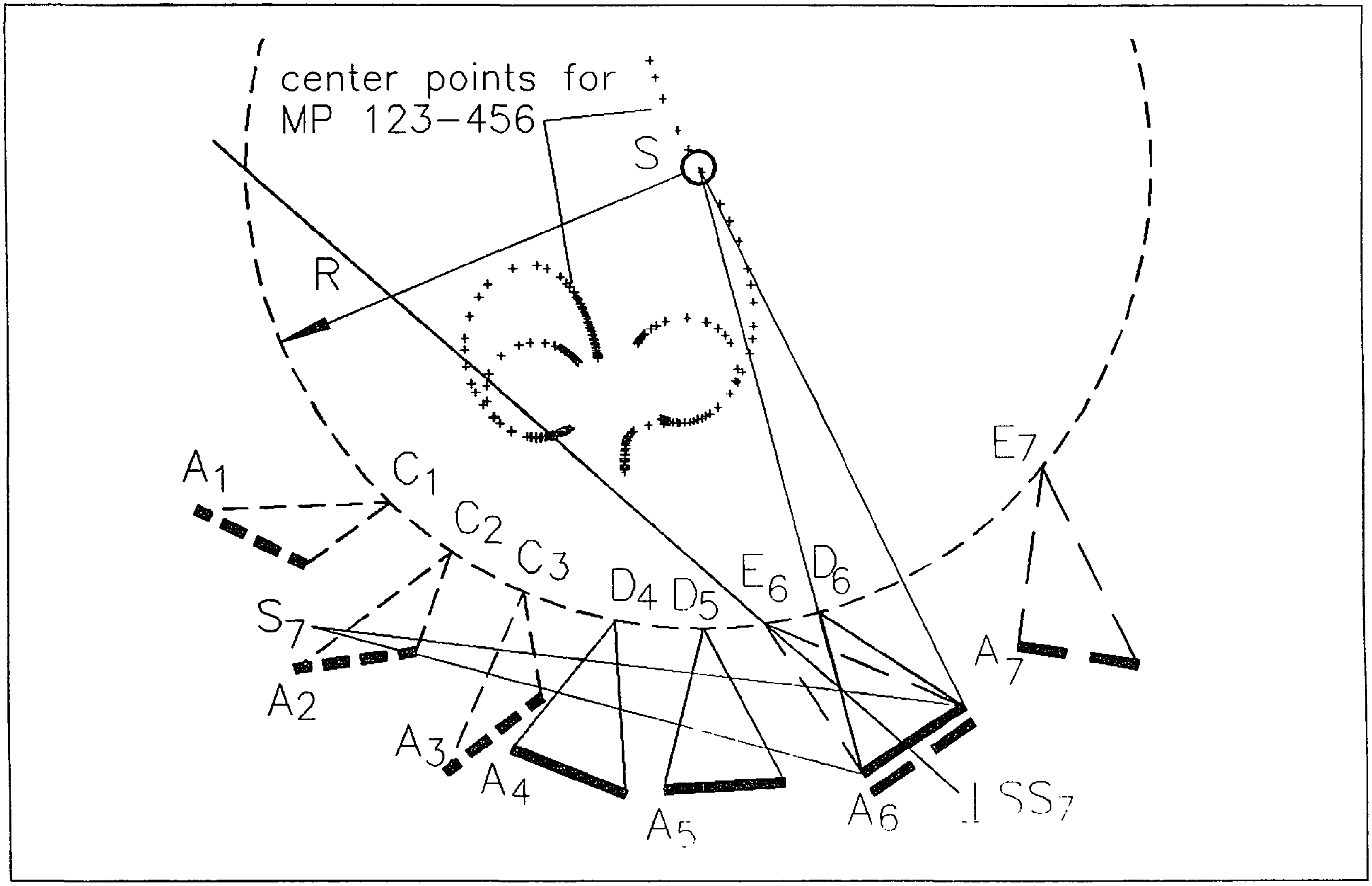

Figure 5.6 Adjustable moving pivot 123–456–67

$$X_i = a_i + p_3 \cos\theta_i - q_3 \sin\theta_i$$
$$Y_i = b_i + p_3 \sin\theta_i + q_3 \cos\theta_i \qquad i = 6,7 \qquad (5.57)$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Any solution for the two phase adjustable moving pivot problem MP 123–456, which has been solved in chapter 3 is a solution for the problem MP 123–456–67. The solution steps are similar to that for the problem in the last section.

Suppose eight prescribed positions and the center points for the two phase problem MP 123–456 are shown in Figure 5.6. A good center point S is picked on the curve, and the circle points C_1, C_2, C_3, D_4, D_5, and D_6 are found as shown.

The next goal is to find a circle point for positions 6 and 7, so that the center point S and the crank length R remain the same as that for phases 1 and 2. That is, the circle points E_6 and E_7 should lie on the circle passing through circle points C_1, C_2, C_3, D_4, D_5, and D_6.

Invert center point S from position 7 into position 6 to get point S_7. Plot a right bisector for line segment SS_7; this bisector intersects the circle at point E_6, which is the circle point at position 6 of phase 3. Finally, E_7 can be found by geometric similarity.

5.8 Positions 123–345–67

This problem needs three positions in both phases 1 and 2, and two positions in phase 3 with one position shared by phases 1 and 2. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \qquad (5.58)$$

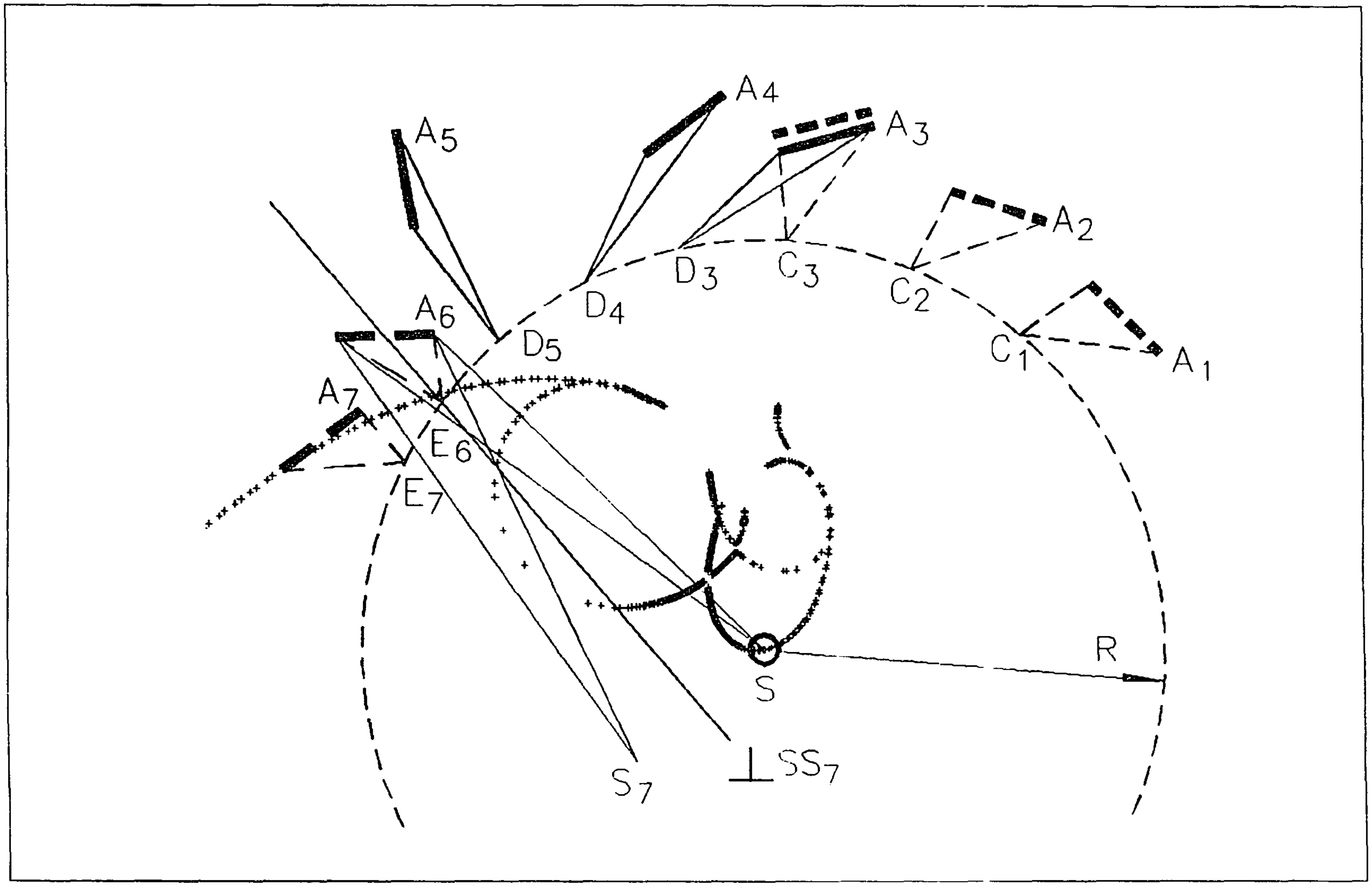

Figure 5.7 Adjustable moving pivot 123–345–67

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.59}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.60}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.61}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.62}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.63}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.64}$$

Equation (2.2) takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \\ Y_i &= b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \end{aligned} \tag{5.65}$$

$$\begin{aligned} X_i &= a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \\ Y_i &= b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 3,4,5 \end{aligned} \tag{5.66}$$

$$\begin{aligned} X_i &= a_i + p_3 \cos \theta_i - q_3 \sin \theta_i \\ Y_i &= b_i + p_3 \sin \theta_i + q_3 \cos \theta_i \qquad i = 6,7 \end{aligned} \tag{5.67}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Any solution for the two phase adjustable moving pivot problem MP 123–345, which has been solved in chapter 3 is a solution for the problem MP 123–345–67. The solution steps are similar to that for the problem in the last section.

Suppose seven prescribed positions and the center points for the two phase problem MP 123–345 are shown in Figure 5.7. A good center point S is picked on the curve, and the circle points C_1, C_2, C_3, D_3, D_4, and D_5 are found as shown.

The next goal is to find a circle point for positions 6 and 7, so that the center point S and the crank length R remain the same as that for phases 1 and 2. That is, the circle points E_6 and E_7 should lie on the circle passing through circle points C_1, C_2, C_3, D_3, D_4, and D_5.

Invert center point S from position 7 into position 6 to get point S_7. Plot a right bisector for line segment SS_7; this bisector intersects the circle at point E_6, which is the circle point at position 6 of phase 3. Finally, E_7 can be found by geometric similarity.

5.9 Positions 123–345–56

This problem needs three positions in both phases 1 and 2, and two positions in phase 3 with one position shared by phases 1 and 2, and another position shared by phases 2 and 3. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.68}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.69}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.70}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.71}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.72}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.73}$$

Equation (2.2) takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos\theta_i - q_1 \sin\theta_i \\ Y_i &= b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3 \end{aligned} \tag{5.74}$$

$$\begin{aligned} X_i &= a_i + p_2 \cos\theta_i - q_2 \sin\theta_i \\ Y_i &= b_i + p_2 \sin\theta_i + q_2 \cos\theta_i \qquad i = 3,4,5 \end{aligned} \tag{5.75}$$

$$\begin{aligned} X_i &= a_i + p_3 \cos\theta_i - q_3 \sin\theta_i \\ Y_i &= b_i + p_3 \sin\theta_i + q_3 \cos\theta_i \qquad i = 5,6 \end{aligned} \tag{5.76}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

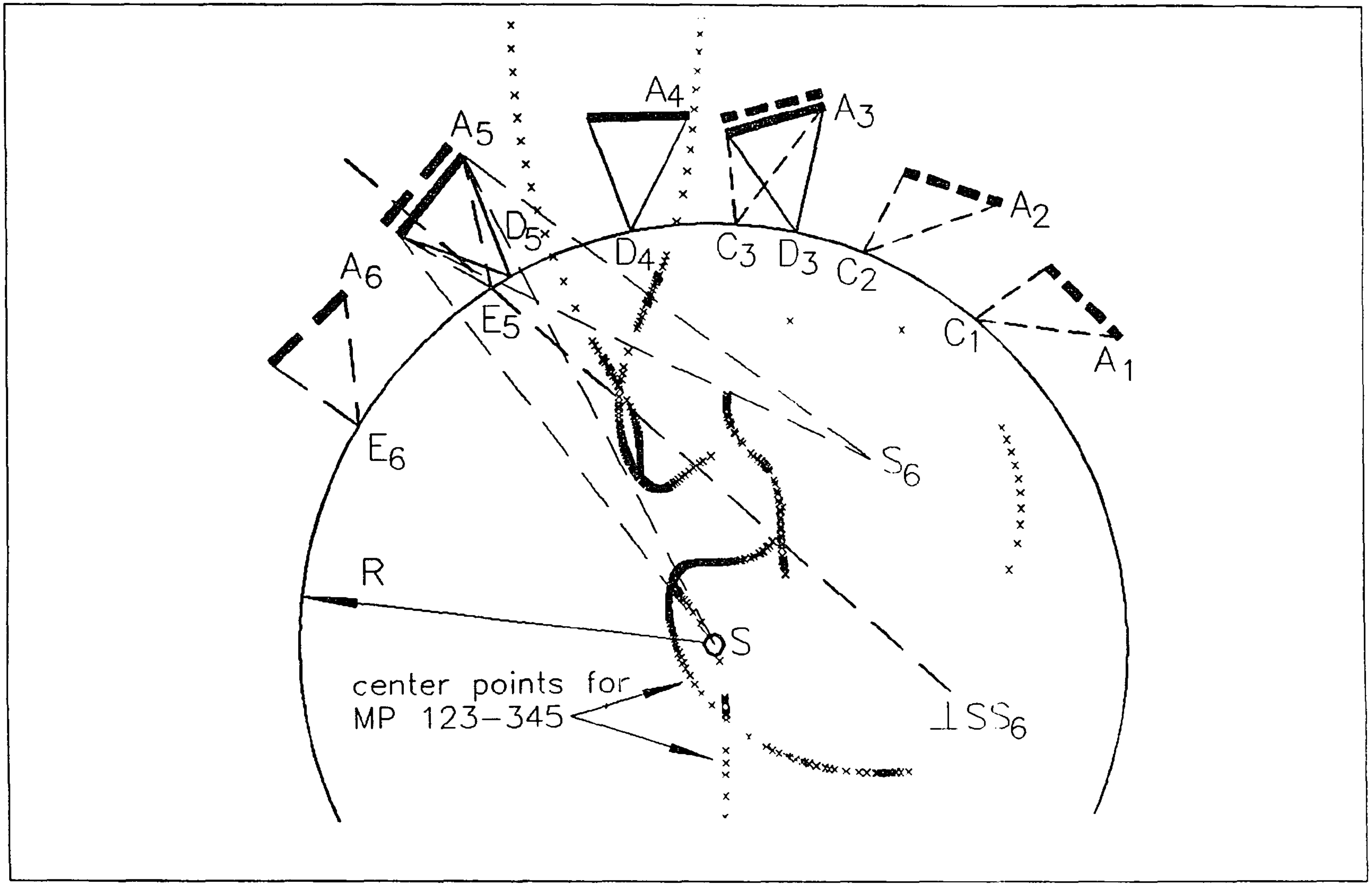

Figure 5.8 Adjustable moving pivot 123–345–56

Any solution for the two phase adjustable moving pivot problem MP 123–345, which has been solved in chapter 3 is a solution for the problem MP 123–345–56. The solution steps are similar to that for the problem in the last section.

Suppose seven prescribed positions and the center points for the two phase problem MP 123–345 are shown in Figure 5.8. A good center point S is picked on the curve, and the circle points C_1, C_2, C_3, D_3, D_4, and D_5 are found as shown.

The next goal is to find a circle point for positions 5 and 6, so that the center point S and the crank length R remain the same as that for phases 1 and 2. That is, the circle points E_5 and E_6 should lie on the circle passing through circle points C_1, C_2, C_3, D_3, D_4, and D_5.

Invert center point S from position 6 into position 5 to get point S_6. Plot a right bisector for line segment SS_6; this bisector intersects the circle at point E_5, which is the circle point at position 5 of phase 3. Finally, E_6 can be found by geometric similarity.

5.10 Positions 123–456–789

This problem needs three positions in phase 1 as well as phase 2 and phase 3 with no shared position. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \quad (5.77)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \quad (5.78)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \quad (5.79)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \quad (5.80)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \quad (5.81)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \quad (5.82)$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \quad (5.83)$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R^2 \tag{5.84}$$

$$(X_9 - P)^2 + (Y_9 - Q)^2 = R^2 \tag{5.85}$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos \theta_i - q_1 \sin \theta_i$$

$$Y_i = b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \tag{5.86}$$

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$

$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 4,5,6 \tag{5.87}$$

$$X_i = a_i + p_3 \cos \theta_i - q_3 \sin \theta_i$$

$$Y_i = b_i + p_3 \sin \theta_i + q_3 \cos \theta_i \qquad i = 7,8,9 \tag{5.88}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in nine equations. Thus, the equations have no free choice of parameter.

Plot center point curves MP 123–456 and MP 123–789 as shown in Figure 5.9 by means of the method developed in chapter 3. The solution, if it exists, should be at the intersection point of the center point curves MP 123–456 and MP 123–789. A good center point is found at the intersection point S in Figure 5.9. Figure 5.10 is an enlarged view at the vicinity of the intersection point S.

Invert center point S from positions 2 and 3 into position 1 to get circle points C_1, C_2, and C_3. Similarly, locate circle points D_4, D_5, and D_6 for phase 2, and circle points E_7, E_8, and E_9 for phase 3.

Notice that in the particular example shown in Figure 5.9, the crank rotates counterclockwise for positions 1 through 7 and then clockwise for positions 8 and 9.

As shown in the figure, all nine circle points lie precisely on the same circle with a unique center point S and radius R. This indicates the validity of both the method and the program.

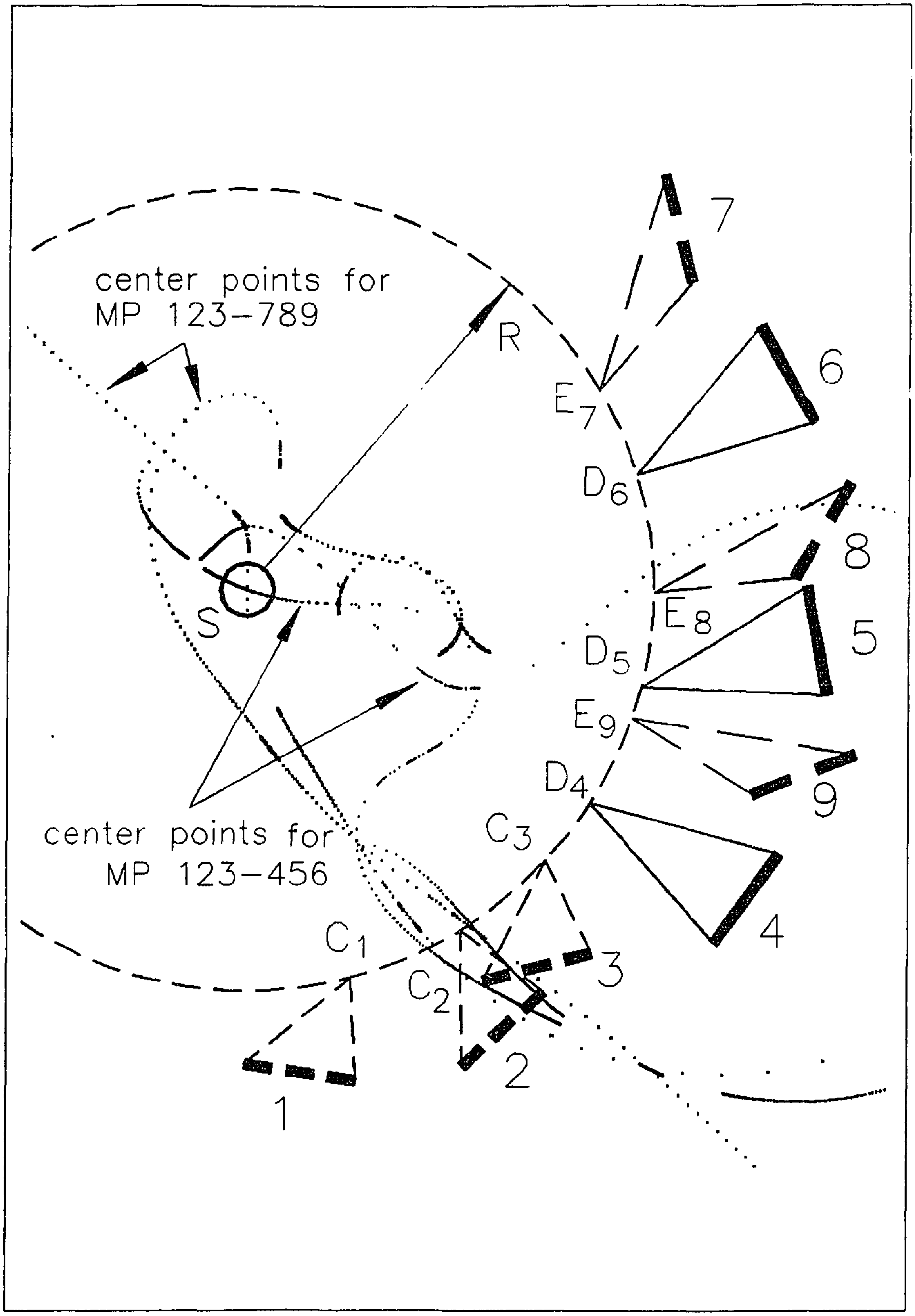

Figure 5.9 Adjustable moving pivot 123–456–789

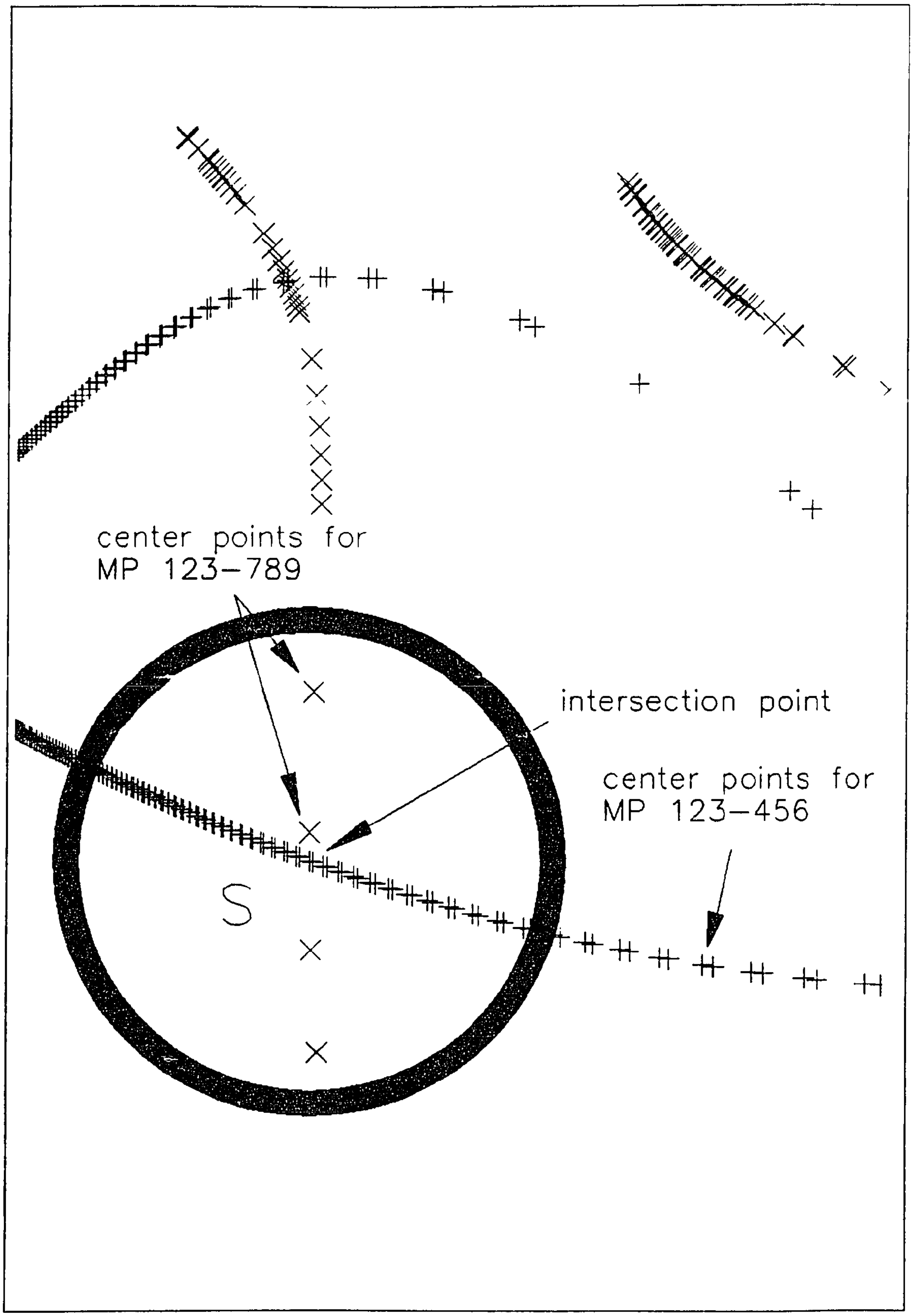

Figure 5.10 An enlarged view at intersection point S

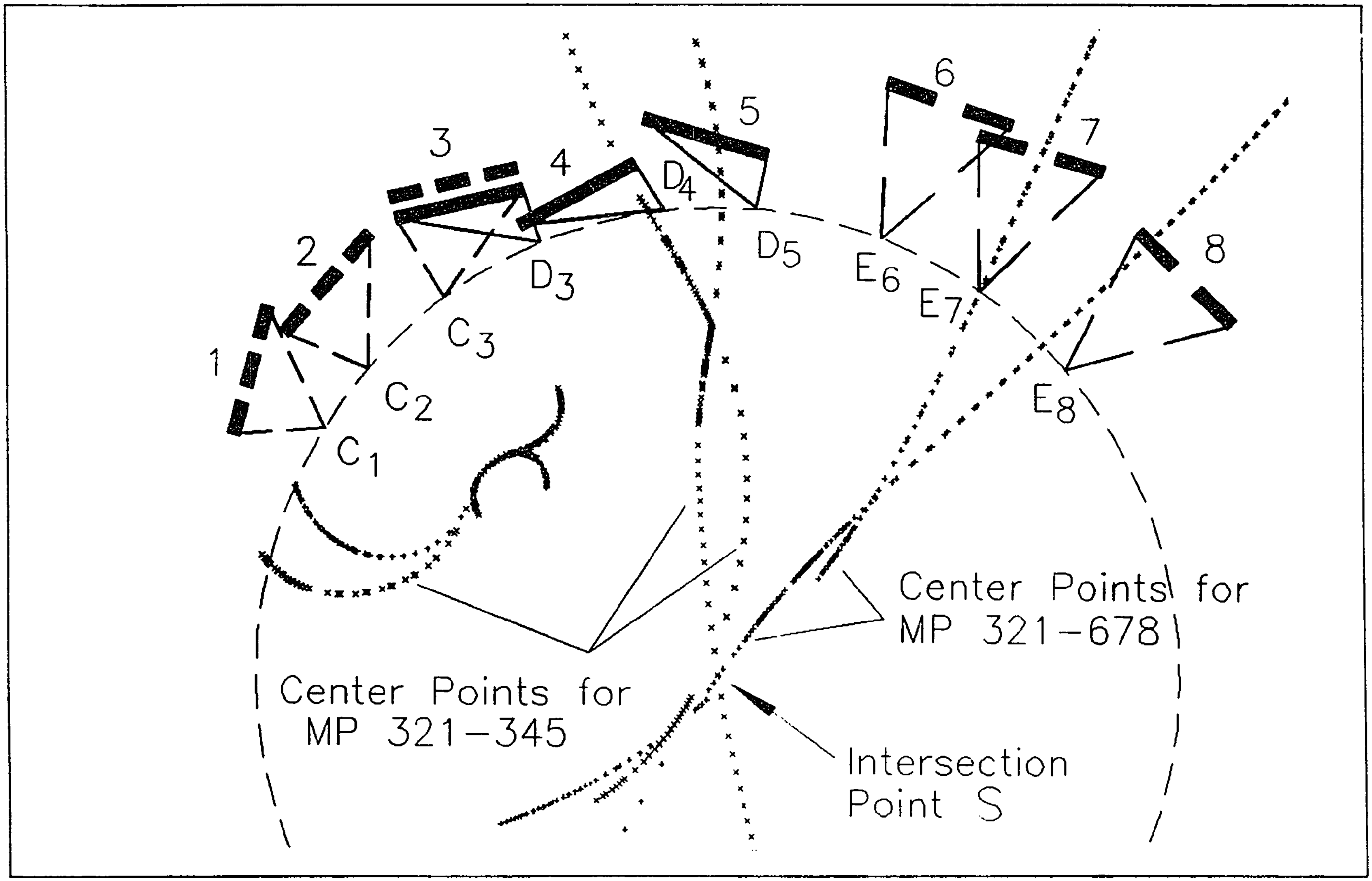

Figure 5.11 Adjustable moving pivot 123–345–678

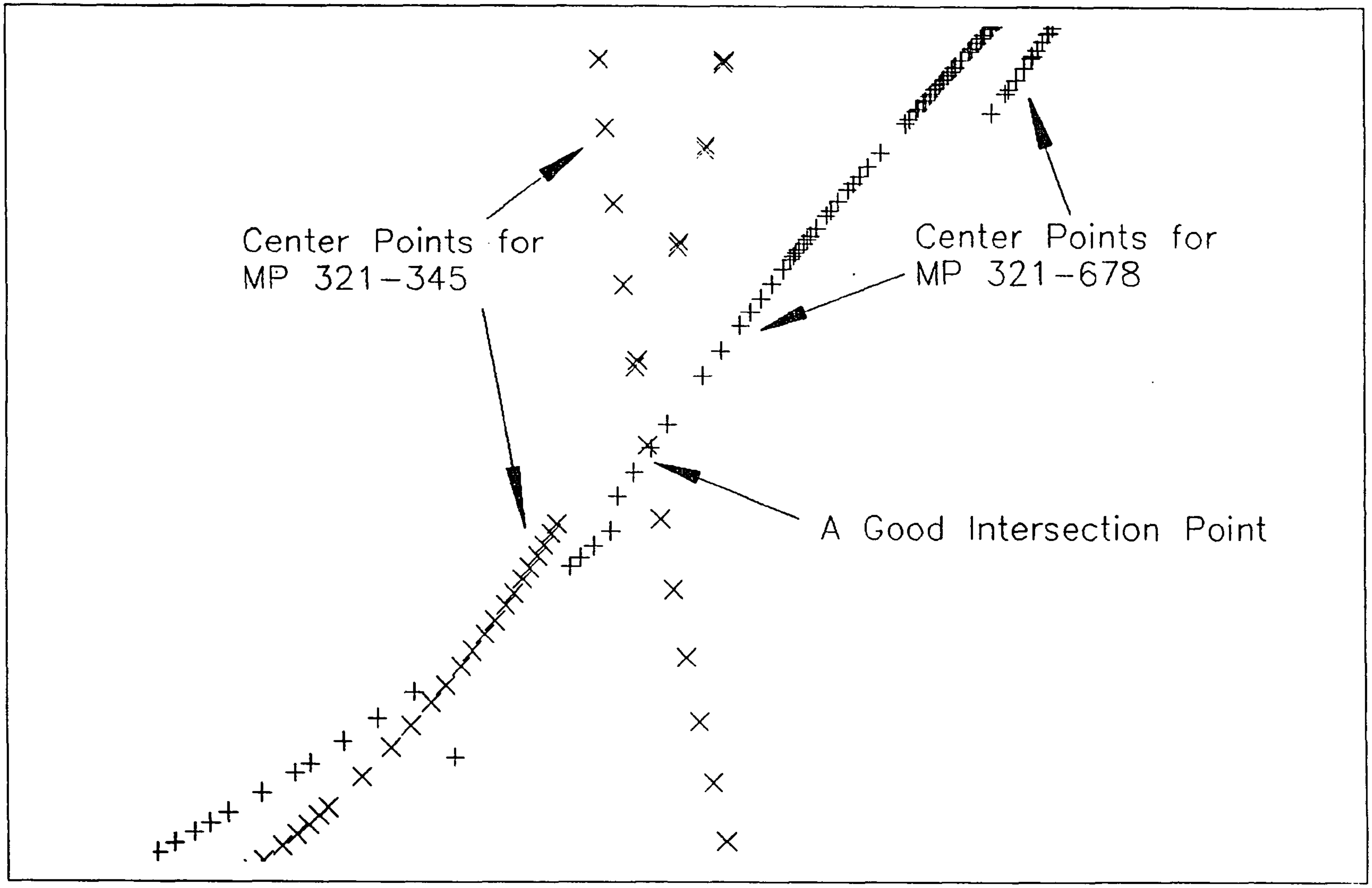

Figure 5.12 An enlarged view at intersection point S

5.11 Positions 123–345–678

This problem needs three positions in phase 1, phase 2, and phase 3 with one position shared by phases 1 and 2. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.89}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.90}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.91}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.92}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.93}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.94}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.95}$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R^2 \tag{5.96}$$

Equation (2.2) takes the form of

$$\begin{aligned} X_i &= a_i + p_1 \cos \theta_i - q_1 \sin \theta_i \\ Y_i &= b_i + p_1 \sin \theta_i + q_1 \cos \theta_i \qquad i = 1,2,3 \end{aligned} \tag{5.97}$$

$$\begin{aligned} X_i &= a_i + p_2 \cos \theta_i - q_2 \sin \theta_i \\ Y_i &= b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 3,4,5 \end{aligned} \tag{5.98}$$

$$\begin{aligned} X_i &= a_i + p_3 \cos \theta_i - q_3 \sin \theta_i \\ Y_i &= b_i + p_3 \sin \theta_i + q_3 \cos \theta_i \qquad i = 6,7,8 \end{aligned} \tag{5.99}$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in nine equations. Thus, the equations have no free choice of parameter.

Plot center point curves MP 321–345 and MP 321–678 as shown in Figure 5.11 by means of the method developed in chapter 3. The solution, if it exists, should be at the intersection point of the center point curves MP 321–345 and MP 321–678. A good center point is found at the intersection point S in Figure 5.11. Figure 5.12 is an enlarged view at the vicinity of the intersection point S.

Notice that the two phase adjustable moving pivot problems MP 321–345 and MP 123–345 represent the same problem, but MP 321–678 and MP 123–678 are different problems and will result in different center point curves.

Invert center point S from positions 2 and 3 into position 1 to get circle points C_1, C_2, and C_3. Similarly, locate circle points D_3, D_4, and D_5 for phase 2, and circle points E_6, E_7, and E_8 for phase 3. As shown in the figure, all nine circle points lie precisely on the same circle with a unique center point S and radius R. This indicates the validity of both the method and the program.

5.12 Positions 1234–567–89

This problem needs four positions in phase 1, three positions in phase 2, and two positions in phase 3 with no shared position. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R^2 \tag{5.100}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R^2 \tag{5.101}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R^2 \tag{5.102}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R^2 \tag{5.103}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R^2 \tag{5.104}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R^2 \tag{5.105}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R^2 \tag{5.106}$$

$$(X_8 - P)^2 + (Y_8 - Q)^2 = R^2 \tag{5.107}$$

$$(X_9 - P)^2 + (Y_9 - Q)^2 = R^2 \tag{5.108}$$

Equation (2.2) takes the form of

$$X_i = a_i + p_1 \cos\theta_i - q_1 \sin\theta_i$$

$$Y_i = b_i + p_1 \sin\theta_i + q_1 \cos\theta_i \qquad i = 1,2,3,4 \tag{5.109}$$

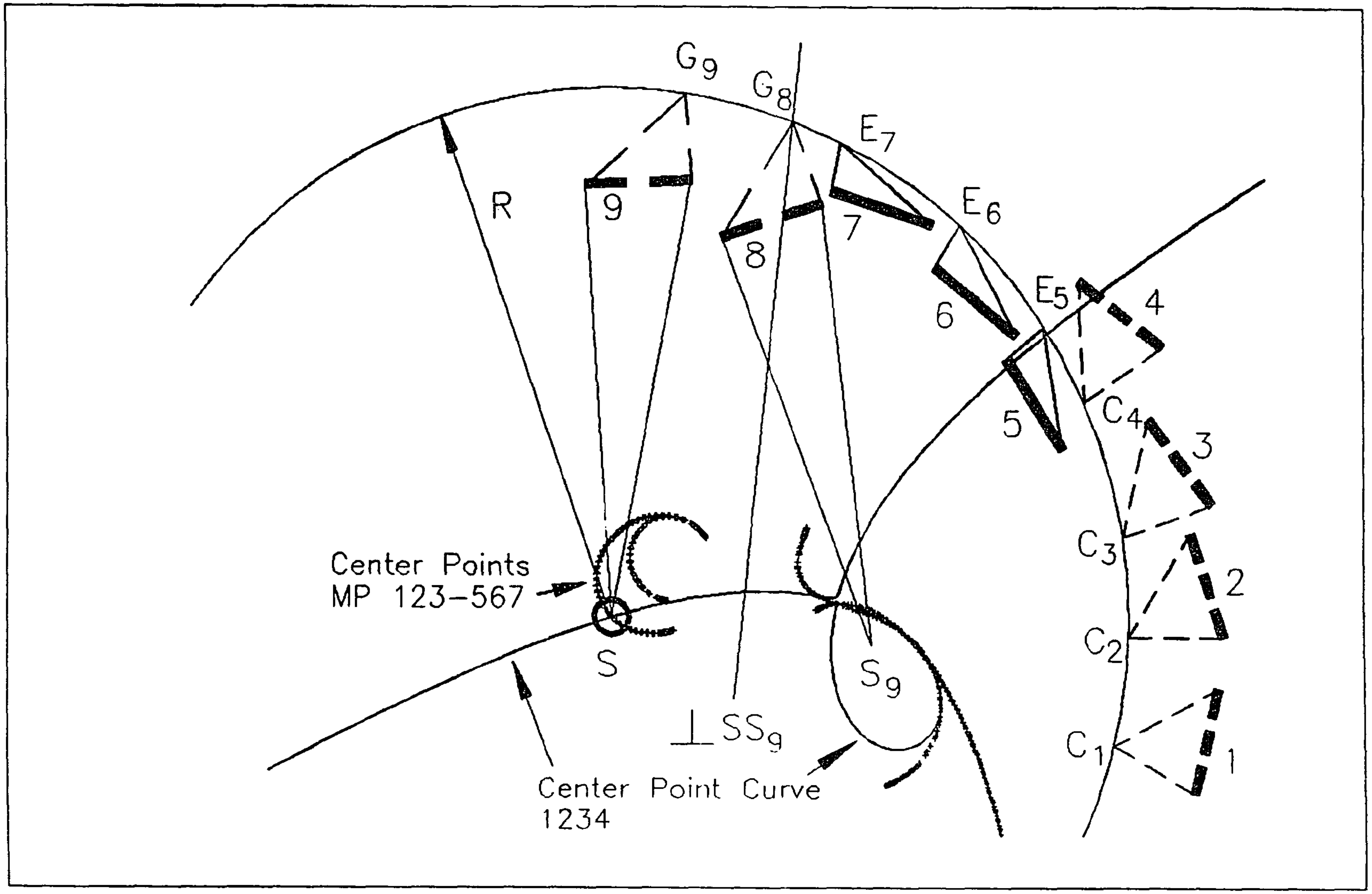

Figure 5.13 Adjustable moving pivot 1234–567–89

$$X_i = a_i + p_2 \cos \theta_i - q_2 \sin \theta_i$$
$$Y_i = b_i + p_2 \sin \theta_i + q_2 \cos \theta_i \qquad i = 5,6,7 \qquad (5.110)$$
$$X_i = a_i + p_3 \cos \theta_i - q_3 \sin \theta_i$$
$$Y_i = b_i + p_3 \sin \theta_i + q_3 \cos \theta_i \qquad i = 8,9 \qquad (5.111)$$

Nine parameters, P, Q, p_1, q_1, p_2, q_2, p_3, q_3, and R, are involved in nine equations. Thus, the number of free choice of parameter is zero.

An example problem is shown in Figure 5.13. The prescribed positions for phases 1 and 2 are the same as that in the example of section 3.8. A solution center point for the two phase adjustable moving pivot problem MP 1234–567 is also a solution center point for the three phase problem MP 1234–567–89.

Thus, the first step is to solve phases 1 and 2 as it has been done in section 3.8. The second step is to solve phase 3. Invert center point S from position 9 into position 8 to get point S_9. Plot a right bisector for line segment SS_9; this bisector intersects the circle passing through circle points C_1, C_2, C_3, C_4, E_5, E_6, and E_7 at G_8, which is the circle point at position 8 of phase 3. G_9 can be found by geometric similarity.

Chapter 6

Three Phase Adjustable Crank Length Problems

6.1 Introduction

This chapter deals with the problems of three phases with adjustable crank length. Seven parameters need to be determined for this group of problems, they are P, Q, p, q, R_1, R_2, and R_3. Thus, the maximum prescribed positions would be seven.

Three problems listed in Table 6.1 are going to be solved in this chapter. The minimum number of prescribed positions included in one phase is two, and the maximum number is three. The total number of positions for all three phases are seven which is the maximum allowable number.

Table 6.1 Three phase adjustable crank length problems

ph.1	positions ph.2	ph.3	shared pos.	unknowns	free choices
1,2	3,4	5,6	0	6	1
1,2,3	4,5	6,7	0	7	0
1,2	3,4,5	6,7	0	7	0

No shared positions need to be considered here, because the coupler positions coincide with each other at the shared position , and the positions of circle points at the shared position coincide with each other too. This will cause the same crank length for two phases with a shared position, which conflicts the original requirement. This is an adjustable crank length problem, different phases must have their own crank lengths, and neither center point nor circle point need to be adjusted for different phases.

6.2 Positions 12–34–56

For the case of two positions in each of the three phases with no position shared, the following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{6.1}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{6.2}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_2^2 \tag{6.3}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{6.4}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_3^2 \tag{6.5}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_3^2 \tag{6.6}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2,\ldots,6 \tag{6.7}$$

Seven parameters, P, Q, p, q, R_1, R_2, and R_3, are involved in six equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Eliminate R_1 from equations (6.1) and (6.2), we get

$$(X_2 - P)^2 + (Y_2 - Q)^2 = (X_1 - P)^2 + (Y_1 - Q)^2 \tag{6.8}$$

Substitute equation (6.7) into equation (6.8) and collect terms in P and Q,

$$\begin{aligned}
&[-p \cos \theta_2 + q \sin \theta_2 - a_2]\, P + [-q \cos \theta_2 - p \sin \theta_2 - b_2]\, Q + \\
&+ p\,(a_2 \cos \theta_2 + b_2 \sin \theta_2) + q\,(b_2 \cos \theta_2 - a_2 \sin \theta_2) + (a_2^2 + b_2^2) / 2 \\
&= [-p \cos \theta_1 + q \sin \theta_1 - a_1]\, P + [-q \cos \theta_1 - p \sin \theta_1 - b_1]\, Q + \\
&+ p\,(a_1 \cos \theta_1 + b_1 \sin \theta_1) + q\,(b_1 \cos \theta_1 - a_1 \sin \theta_1) + (a_1^2 + b_1^2) / 2
\end{aligned} \tag{6.9}$$

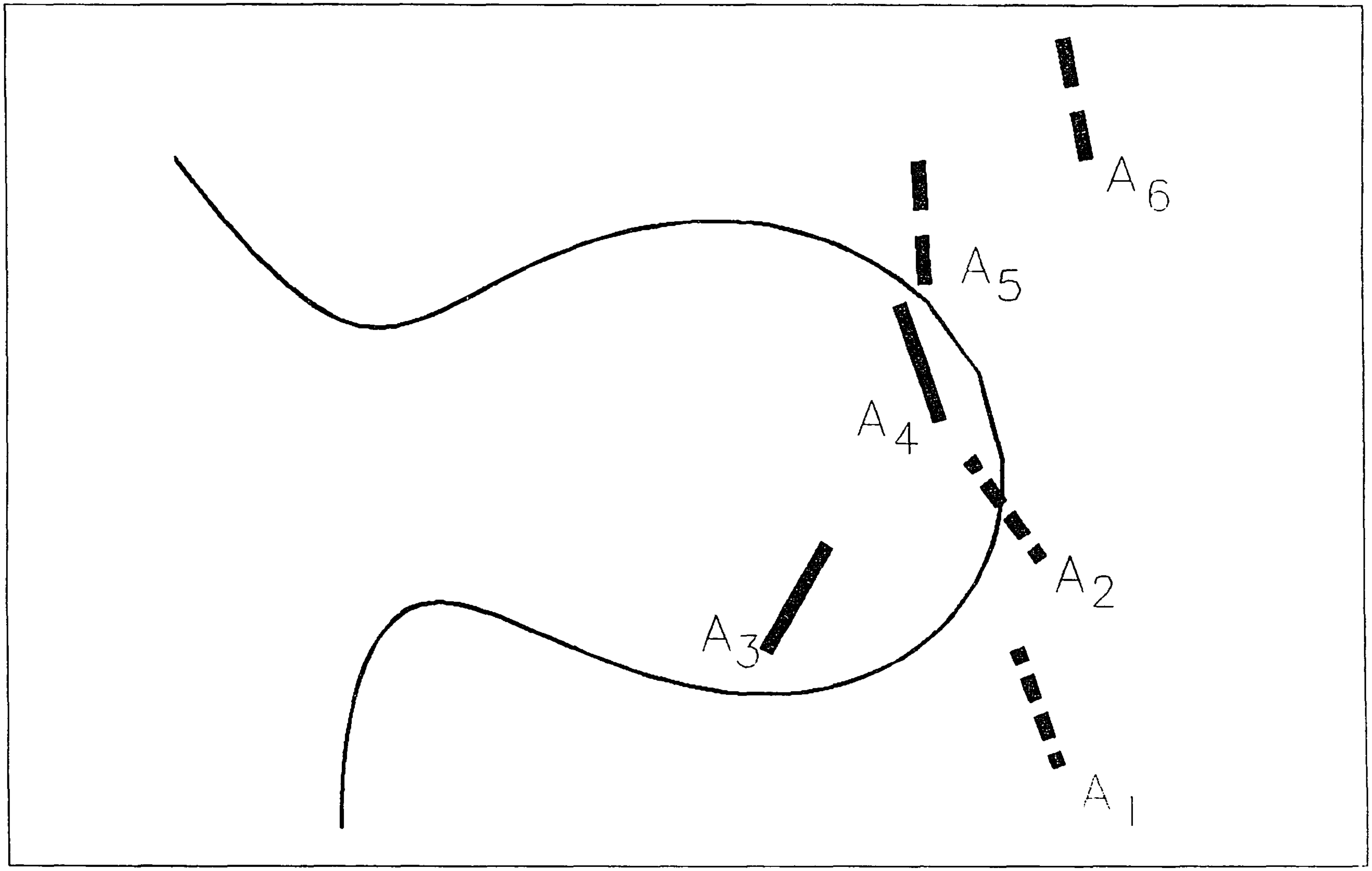

Figure 6.1 Circle point curve for CL 12–34–56

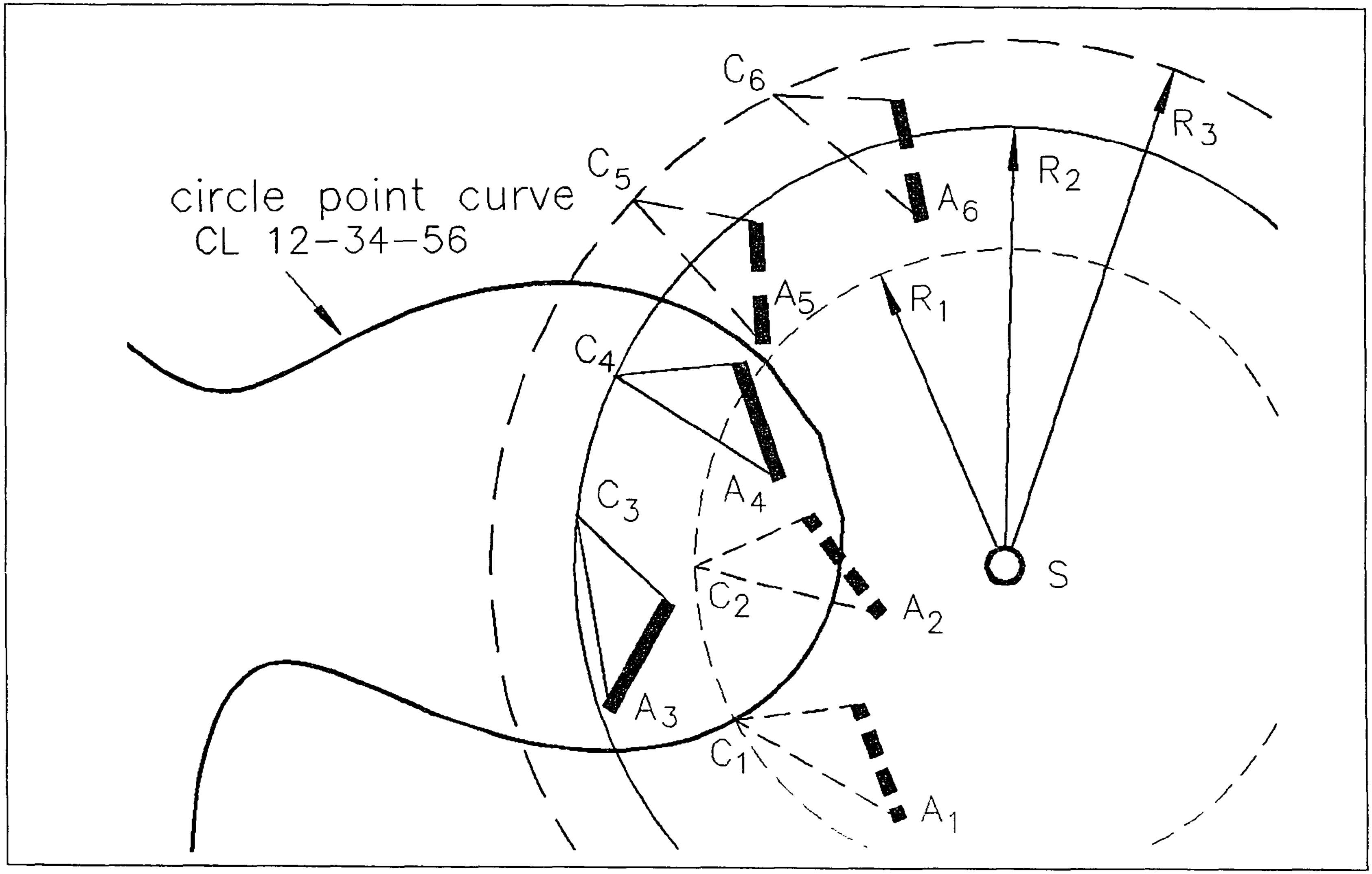

Figure 6.2 A good solution for adjustable crank length 12–34–56

that is,

$$L_{12}\,P + M_{12}\,Q + N_{12} = 0 \tag{6.10}$$

Similarly, we have

$$L_{34}\,P + M_{34}\,Q + N_{34} = 0 \tag{6.11}$$

$$L_{56}\,P + M_{56}\,Q + N_{56} = 0 \tag{6.12}$$

where

$$\begin{aligned}
L_{ij} &= -a_i + a_j - p(\cos\theta_i - \cos\theta_j) + q(\sin\theta_i - \sin\theta_j)\\
M_{ij} &= -b_i + b_j - p(\sin\theta_i - \sin\theta_j) - q(\cos\theta_i - \cos\theta_j)\\
N_{ij} &= p\,(a_i\cos\theta_i - a_j\cos\theta_j + b_i\sin\theta_j - b_j\sin\theta_j)\\
&\quad -q\,(a_i\sin\theta_i - a_j\sin\theta_j - b_i\cos\theta_i + b_j\cos\theta_j)\\
&\quad +\ (a_i^2 - a_j^2 + b_i^2 - b_j^2)/2
\end{aligned} \tag{6.13}$$

where $i = 1, 3, 5$, and $j = 2, 4, 6$ respectively.

For a nontrivial solution for P and Q in equations (6.10) through (6.12), the following determinant must be equal to zero:

$$\begin{vmatrix} L_{12} & M_{12} & N_{12} \\ L_{34} & M_{34} & N_{34} \\ L_{56} & M_{56} & N_{56} \end{vmatrix} = 0 \tag{6.14}$$

Expand equation (6.14), after considerable derivation the following circle point curve equation is obtained:

$$(Ap + Bq)(p^2 + q^2) + Cpq + Dp^2 + Eq^2 + Fp + Gq + H = 0 \tag{6.15}$$

where A, B, C, D, E, E, F, G, and H are functions of a_i, b_i, and θ_i. Points which satisfy equation (6.15) should satisfy equations (6.1) through (6.7) for the given synthesis problem.

The expressions for the coefficients A through H are similar to that in Wilhelm's work [12]. The Turbo Pascal program CL_2_2_2.PAS is designed for finding the circle points for the synthesis problem of this section.

An example problem with six prescribed positions is shown in Figure 6.1. The circle point curve is plotted in the figure by running the program CL_2_2_2.PAS along with AutoCAD on IBM PC. A good circle point C_1 is chosen on the curve as shown in Figure 6.2. Find circle points C_2, C_3, C_4, C_5, and C_6 by geometric similarity. The center point S could be found by intersecting right bisectors for line segments C_1C_2 and C_3C_4. Draw two circles with center S passing through C_1 and C_3. The radii of the circles, R_1 and R_2 are crank lengths for phases 1 and 2 respectively.

The crank length for phase 3 can be found by plotting the third circle passing through circle point C_5. As shown in the figure, the circle also precisely passes through circle point C_6. This indicates the validity of both the method and the program CL_2_2_2.PAS. The crank length R_3 for phase 3 is the radius of the third circle.

6.3 Positions 123–45–67

This problem needs three positions in phase 1 and two positions in both phases 2 and 3 with no shared position. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \tag{6.16}$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \tag{6.17}$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_1^2 \tag{6.18}$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \tag{6.19}$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \tag{6.20}$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_3^2 \tag{6.21}$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_3^2 \tag{6.22}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

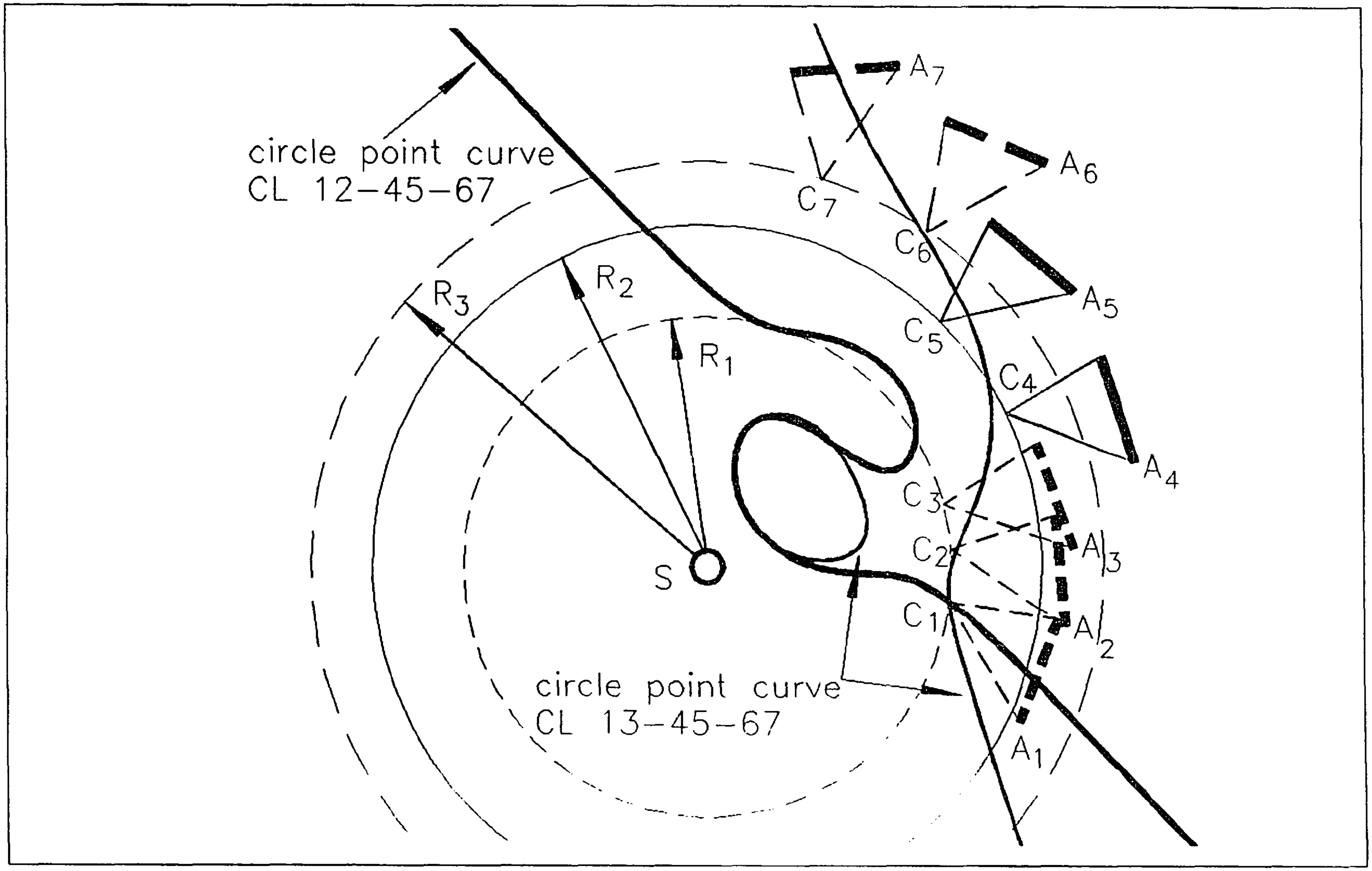

Figure 6.3 Adjustable crank length 123–45–67

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2,...,7 \qquad (6.23)$$

The total number of prescribed positions reaches the maximum value. An example problem is shown in Figure 6.3. The solution circle point, if exists should be at the intersection point of the circle point curves CL 12–45–67 and CL 13–45–67. A good circle point C_1 is found as shown. The circle points C_2 through C_7 are found by geometric similarity. The center point S for all three phases is found by intersecting right bisectors for line segments C_1C_2 and C_2C_3.

Notice that for all seven prescribed positions of all three phases, the center point S and the circle point C are the same. The crank lengths R_1, R_2, and R_3 are for phases 1, 2, and 3 respectively. This indicates the validity of both the method and the program CL_2_2_2.PAS.

Also, no order defect occurs in Figure 6.3.

6.4 Positions 12–345–67

This problem needs three positions in phase 2 and two positions in both phases 1 and 3 with no shared position. The following equations should be satisfied:

$$(X_1 - P)^2 + (Y_1 - Q)^2 = R_1^2 \qquad (6.24)$$

$$(X_2 - P)^2 + (Y_2 - Q)^2 = R_1^2 \qquad (6.25)$$

$$(X_3 - P)^2 + (Y_3 - Q)^2 = R_2^2 \qquad (6.26)$$

$$(X_4 - P)^2 + (Y_4 - Q)^2 = R_2^2 \qquad (6.27)$$

$$(X_5 - P)^2 + (Y_5 - Q)^2 = R_2^2 \qquad (6.28)$$

$$(X_6 - P)^2 + (Y_6 - Q)^2 = R_3^2 \qquad (6.29)$$

$$(X_7 - P)^2 + (Y_7 - Q)^2 = R_3^2 \qquad (6.30)$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

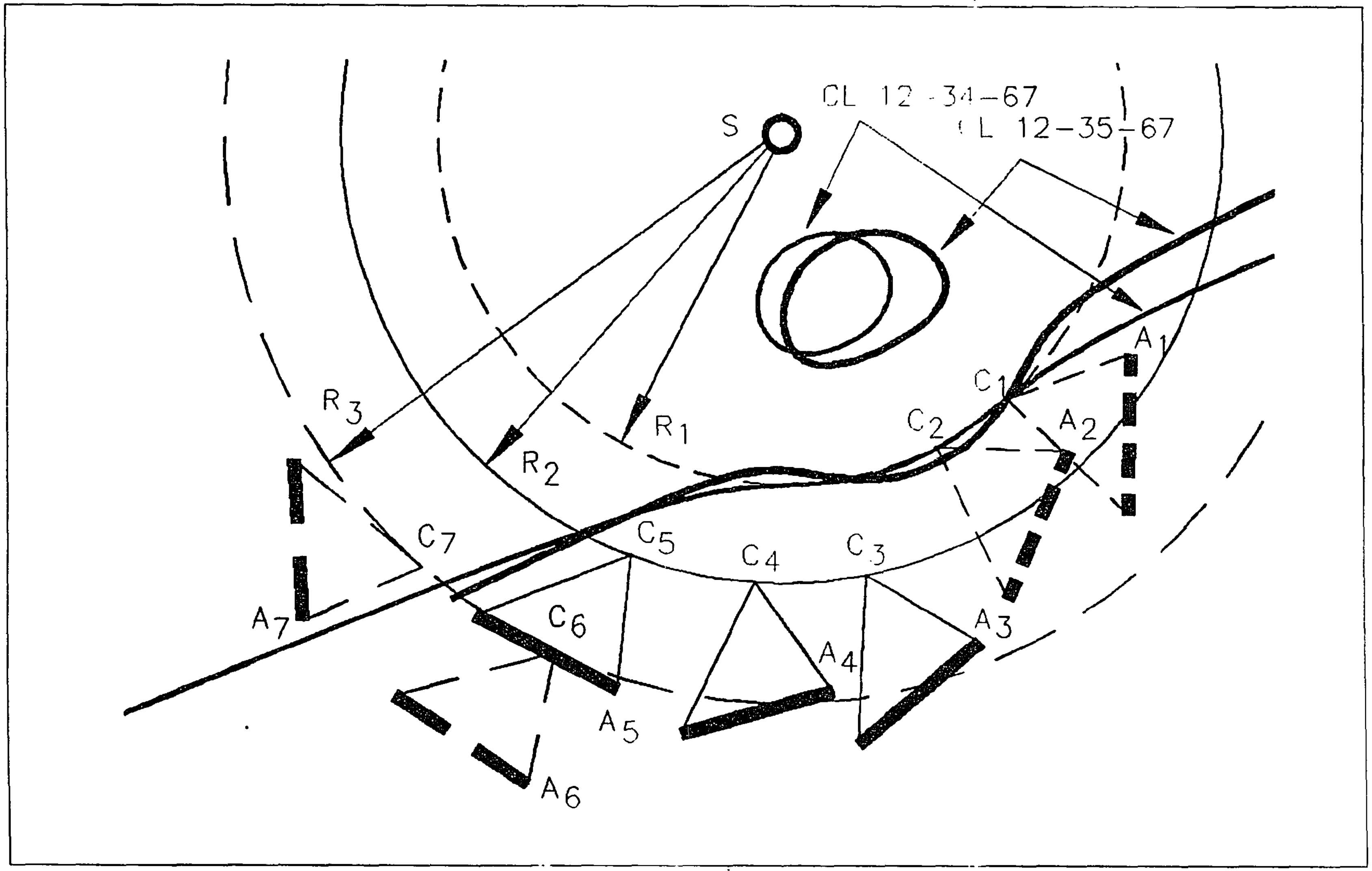

Figure 6.4 Adjustable crank length 12–345–67

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2,...,7 \qquad (6.31)$$

The total number of prescribed positions reaches the maximum value. An example problem is shown in Figure 6.4. Similar to the problem in the last section, the solution circle point, if exists should be at the intersection point of the circle point curves CL 12–34–67 and CL 12–35–67. A good circle point C_1 is found as shown. The circle points C_2 through C_7 are found by geometric similarity. The center point S for all three phases is found by intersecting right bisectors for line segments C_3C_4 and C_4C_5.

For all seven prescribed positions in all three phases, the center point S and the circle point C are the same. The crank lengths R_1, R_2, and R_3 are for phases 1, 2, and 3 respectively. This indicates the validity of both the method and the program CL_2_2_2.PAS.

Also, no order defect occurs in Figure 6.4.

Chapter 7

Three Phase Adjustable Fixed Pivot Problems

7.1 Introduction

This chapter deals with the problem of three phase adjustable fixed pivot. Nine parameters which need to be determined for this group of problems are P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R. Thus, the maximum prescribed positions would be nine.

Six problems listed in Table 7.1 are going to be solved in this chapter. The minimum number of prescribed positions included in one phase is two, and the maximum number is three. The maximum total number of positions is nine, which is the maximum possible value. The maximum number of shared positions is one.

Table 7.1 Three phase adjustable fixed pivot problems

positions ph.1	 ph.2	 ph.3	shared pos.	unknowns	free choices
1,2,3	4,5	6,7	0	7	2
1,2,3	3,4	5,6	1	7	2
1,2,3	4,5,6	7,8	0	8	1
1,2,3	3,4,5	6,7	1	8	1
1,2,3	4,5,6	7,8,9	0	9	0
1,2,3	3,4,5	6,7,8	1	9	0

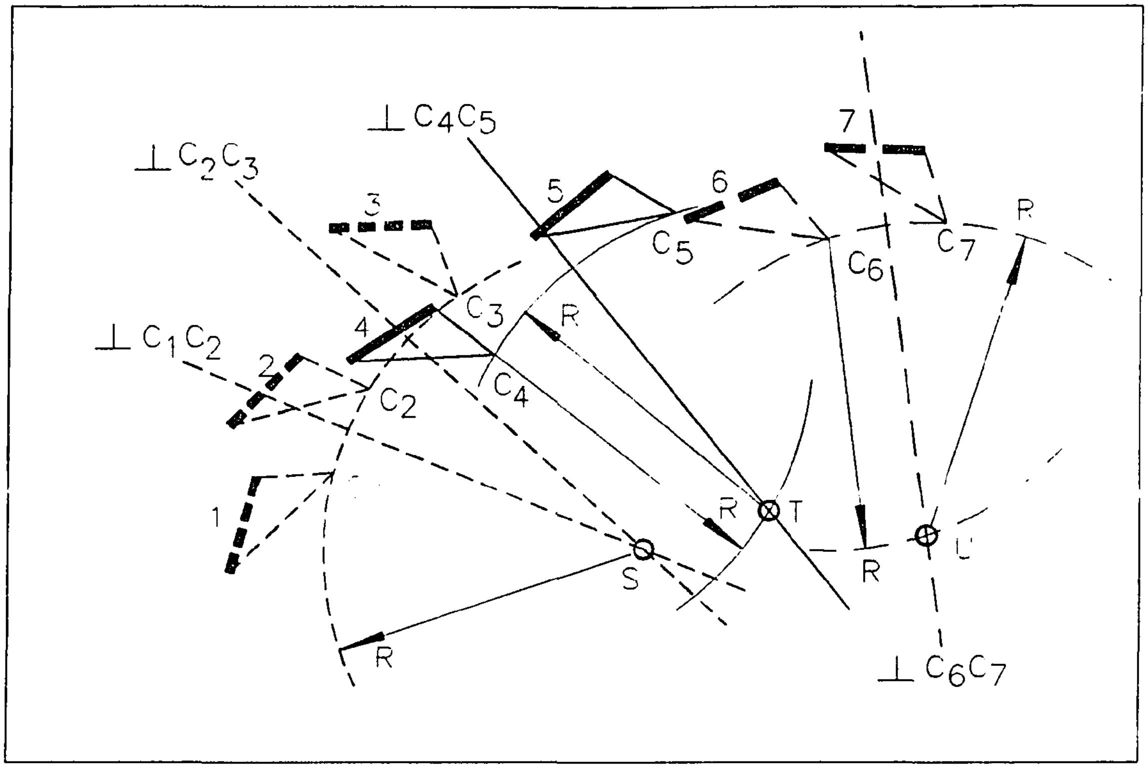

Figure 7.1 Adjustable fixed pivot 123–45–67

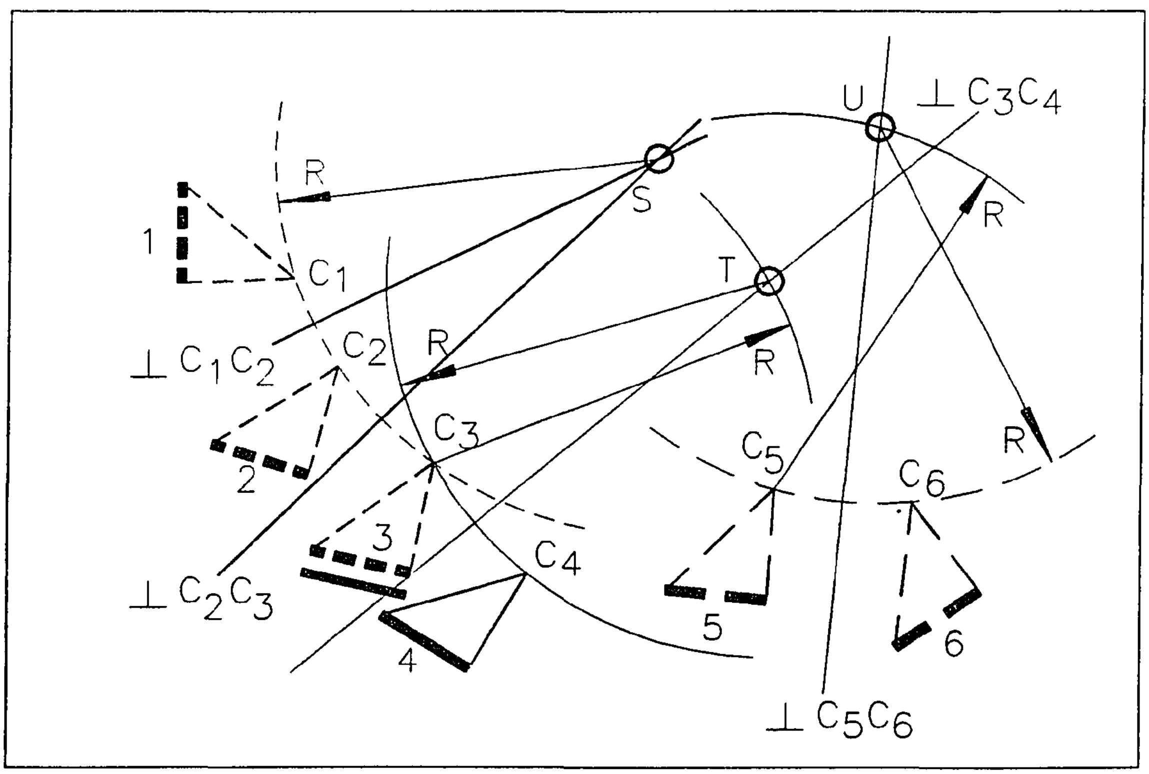

Figure 7.2 Adjustable fixed pivot 123–34–56

In this chapter the method for solving three phase adjustable fixed pivot problems will be developed.

7.2 Positions 123–45–67

In this case, three positions are needed for phase 1, two positions are required for both phases 2 and 3 with no position shared. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \qquad (7.1)$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \qquad (7.2)$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \qquad (7.3)$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \qquad (7.4)$$

$$(X_5 - P_2)^2 + (Y_5 - Q_2)^2 = R^2 \qquad (7.5)$$

$$(X_6 - P_3)^2 + (Y_6 - Q_3)^2 = R^2 \qquad (7.6)$$

$$(X_7 - P_3)^2 + (Y_7 - Q_3)^2 = R^2 \qquad (7.7)$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,7 \qquad (7.8)$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in seven equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

Suppose seven prescribed positions are shown in Figure 7.1, and the relative coordinates of the circle point C_1 are chosen as two free choices. Find C_2 through C_7 by geometric similarity after locating C_1. Draw right bisectors for line segments C_1C_2 and C_2C_3. Find the intersection point S of the two right bisectors. Plot a circle passing through circle points C_1, C_2 and C_3 with center S. The radius of the circle is the crank length R. Draw a circle with center at C_4 and radius R; this circle intersects the right bisector for the line segment C_4C_5 at point T, which is the center point for phase 2. Similarly, find the center point U for phase 3.

7.3 Positions 123–34–56

In this case, three positions are needed for phase 1, two positions are required for both phases 2 and 3 with one position shared by phases 1 and 2. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \tag{7.9}$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \tag{7.10}$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \tag{7.11}$$

$$(X_3 - P_2)^2 + (Y_3 - Q_2)^2 = R^2 \tag{7.12}$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \tag{7.13}$$

$$(X_5 - P_3)^2 + (Y_5 - Q_3)^2 = R^2 \tag{7.14}$$

$$(X_6 - P_3)^2 + (Y_6 - Q_3)^2 = R^2 \tag{7.15}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,6 \tag{7.16}$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in seven equations. Thus, the equations can be solved with two free choices of parameters, and have infinite solutions. Either an algebraic method or a graphic method can solve this problem.

The solution steps are almost the same as that in the last section. Six prescribed positions are shown in Figure 7.2. As in the last section, p and q are chosen as the two free choices of parameters. The work for phase 1 is the same as that in the last section. For phase 2, plot a right bisector for line segment C_3C_4. Draw a circle with radius R and center C_3; this circle intersects the bisector at T, which is the center point for phase 2. For phase 3, plot a bisector for line segment C_5C_6. Draw a circle with radius R and center C_5; this circle intersects the bisector at U, which is the center point for phase 3.

7.4 Positions 123–456–78

This problem needs three positions in phases 1 and 2, two positions in phase 3 with no position shared. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \tag{7.17}$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \tag{7.18}$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \tag{7.19}$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \tag{7.20}$$

$$(X_5 - P_2)^2 + (Y_5 - Q_2)^2 = R^2 \tag{7.21}$$

$$(X_6 - P_2)^2 + (Y_6 - Q_2)^2 = R^2 \tag{7.22}$$

$$(X_7 - P_3)^2 + (Y_7 - Q_3)^2 = R^2 \tag{7.23}$$

$$(X_8 - P_3)^2 + (Y_8 - Q_3)^2 = R^2 \tag{7.24}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,8 \tag{7.25}$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Any solution for the two phase adjustable fixed pivot problem FP 123–456 is a solution for the three phase problem FP 123–456–78.

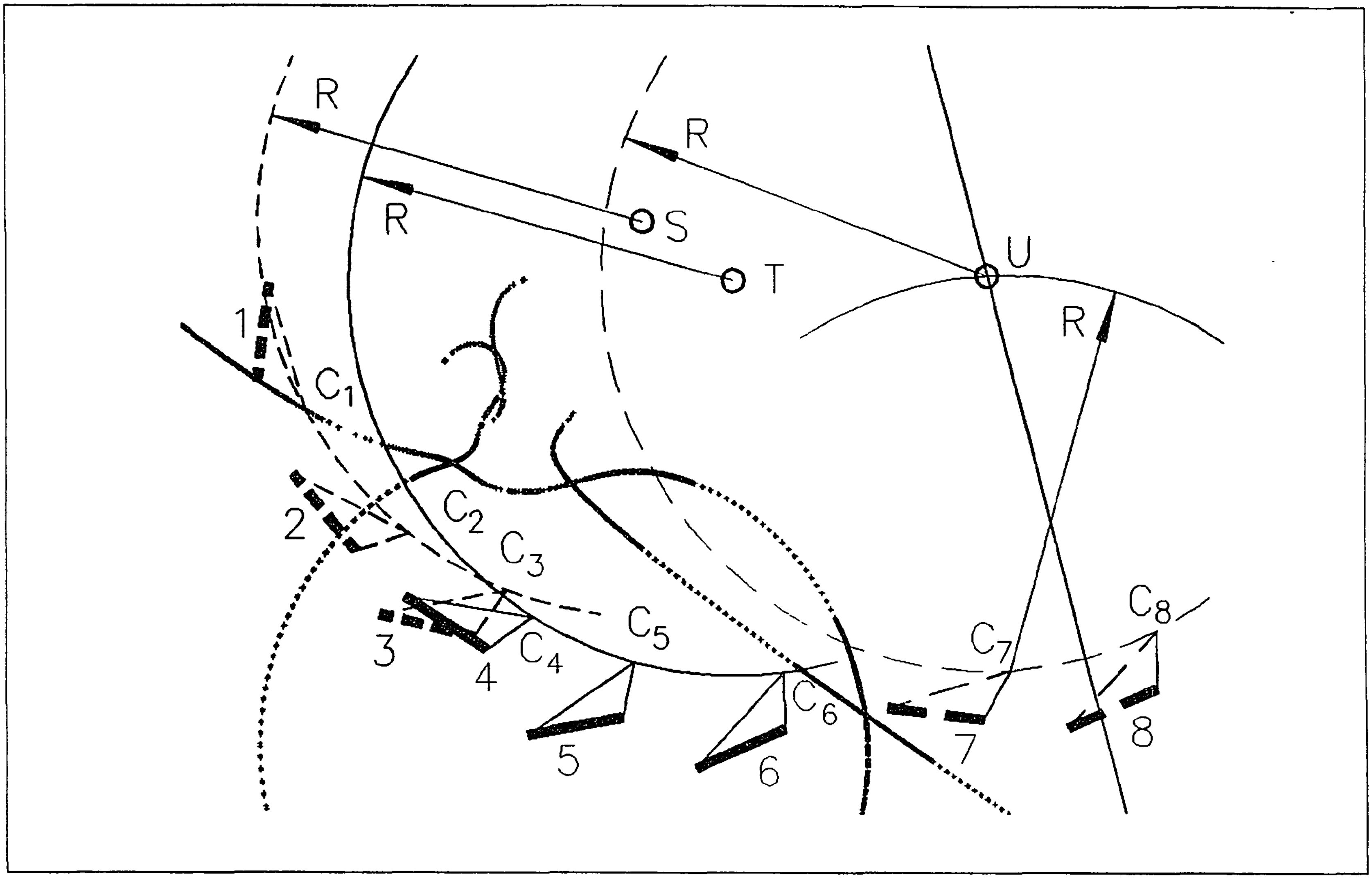

Figure 7.3 Adjustable fixed pivot 123–456–78

Suppose eight prescribed positions are shown in Figure 7.3, and the circle points for phases 1 and 2 are plotted by means of the program FP_3_3.PAS[12]. A good circle point C_1 for position 1 is picked on the curve. Find C_2 through C_8 for positions 2 through 8 by geometric similarity. The center point S for phase 1 should be located at the center of the circle passing through circle points C_1, C_2, and C_3. The radius of the circle is the unique crank length R.

Similarly, the center point T for phase 2 can be found at the center of the circle passing through the circle points C_4, C_5, and C_6. The radius of the circle is also equal to the crank length R.

The center point U for phase 3 can be found by intersecting the right bisector for line segment C_7C_8 and the circle with center C_7 and radius R.

7.5 Positions 123–345–67

This problem needs three positions in phases 1 and 2, and two positions in phase 3 with one position shared by phases 1 and 2. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \tag{7.26}$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \tag{7.27}$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \tag{7.28}$$

$$(X_3 - P_2)^2 + (Y_3 - Q_2)^2 = R^2 \tag{7.29}$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \tag{7.30}$$

$$(X_5 - P_2)^2 + (Y_5 - Q_2)^2 = R^2 \tag{7.31}$$

$$(X_6 - P_3)^2 + (Y_6 - Q_3)^2 = R^2 \tag{7.32}$$

$$(X_7 - P_3)^2 + (Y_7 - Q_3)^2 = R^2 \tag{7.33}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

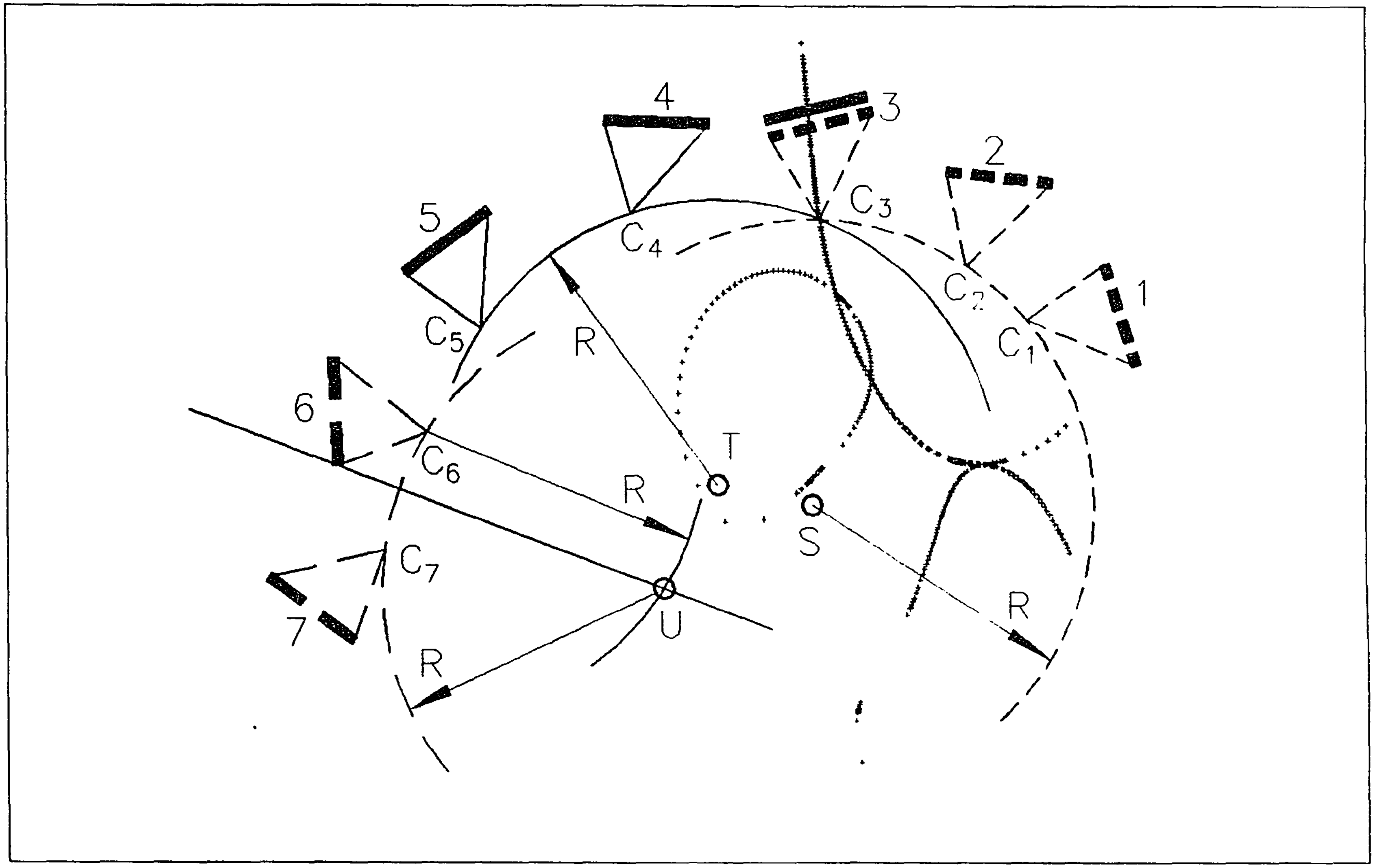

Figure 7.4 Adjustable fixed pivot 123–345–67

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,7 \quad (7.34)$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in eight equations. Thus, the equations can be solved with one free choice of parameter, and have infinite solutions.

Any solution for the two phase adjustable fixed pivot problem FP 123–345 is a solution for the three phase problem FP 123–345–67.

Suppose seven prescribed positions are shown in Figure 7.4, and the circle points for phases 1 and 2 are plotted by means of the program FP_3_3_1.PAS. A circle point C_3 for position 3 is picked on the curve. Find C_1, C_2, C_4, C_5, C_6, and C_7 by geometric similarity. The center point S for phase 1 should be located at the center of the circle passing through circle points C_1, C_2, and C_3. The radius of the circle is the unique crank length R.

Similarly, the center point T for phase 2 can be found at the center of the circle passing through the circle points C_3, C_4, and C_5. The radius of the circle is also equal to the crank length R.

The center point U for phase 3 can be found by intersecting the right bisector for line segment C_6C_7 and the circle with center C_6 and radius R.

7.6 Positions 123–456–789

This problem needs three positions in phases 1, 2, and 3 with no position shared. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \qquad (7.35)$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \qquad (7.36)$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \qquad (7.37)$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \qquad (7.38)$$

$$(X_5 - P_2)^2 + (Y_5 - Q_2)^2 = R^2 \qquad (7.39)$$

$$(X_6 - P_2)^2 + (Y_6 - Q_2)^2 = R^2 \qquad (7.40)$$

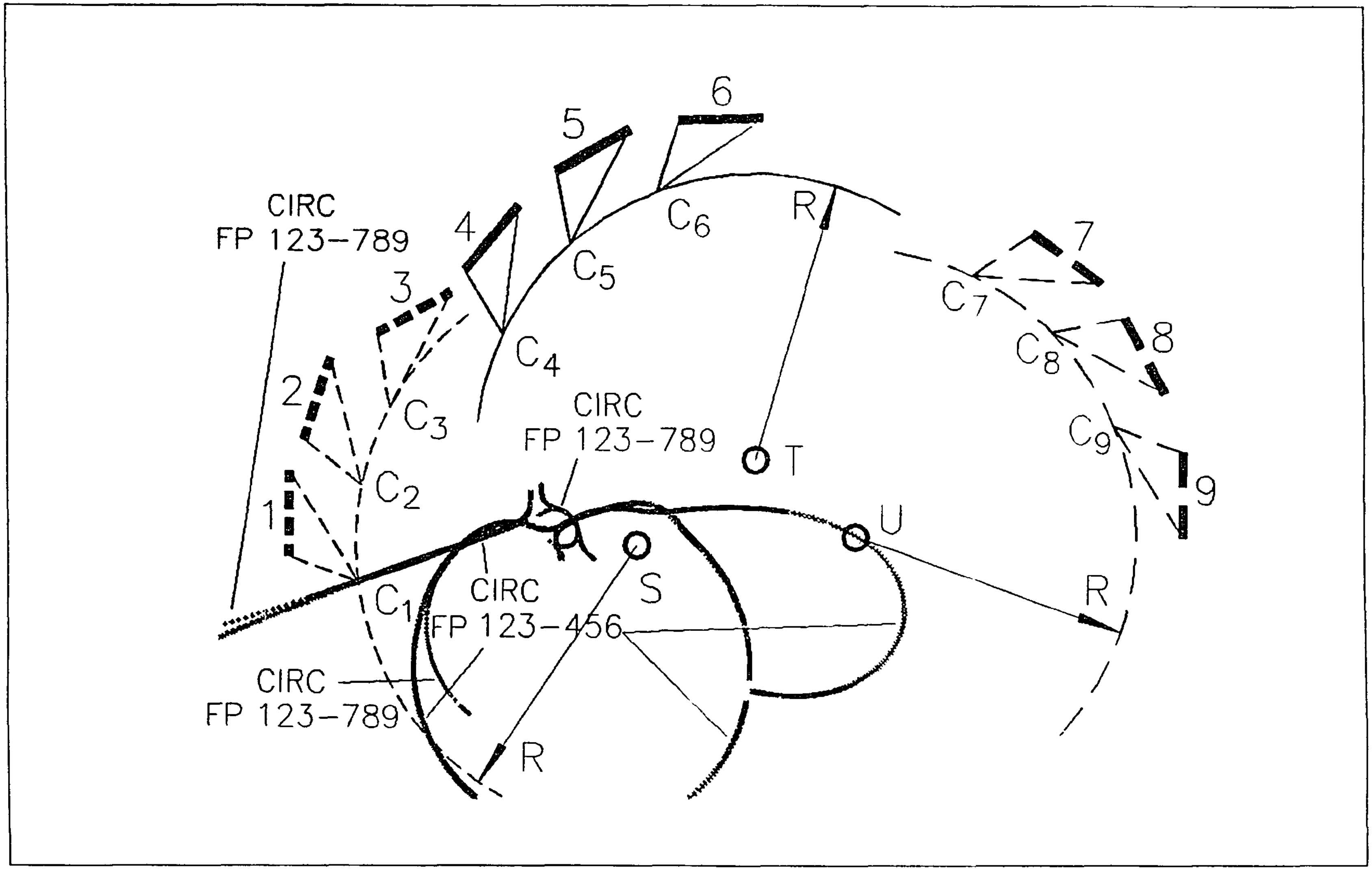

Figure 7.5 Adjustable fixed pivot 123–456–789

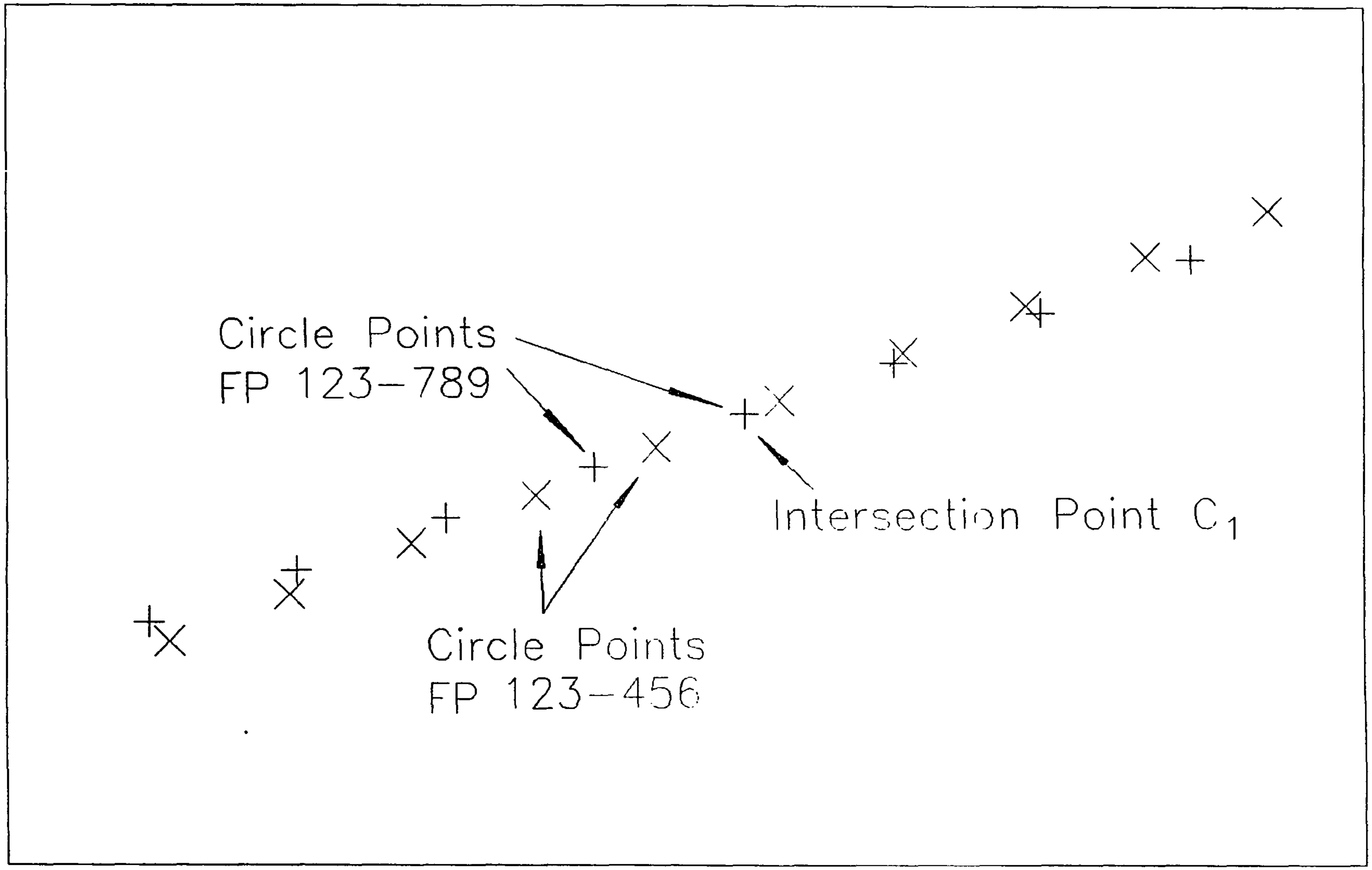

Figure 7.6 **An enlarged view at the intersection point C_1**

$$(X_7 - P_3)^2 + (Y_7 - Q_3)^2 = R^2 \tag{7.41}$$

$$(X_8 - P_3)^2 + (Y_8 - Q_3)^2 = R^2 \tag{7.42}$$

$$(X_9 - P_3)^2 + (Y_9 - Q_3)^2 = R^2 \tag{7.43}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,9 \quad (7.44)$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in nine equations. Thus, the equations have no free choice of parameter.

Suppose nine prescribed positions are shown in Figure 7.5. The solution, if exists, should be at the intersection point of the circle point curves FP 123–456 and FP 123–789.

Plot circle points for FP 123–456 and FP 123–789 in Figure 7.5. A good circle point is found at the intersection point C_1. Locate C_2, C_3, ..., C_9 by geometric similarity.

The center S of the circle passing through circle points C_1, C_2, and C_3 is the center point for phase 1. The radius of the circle is the crank length R. Similarly, the center T of the circle passing through circle points C_4, C_5, and C_6 is the center point for phase 2. Finally, the center U of the circle passing through circle points C_7, C_8, and C_9 is the center point for phase 3. The radii of three circles equal to the unique crank length R.

It can be seen in the figure that the unique circle point is C for all 9 positions, and the unique crank length is R for all three phases. The center point is moved form S to T, and then to U.

Figure 7.6 is an enlarged view at the intersection point C_1. In order to get a more precise solution, more points are plotted in the figure.

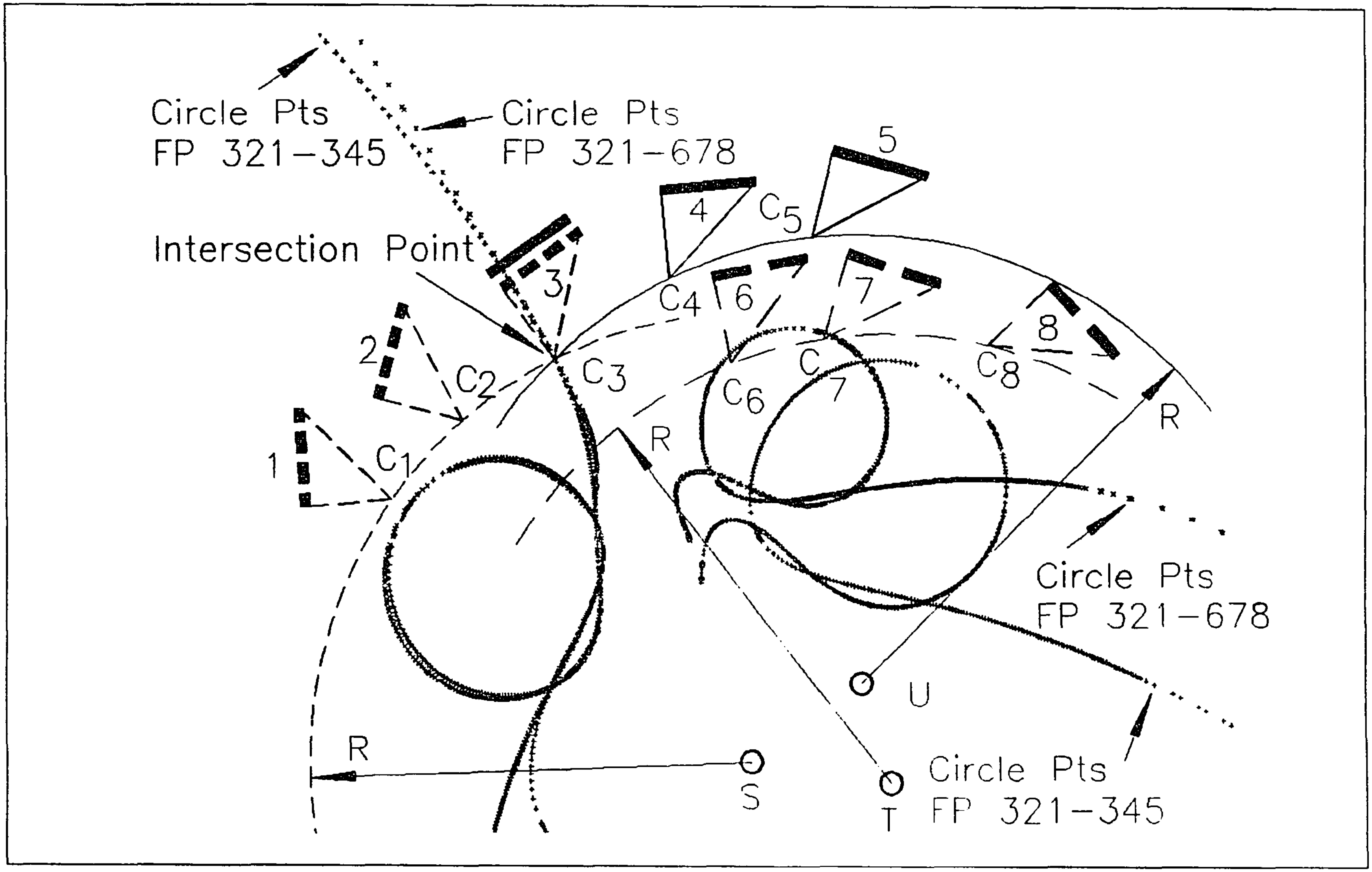

Figure 7.7 Adjustable fixed pivot 123–345–678

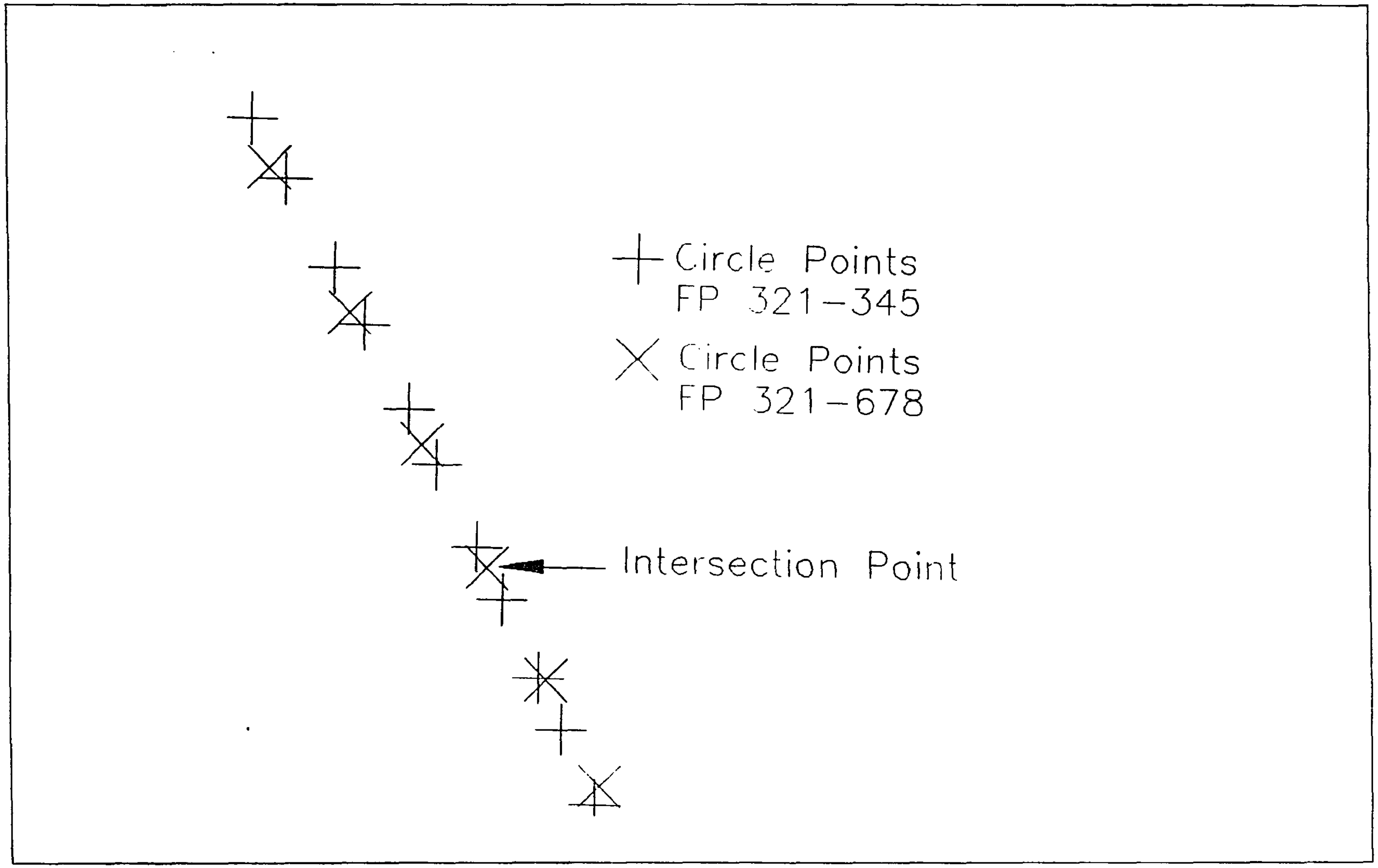

Figure 7.8 An enlarged view at the intersection point C_3

7.7 Positions 123–345–678

This problem needs three positions in phases 1, 2, and 3 with one position shared by phases 1 and 2. The following equations should be satisfied:

$$(X_1 - P_1)^2 + (Y_1 - Q_1)^2 = R^2 \tag{7.45}$$

$$(X_2 - P_1)^2 + (Y_2 - Q_1)^2 = R^2 \tag{7.46}$$

$$(X_3 - P_1)^2 + (Y_3 - Q_1)^2 = R^2 \tag{7.47}$$

$$(X_3 - P_2)^2 + (Y_3 - Q_2)^2 = R^2 \tag{7.48}$$

$$(X_4 - P_2)^2 + (Y_4 - Q_2)^2 = R^2 \tag{7.49}$$

$$(X_5 - P_2)^2 + (Y_5 - Q_2)^2 = R^2 \tag{7.50}$$

$$(X_6 - P_3)^2 + (Y_6 - Q_3)^2 = R^2 \tag{7.51}$$

$$(X_7 - P_3)^2 + (Y_7 - Q_3)^2 = R^2 \tag{7.52}$$

$$(X_8 - P_3)^2 + (Y_8 - Q_3)^2 = R^2 \tag{7.53}$$

Equation (2.2) takes the form of

$$X_i = a_i + p \cos \theta_i - q \sin \theta_i$$

$$Y_i = b_i + p \sin \theta_i + q \cos \theta_i \qquad i = 1,2, \ldots ,8 \tag{7.54}$$

Nine parameters, P_1, Q_1, P_2, Q_2, P_3, Q_3, p, q, and R, are involved in nine equations. Thus, the equations have no free choice of parameter.

Suppose nine prescribed positions are shown in Figure 7.7. The solution, if exists, should be at the intersection point of the circle point curves FP 321–345 and FP 321–678.

Plot circle points for FP 321–345 and FP 321–678 in Figure 7.7. A good circle point is found at the intersection point C_3. Locate C_1, C_2, C_4, C_5, C_6, C_7, and C_8 by geometric similarity.

The center S of the circle passing through circle points C_1, C_2, and C_3 is the center point for phase 1. The radius of the circle is the crank length R. Similarly, the center T of the circle passing through circle points C_3, C_4, and

C_5 is the center point for phase 2. Finally, the center U of the circle passing through circle points C_6, C_7, and C_8 is the center point for phase 3.

Figure 7.8 is an enlarged view at the intersection point C_3. In order to get a more precise solution, more points are plotted in the figure.

Chapter 8
Example Problems

8.1 Introduction

In the previous chapters, only one side of the adjustable linkage is discussed. In order to design a complete four-bar linkage, one more crank should be added.

Good points and solutions in the previous chapters are just solutions which satisfy the basic equations for a particular problem. A good solution for an adjustable linkage must also be free of order and branch defects.

Table 8.1 Example problems

ph.1	positions ph.2	ph.3	driven Side	driving side
1,2	3,4		Adj. MP	Adj. MP
1,2	3,4		Adj. MP	Not Adj.
1,2,3	4,5		Adj. MP	Adj. MP
1,2,3	4,5,6		Adj. MP	Adj. MP
1,2	3,4	5,6	Adj. CL	Adj. CL
1,2,3,4	5,6,7		Adj. MP & CL	Adj. MP & CL

AutoCAD along with AutoLISP allows the user to define their own functions and commands to meet their particular needs. Many user-defined AutoLISP functions and AutoCAD commands are developed to make the adjustable linkage design process automatic but on a flexible trial-and-error basis.

Six example problems listed in Table 8.1 are presented in this chapter.

8.2 Example 1

Four given positions are shown in Figure 8.1. This is the case of two positions in each of the two phases with no shared position. Let us try adjustable moving pivot on both driving and driven side.

As mentioned in section 3.4, there are three free choices of parameters for this problem. After choosing moving pivot C_1, call user-defined AutoLISP function TRIANG to plot a similar triangle to find C_2. Call user-defined AutoLISP function BISECT to plot a bisector for line segment C_1C_2. Draw a circle with center C_1 and the chosen crank length R_1 as the radius; this circle intersects the bisector at S, which is the center point. Invert point S from position 4 into position 3 to get point S_4 by using the user-defined AutoLISP function INVERT. Draw a circle passing through C_1 and C_2 with center point S. Draw another bisector for line segment SS_4; this bisector intersects the circle at E_3, which is the circle point at position 3 of phase 2. Locate E_4 by plotting similar triangles. No order defect has been found by inspection.

Suppose that we have worked on the driven side. The circle points for the driving side should be chosen properly so that no branch defect will occur, since the branch defect may occur even for two positions. A Turbo Pascal program and an user-defined AutoLISP command FILEMON has been developed for plotting the Filemon Construction Lines. Two groups of Filemon Lines starting at two distinct driven side moving pivots are required. The resulting Filemon Lines are shown in Figure 8.2. The driving side circle point D_1 at position 1 should not be chosen in the hatched area passing through C_1. After choosing circle point D_1, choose a crank length R_2 for the driving side, and repeat all the steps as have done for the driven side until

F_3, the circle point at position 3 is found (Figure 8.3). It can be seen from Figure 8.2 that F_3 is outside the shaded area passing through point E_3, which is necessary to avoid a branch defect for phase 2.

The resulting linkage is shown in four consecutive Figures 8.4 through 8.7. Neither a branch defect nor an order problem occurs although the first phase is a double- rocker. The data for this example problem are listed in Tables 8.2 and 8.3.

Table 8.2 The given data for example 1

Position	X	Y	θ
1	3.7000	4.0500	100.00
2	3.9200	6.1000	78.000
3	4.9300	8.0000	10.000
4	7.0469	8.2124	10.000

Table 8.3 The resulting data for example 1

Point	X	Y
C_1	2.5576	4.3097
D_1	2.2555	5.0548
E_3	4.9733	8.5731
F_3	6.3332	10.2901
S	6.4064	4.9450
T	7.9507	4.8247

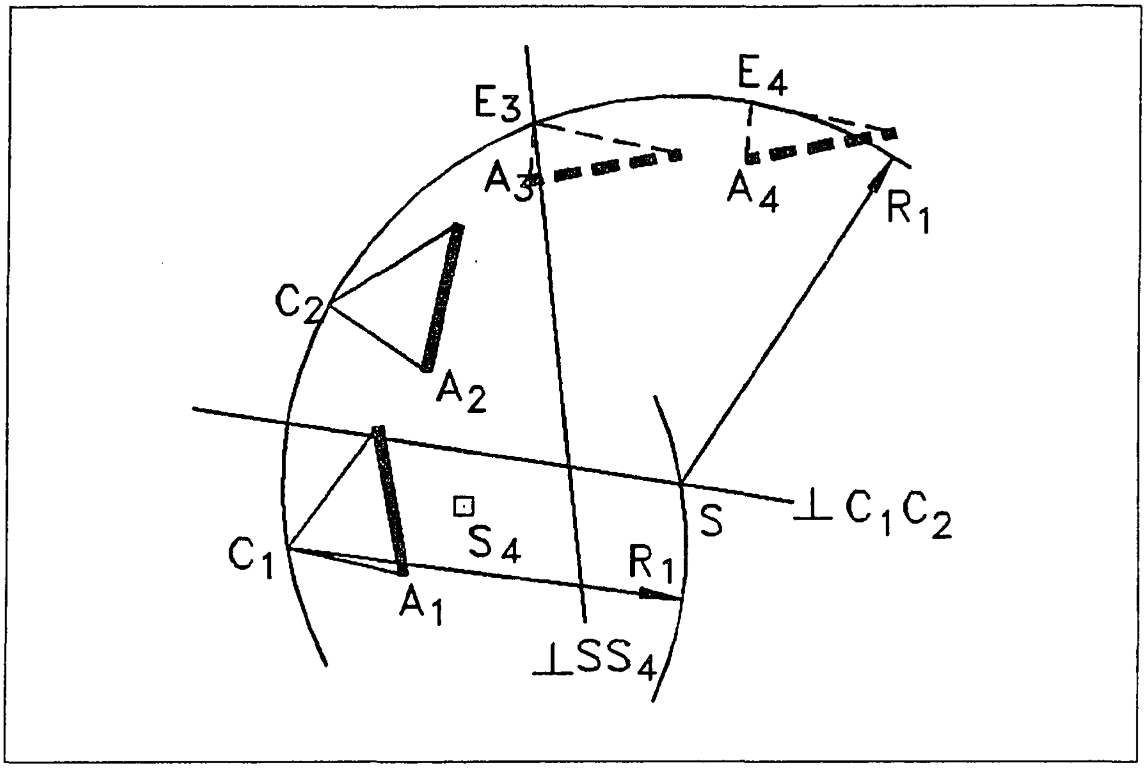

Figure 8.1 The given positions and the driven side for example 1

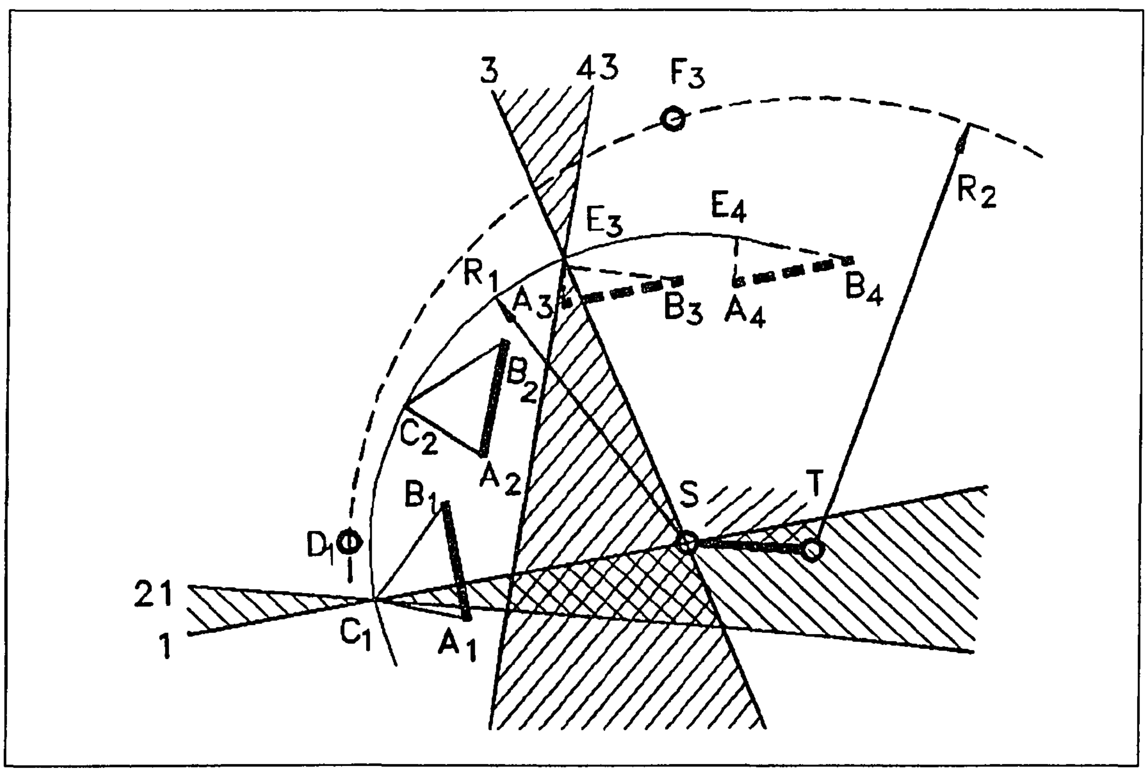

Figure 8.2 The Filemon Construction Lines for example 1

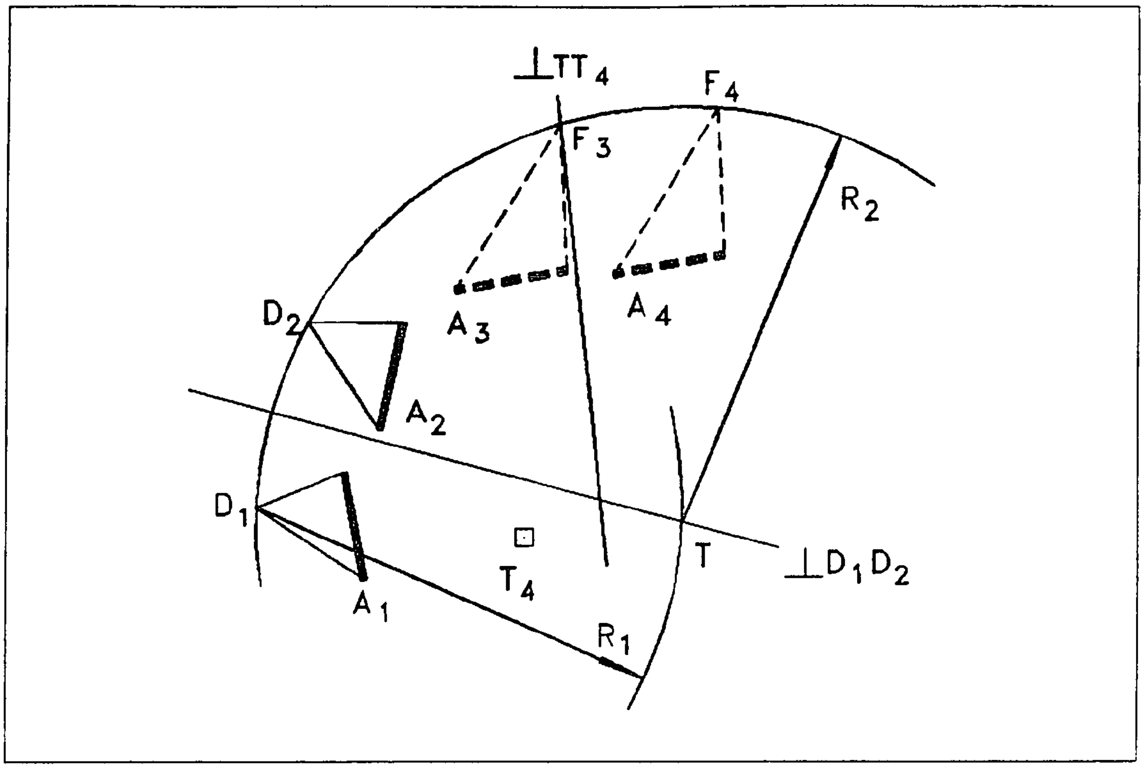

Figure 8.3 The driving side for example 1

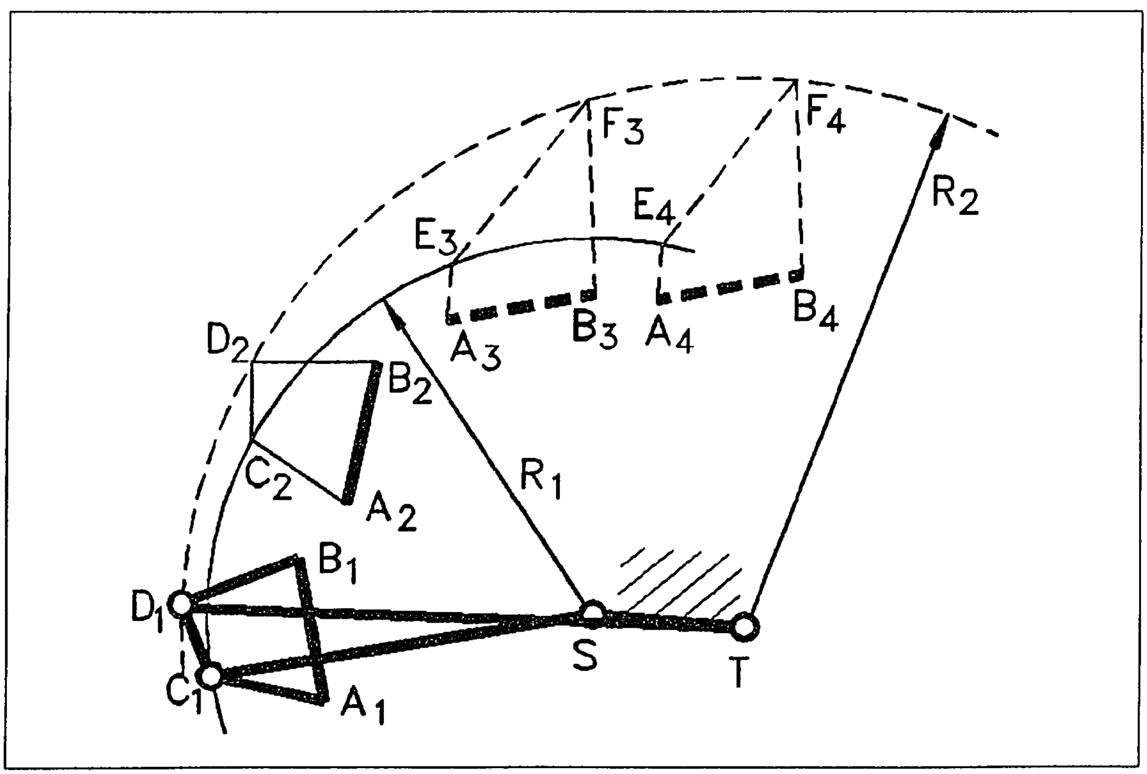

Figure 8.4 The resulting linkage at position 1

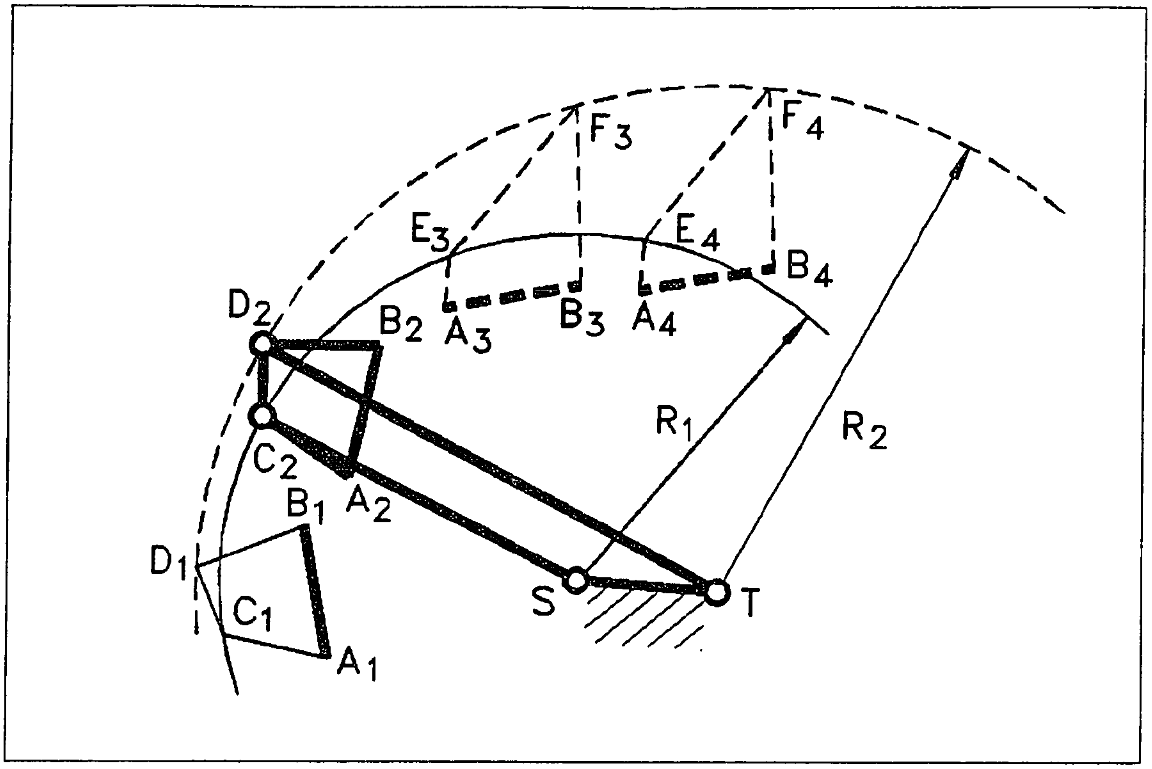

Figure 8.5 The resulting linkage at position 2

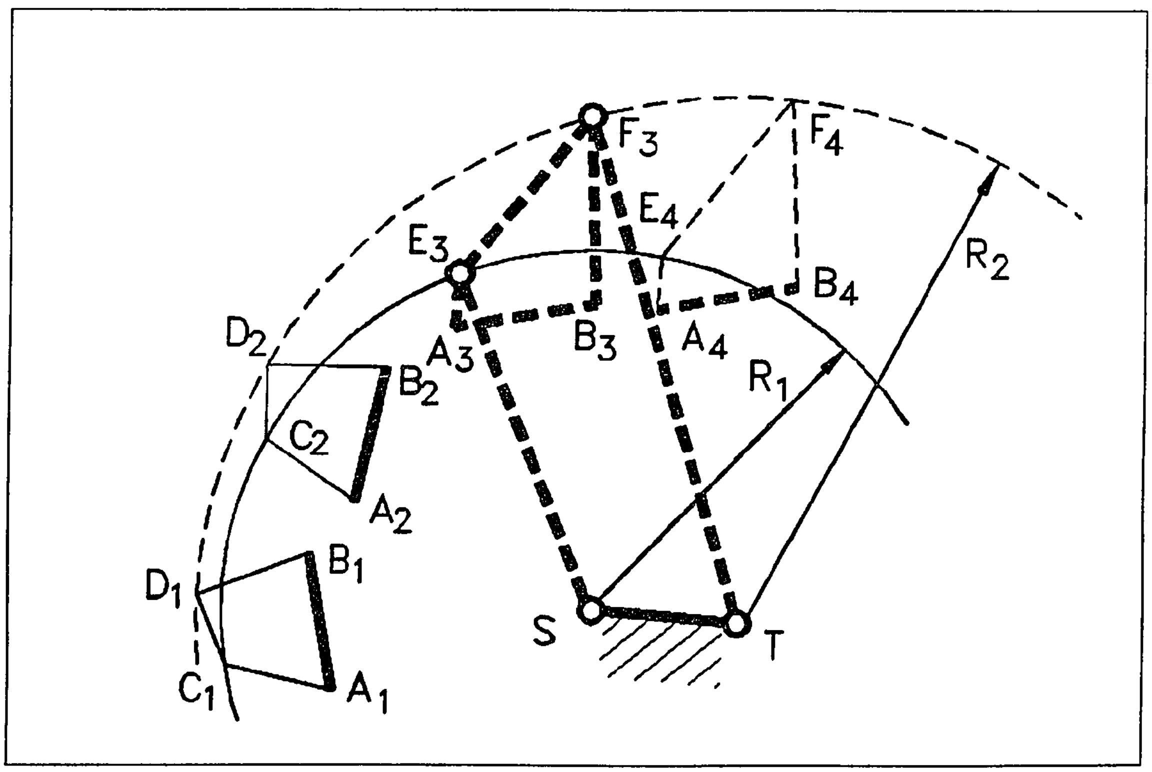

Figure 8.6 The resulting linkage at position 3

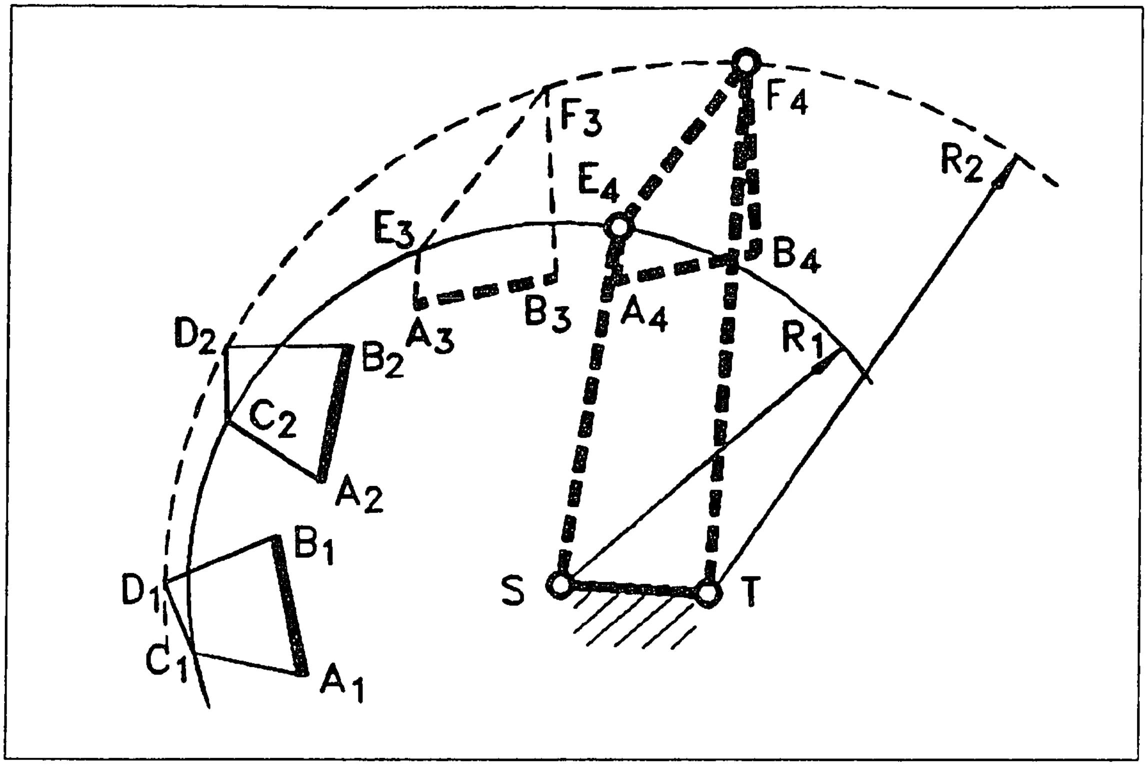

Figure 8.7 The resulting linkage at position 4

8.3 Example 2

This example needs two positions in each of the two phases. Four given positions are the same as that for example 1. The work on the driven side is the same as that we have done for example 1. The driving side of the four-bar linkage could be made not adjustable.

As shown in Figure 8.8, the circle points C_1, E_3, and the Filemon Construction Lines passing through them are the same as that for example 1. Plot the circle point curve by means of CIRC_PT.PAS [9]. The points in the shaded area shown in Figure 8.8 can not be used for the circle points because of the branch problem. Try point D_1 as the circle point for the driving side. Locate the corresponding points D_2, D_3, and D_4 by calling user–defined AutoLISP function TRIANG. The center point T for the driving side is at the center of the circle passing through points D_1, D_2, D_3, and D_4 (Figure 8.9).

The resulting adjustable four-bar linkage is shown in four consecutive Figures 8.10 through 8.13. Neither an order problem nor a branch defect occurs in this linkage. The resulting data are listed in Table 8.4.

Table 8.4 The resulting data for example 2

point	X	Y
C_1	2.5576	4.3097
D_1	3.0521	4.6673
E_3	4.9733	8.5731
S	6.4064	4.9450
T	6.9850	4.9744

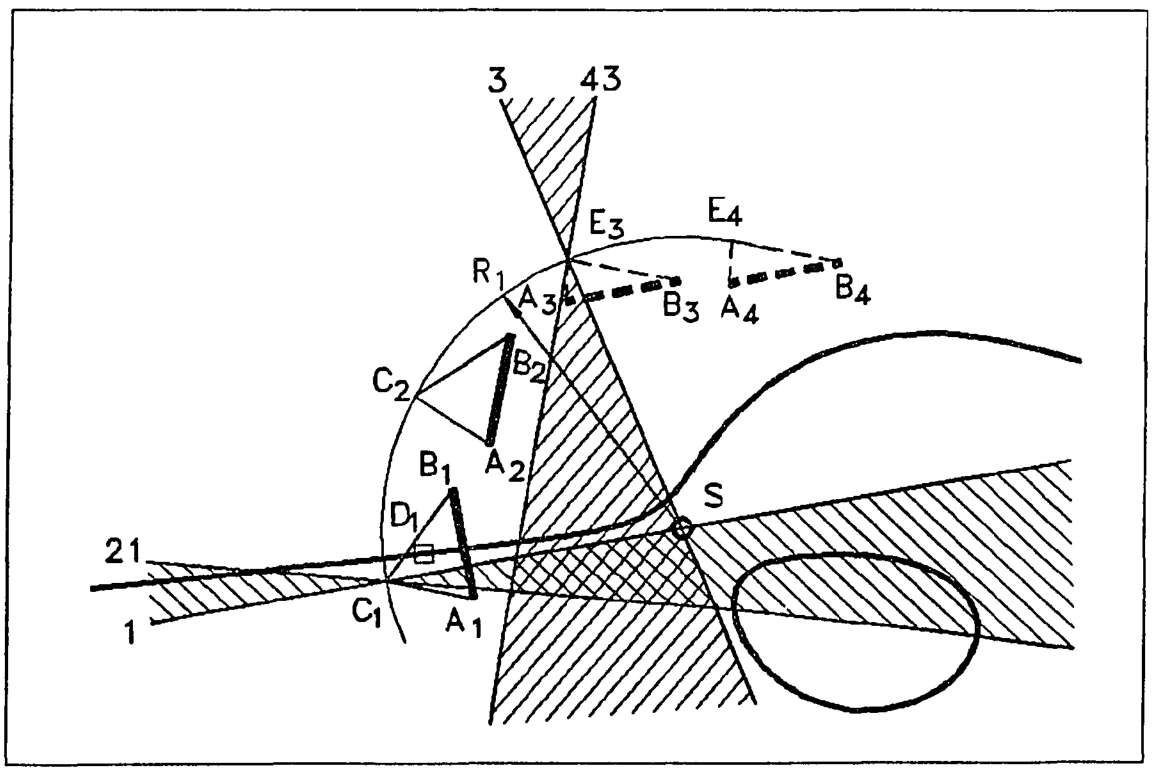

Figure 8.8 The Filemon Construction Lines and the circle point curve for example 2

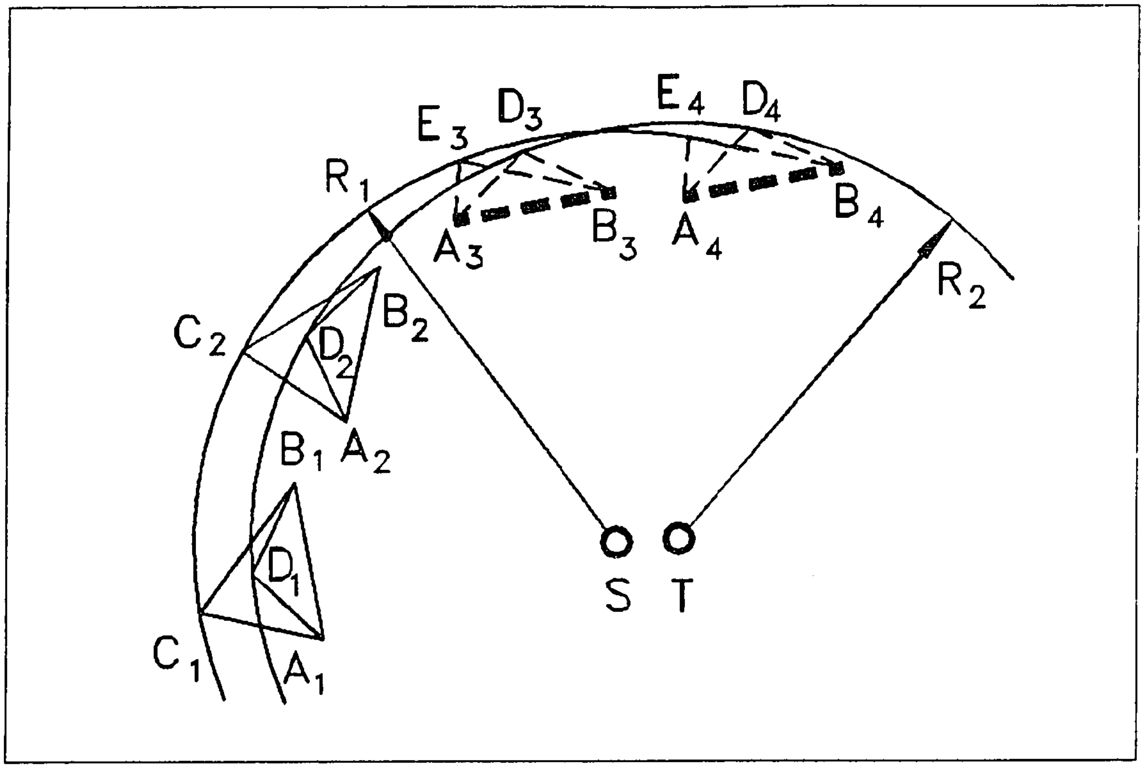

Figure 8.9 The driving side for example 2

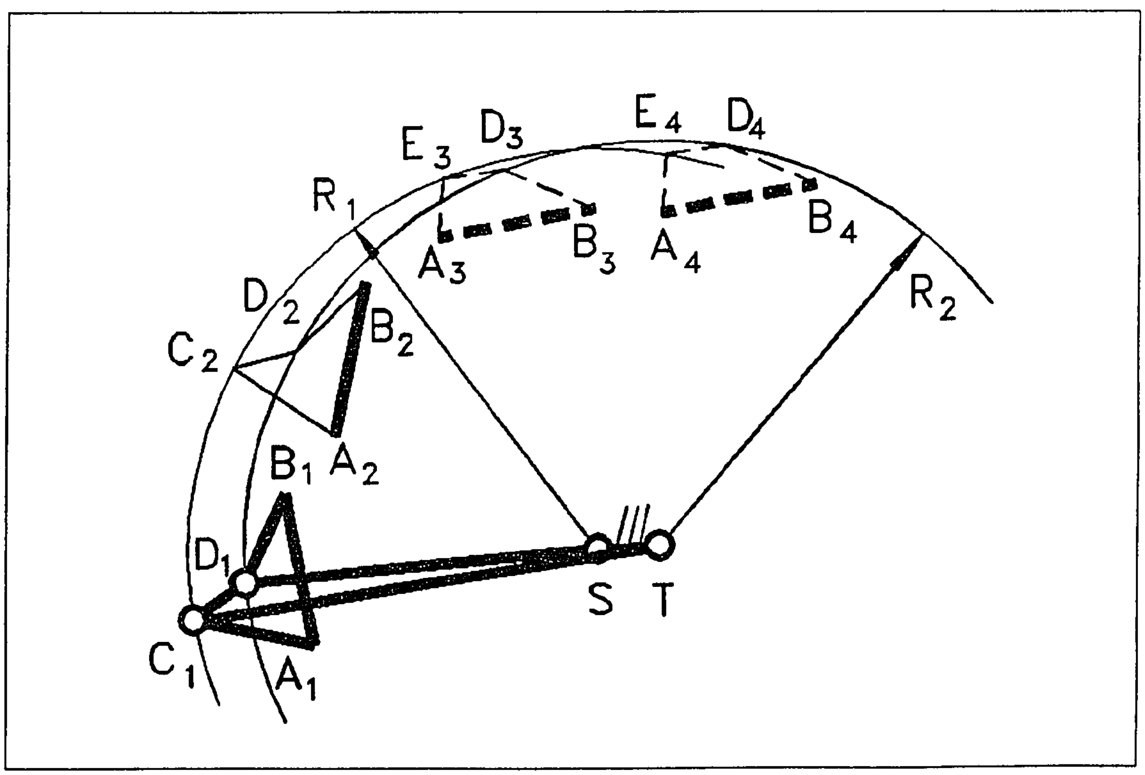

Figure 8.10 The resulting linkage at position 1

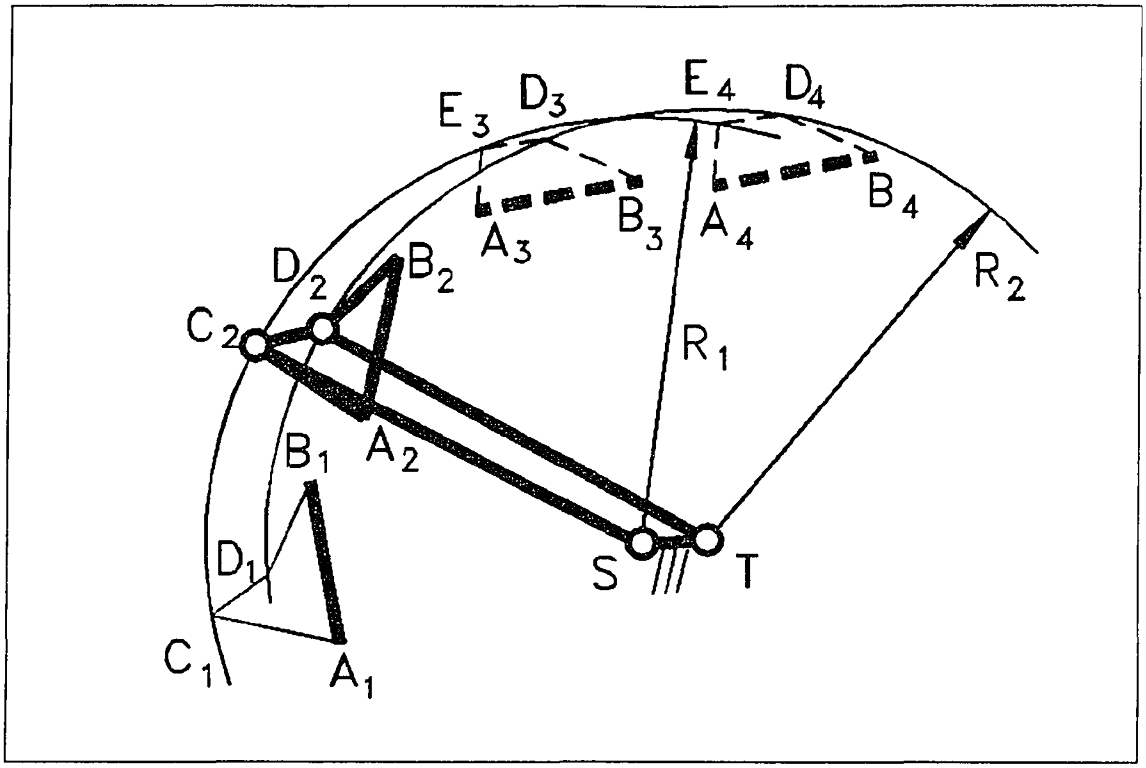

Figure 8.11 The resulting linkage at position 2

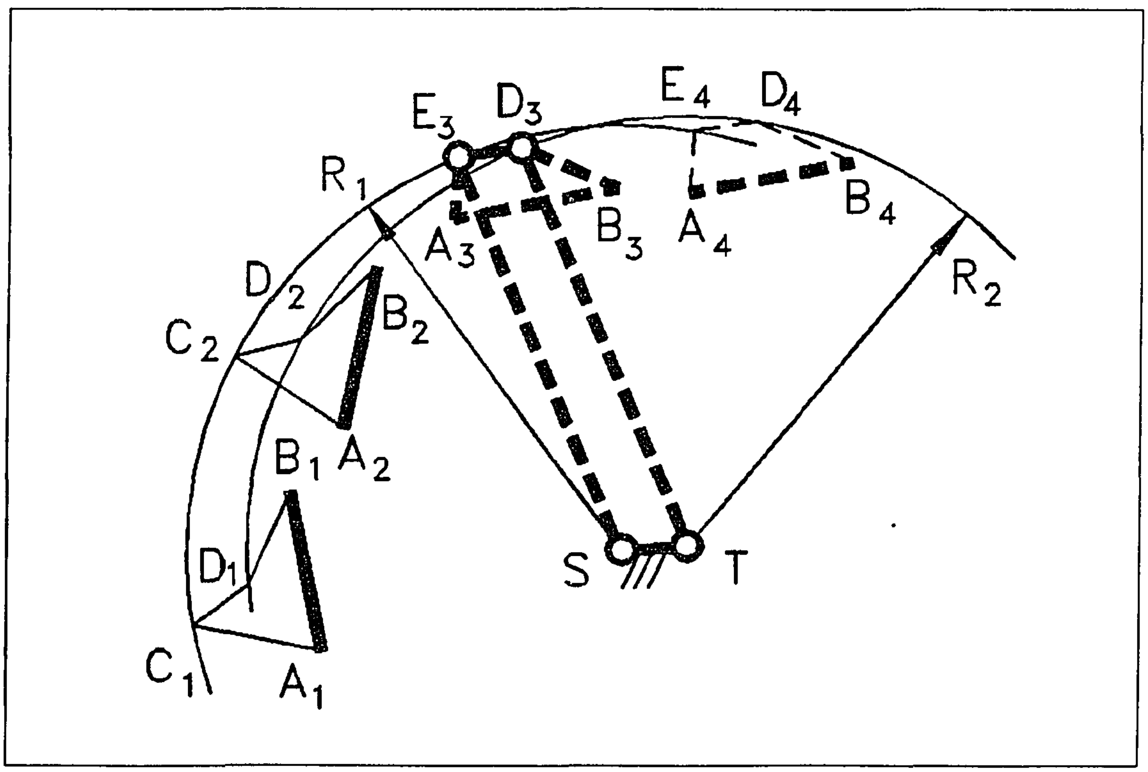

Figure 8.12 The resulting linkage at position 3

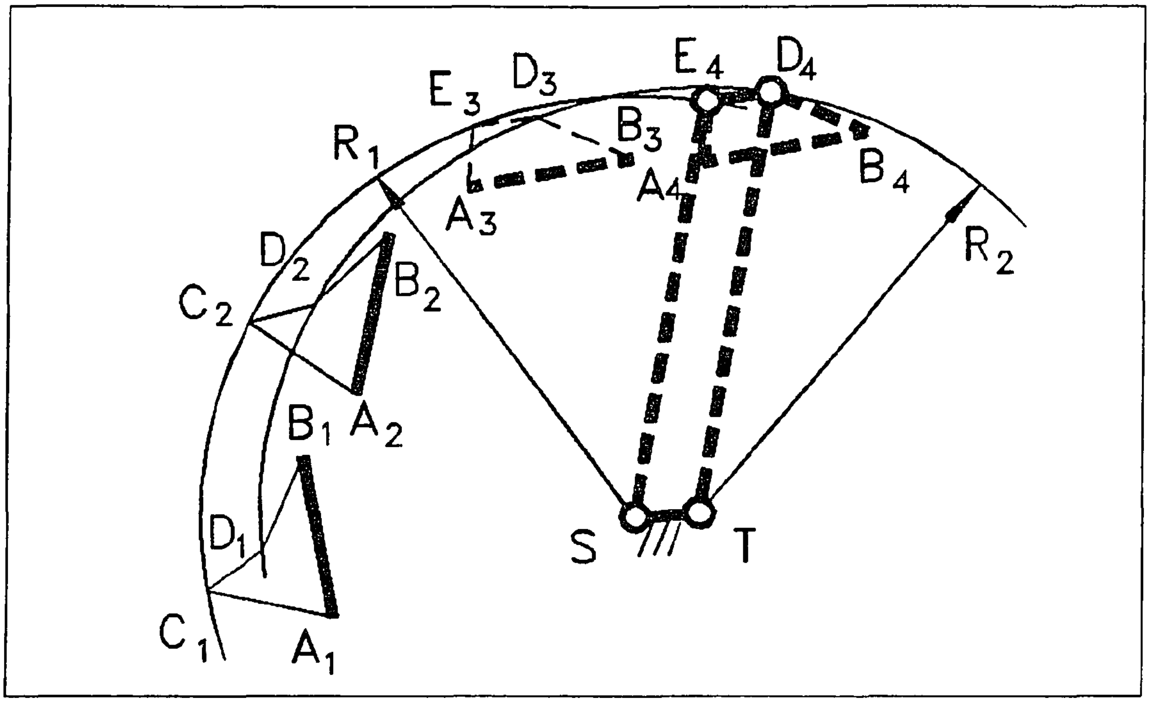

Figure 8.13 The resulting linkage at position 4

8.4 Example 3

Five given positions are shown in Figure 8.15. The problem requires 3 positions in phase 1, and 2 positions in phase 2 with no position shared. Suppose adjustable moving pivot on both driven and driving side is desired.

Plot Waldron Image Pole Circles for the first phase as shown in the figure. The circle point C_1 is chosen outside the shaded area, and the circle points C_2 and C_3 are drawn by using the user-defined AutoLISP function TRIANG. The center S of the circle passing through C_1, C_2, and C_3, is the center point. Invert S from position 5 into position 4 to get point S_5 by calling the user-defined AutoLISP function INVERT. The circle point at position 4, E_4 can be found by intersecting the bisector for the line segment SS_5 and the circle passing through C_1, C_2, and C_3. Thus, the work on the driven side is done.

In order to avoid a branch defect, two groups of Filemon Construction Lines are drawn in Figure 7.14 by using the user-defined AutoLISP command FILEMON for phases 1 and 2 respectively. The circle point D_1 at position 1 should be chosen outside the shaded area, the border lines of which pass through C_1. Similarly, the circle point F_4 at position 4 can not be chosen inside the hatched region, the border lines of which originate at point E_4.

Choose D_1 and do everything as that for the driven side to find F_4. The location of point F_4 is checked to be outside the shaded area.

The resulting adjustable four-bar linkage is shown is 5 consecutive Figures 8.16 through 8.20. Neither a branch defect nor an order problem occurs in the resulting linkage although phase 2 is a double–rocker. The given data and the resulting data are listed in Tables 8.5 and 8.6 respectively.

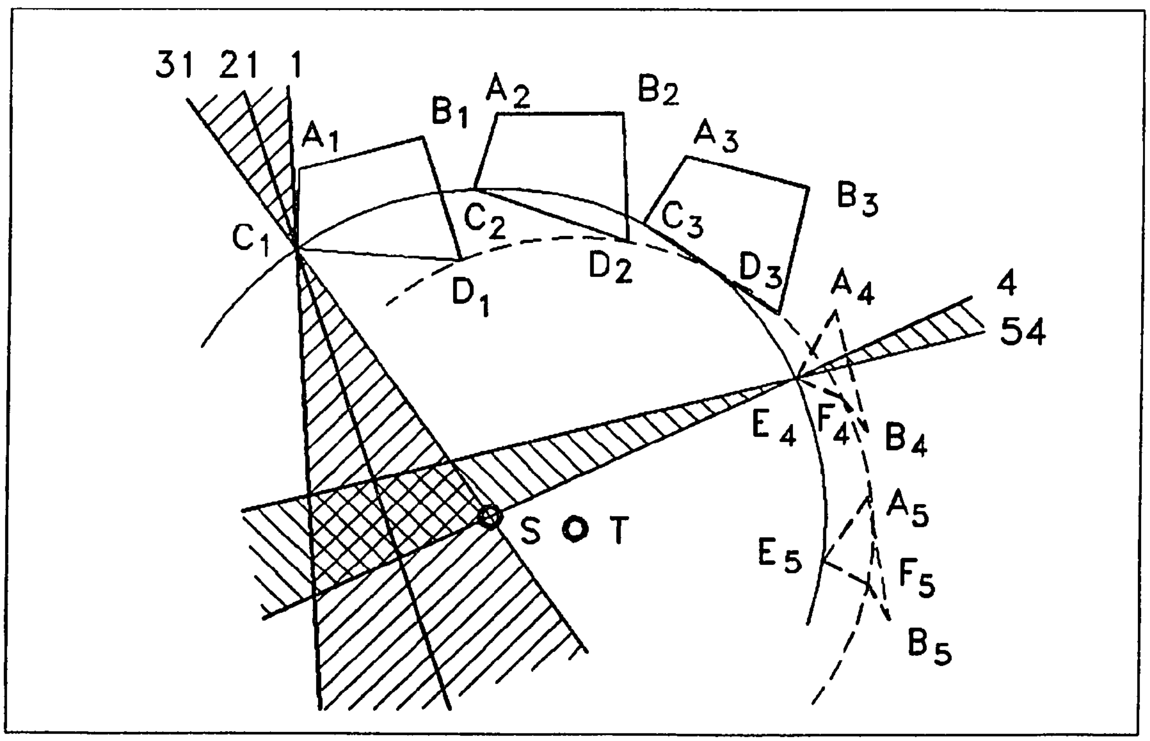

Figure 8.14 The Filemon Construction Lines and the moving pivots on the driving side for example 3

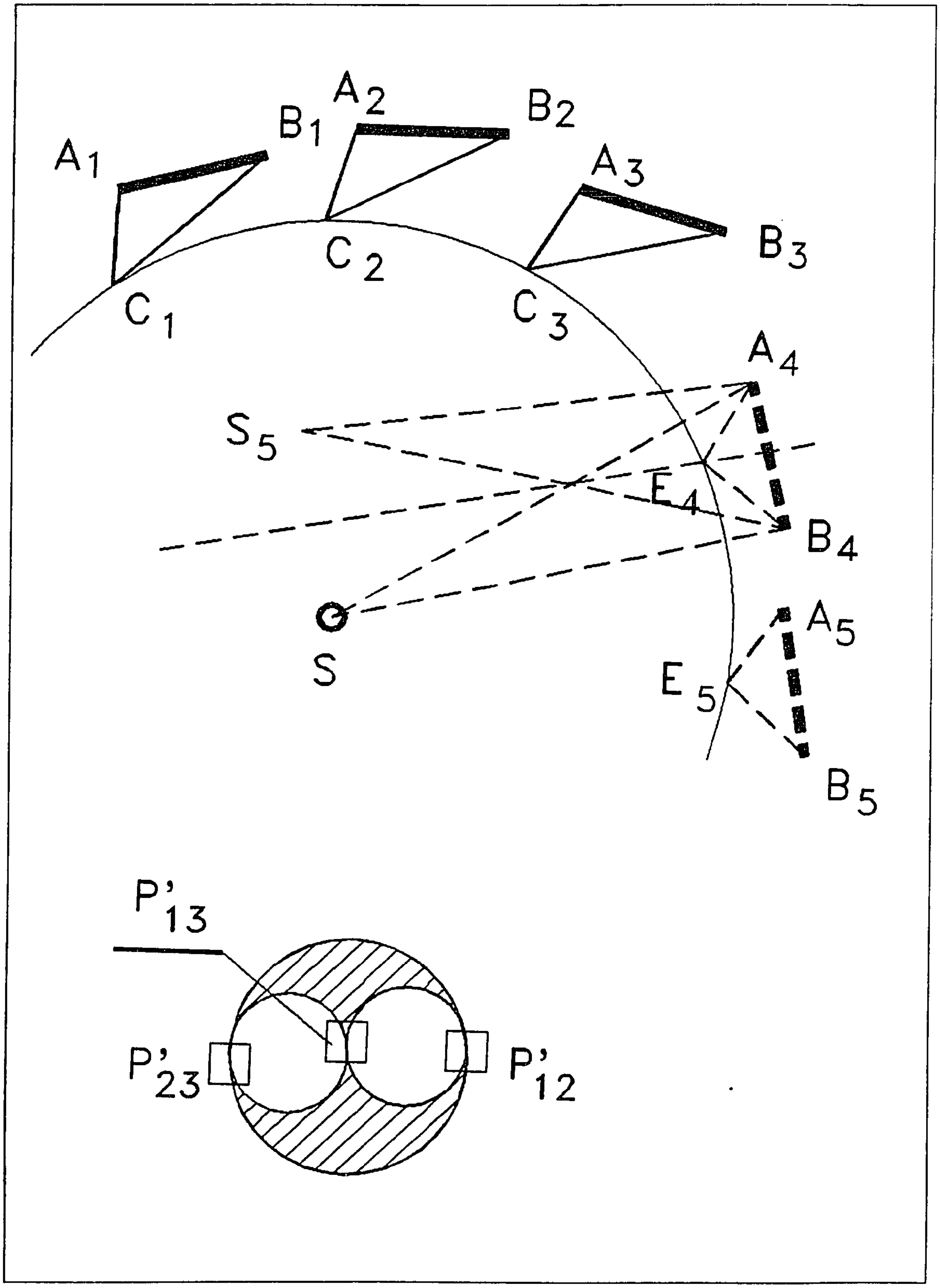

Figure 8.15 The moving pivots for the driven side and the Waldron Image Pole Circles for example 3

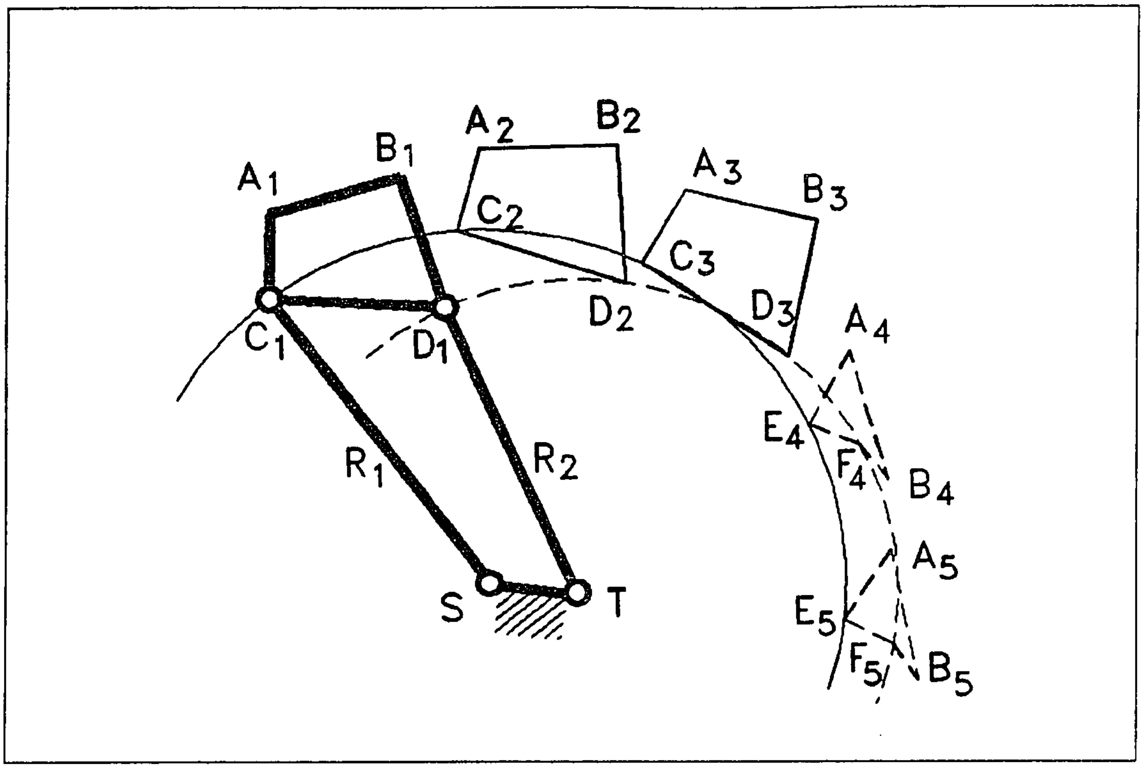

Figure 8.16 The resulting linkage at position 1

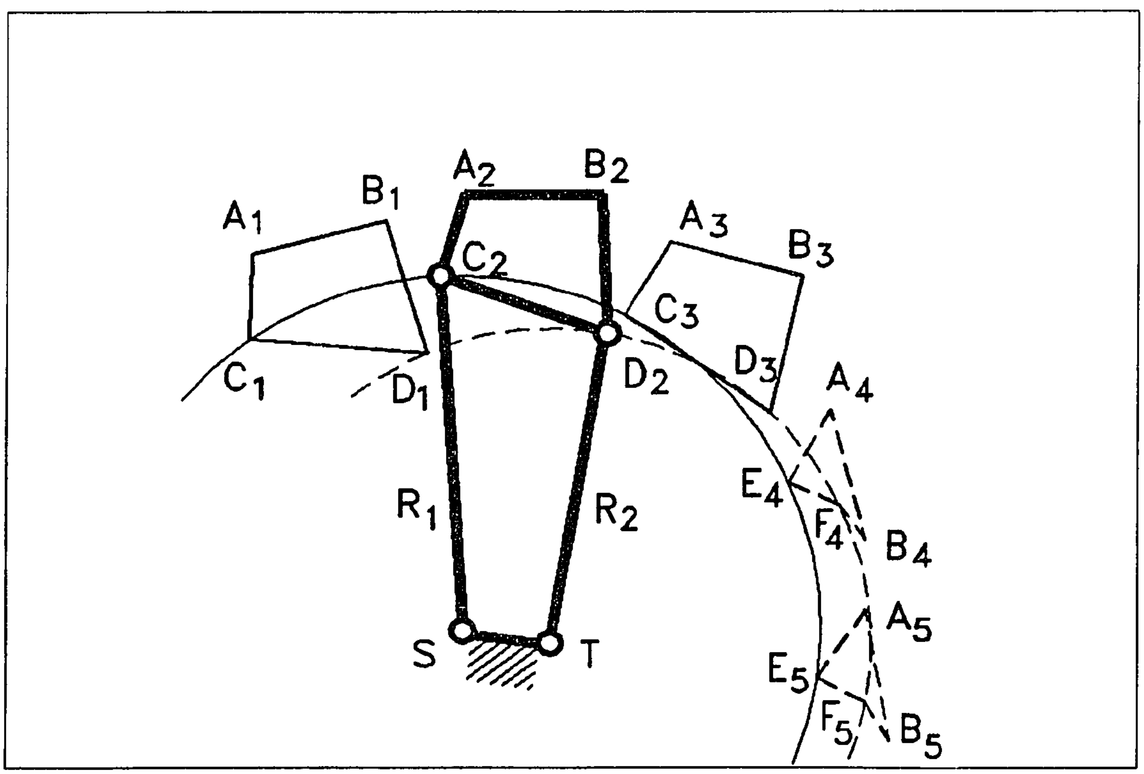

Figure 8.17 The resulting linkage at position 2

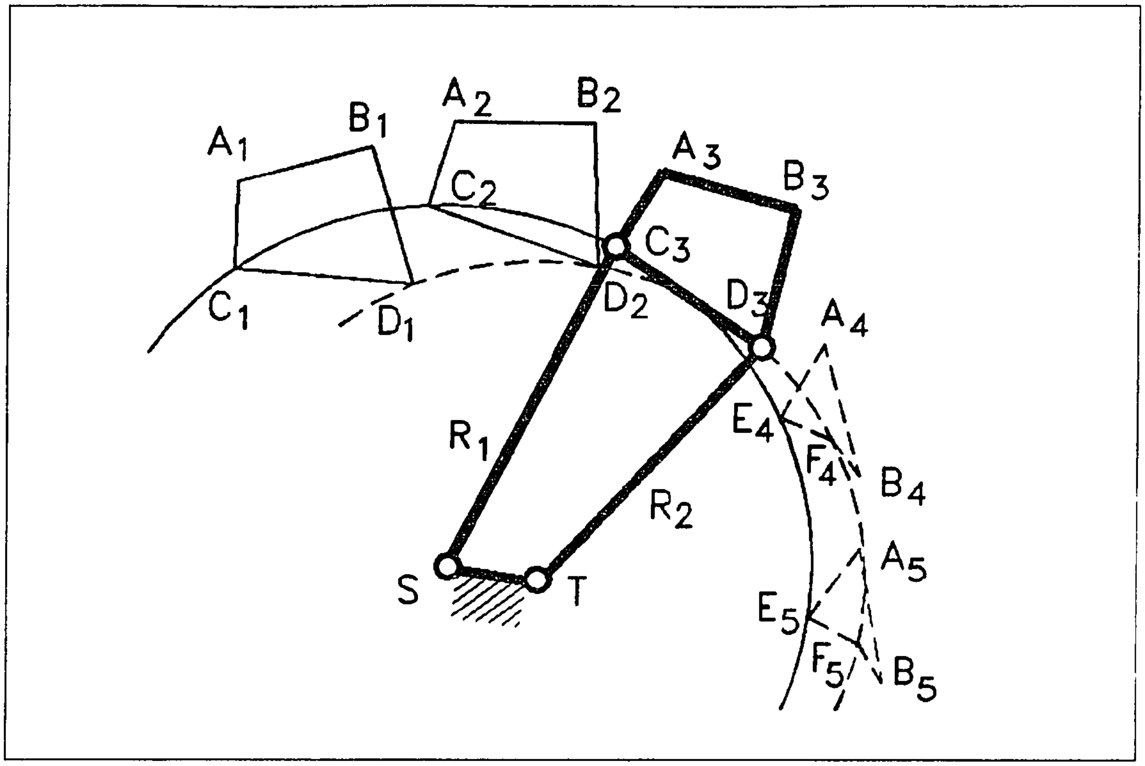

Figure 8.18 The resulting linkage at position 3

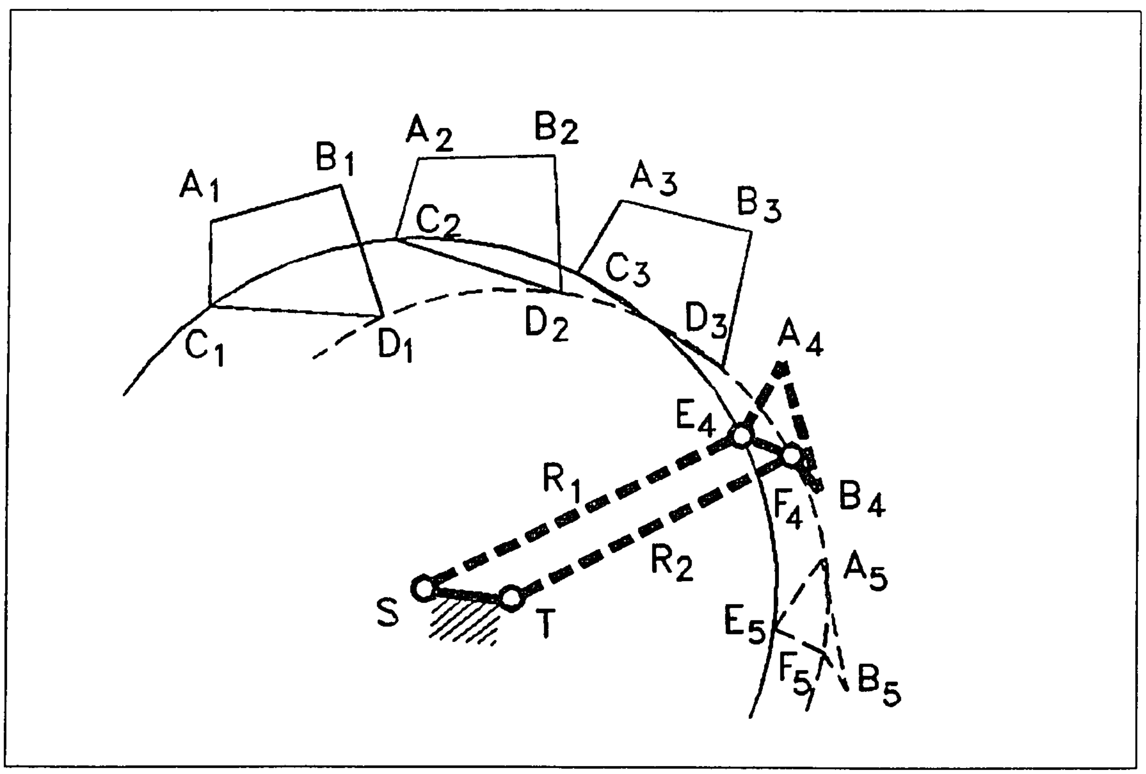

Figure 8.19 The resulting linkage at position 4

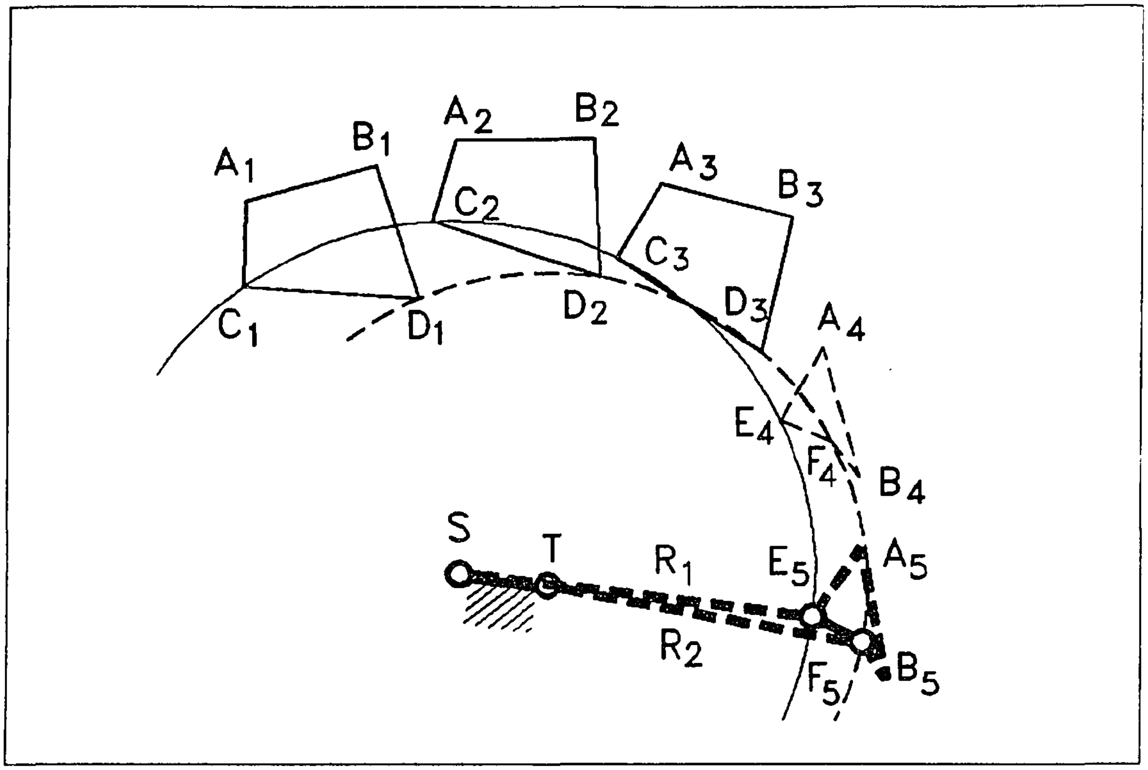

Figure 8.20 The resulting linkage at position 5

Table 8.5 The given data for example 3

Position	X	Y	θ
1	2.3800	7.8500	15.000
2	4.7100	8.5300	0.0000
3	6.9500	8.0000	345.00
4	8.7200	6.1200	285.00
5	9.1200	3.8700	280.00

Table 8.6 The resulting data for example 3

Point	X	Y
C_1	2.3543	6.8908
D_1	4.2867	6.7408
E_4	8.2521	5.3084
F_4	8.8139	5.0674
S	4.6430	3.6390
T	5.6573	3.4881

8.5 Example 4

This problem needs 3 positions on each of the two phases with no position shared as shown in Figure 8.21. An adjustable four-bar linkage is considered because the total number of unique positions is greater than 5.

Let us try adjustable moving pivot for both driving and driven side. Six given positions are drawn in Figure 8.22 by calling an user-defined AutoLISP function PLOT_POS. The center points satisfying equations (3.33) through (3.50) are plotted in the figure by using the Turbo Pascal program MP_3_3.PAS and an user-defined AutoLISP command PLOT_PTS.

Both Waldron Image Pole Circles and Filemon Construction Lines are for choosing circle points to avoid a branch problem. But the points in Figure 8.21 are center points on which none of the two methods apply. However, the program MP_3_3.PAS writes to output files not only the coordinates of center points but also that of their corresponding circle points.

Each center point for the MP_3_3 problem has two relating circle points, one for position 1 of phase 1, and the other for position 4 of phase 2.

The circle points for phase 1 are shown in both Figures 8.22 and 8.23, while that for phase 2 are plotted in both Figures 8.24 and 8.25 by mean of the user-defined AutoLISP function PLOT_PTS.

Two groups of Waldron Image Pole Circles are shown in Figures 8.22 and 8.24 for two different phases by means of the user-defined AutoCAD command IPOL_CIRC. Two groups of Waldron Circles are needed because the moving pivot varies in two phases.

For the same reason, two different sets of Filemon Construction Lines are plotted in Figures 8.23 and 8.25 by using another user–defined AutoCAD command FILEMON. Both IPOL_CIRC and FILEMON are implemented in AutoLISP and Turbo Pascal.

Two points, S and T, are chosen as the center points for the driving and the driven side respectively. Circle points C_1 and E_4 are relating to center point S, while circle points D_1 and F_4 are corresponding to center point T. The circle point for the driven side, D_1 is chosen outside the shaded area of the Image Pole Circles shown in Figure 8.22, while the circle point for the driving side, C_1 is chosen outside the shaded area of the Filemon Construction Lines in Figure 8.23. Similarly, the driven side circle point F_4 for the second phase is chosen outside the hatched region of the Image Pole Circles shown in Figure 8.24, and the driving side circle point E_4 for the second phase is chosen outside the hatched area of the Filemon Construction Lines shown in Figure 8.25.

The resulting four-bar linkage is shown in six consecutive Figures 8.26 through 8.31. Neither a branch defect nor an order problem occurs in this drag–link. In fact, either side could be the driving side for this particular four–bar linkage.

The data for this example are listed in Tables 8.7 and 8.8.

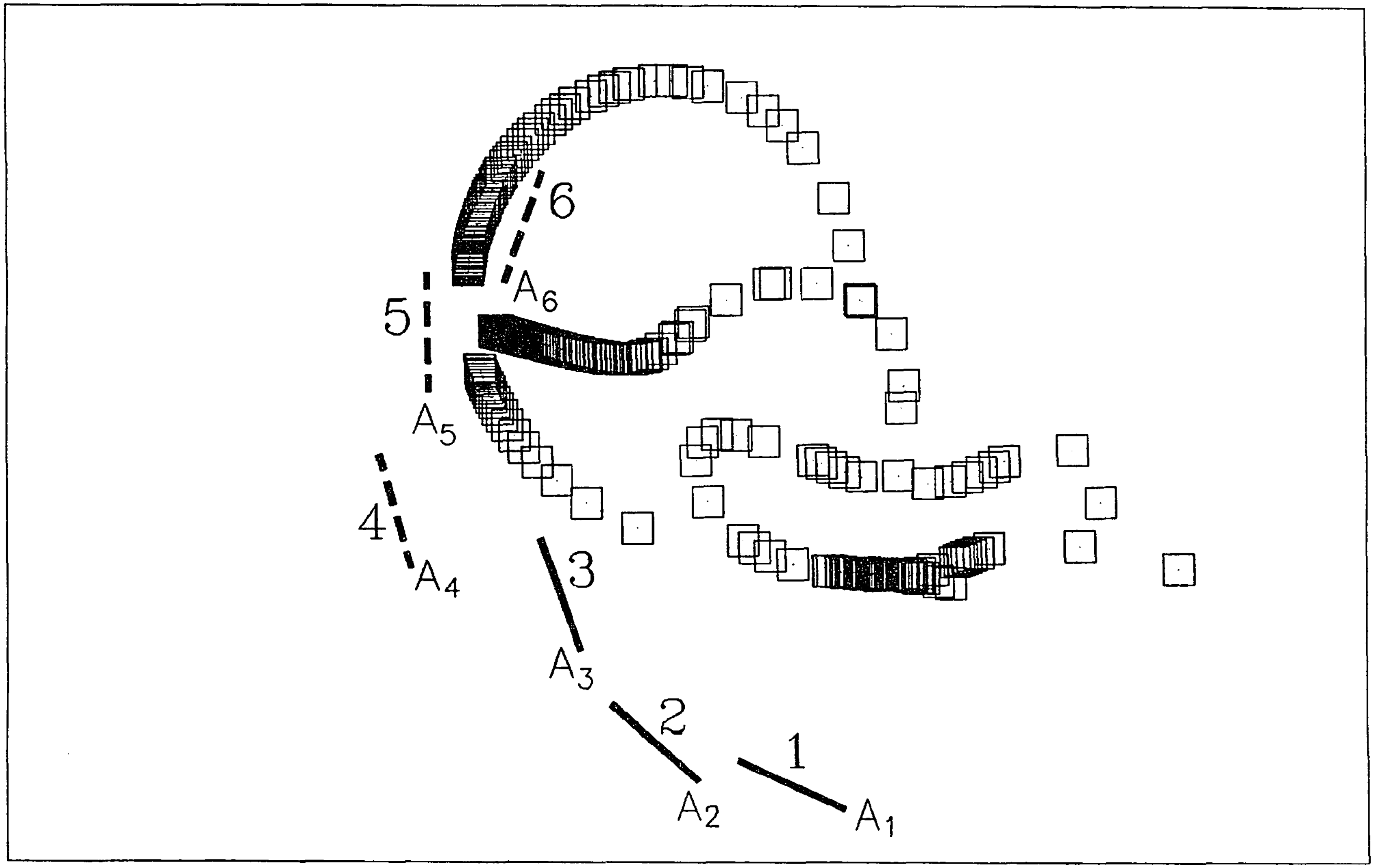

Figure 8.21 The given positions and the center points

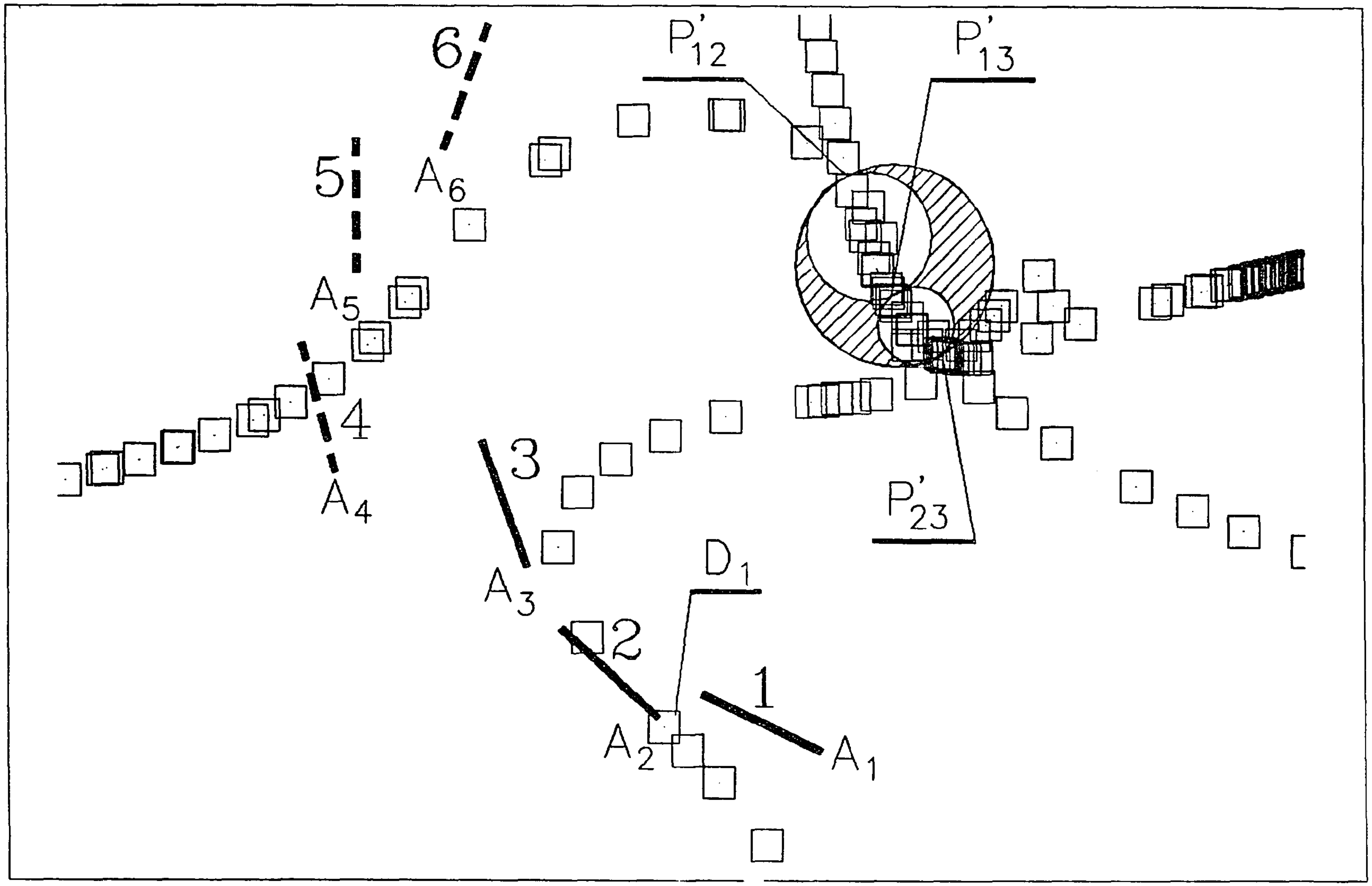

Figure 8.22 The Image Pole Circles for phase 1

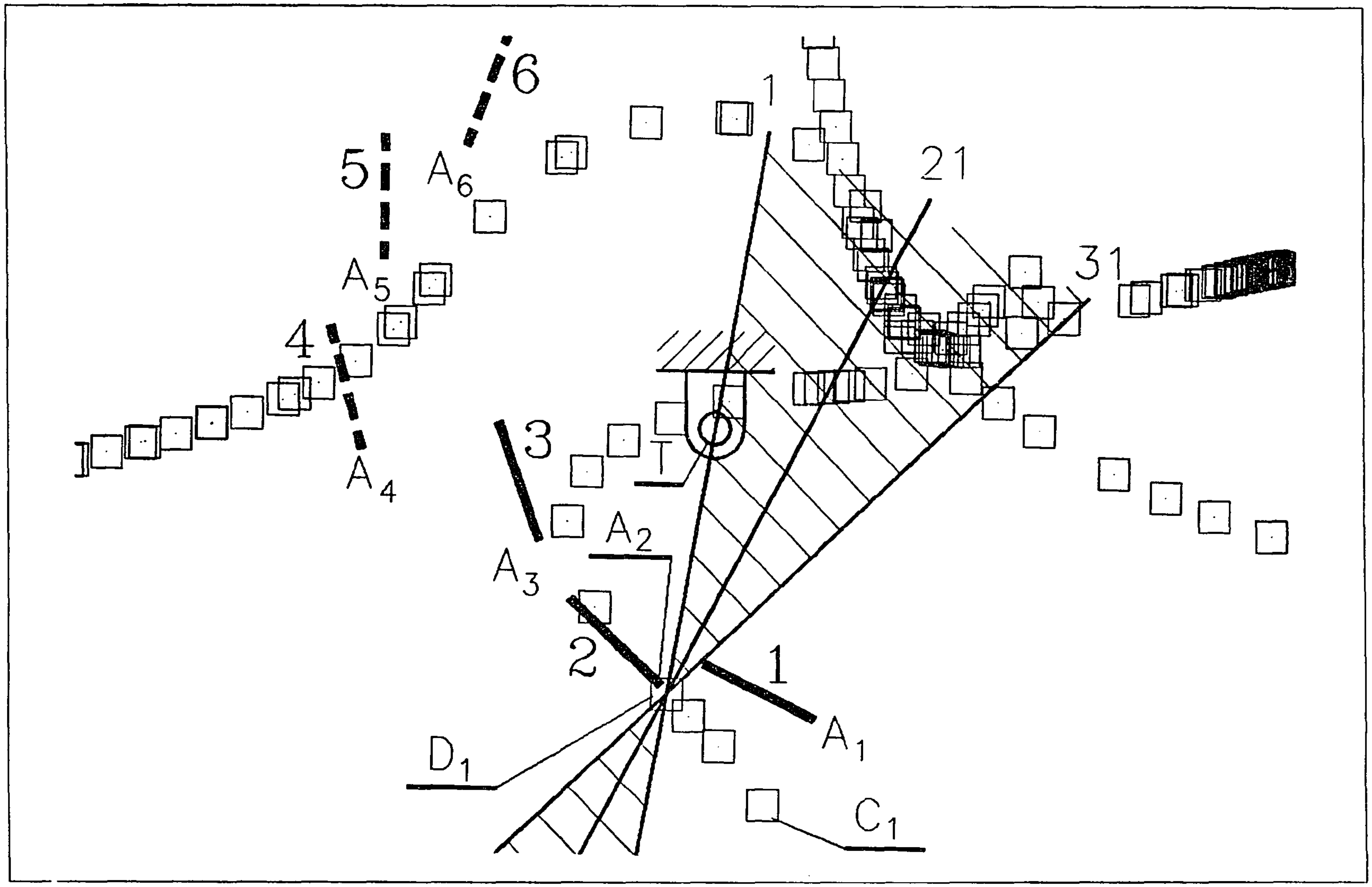

Figure 8.23 The Filemon Construction Lines for phase 1

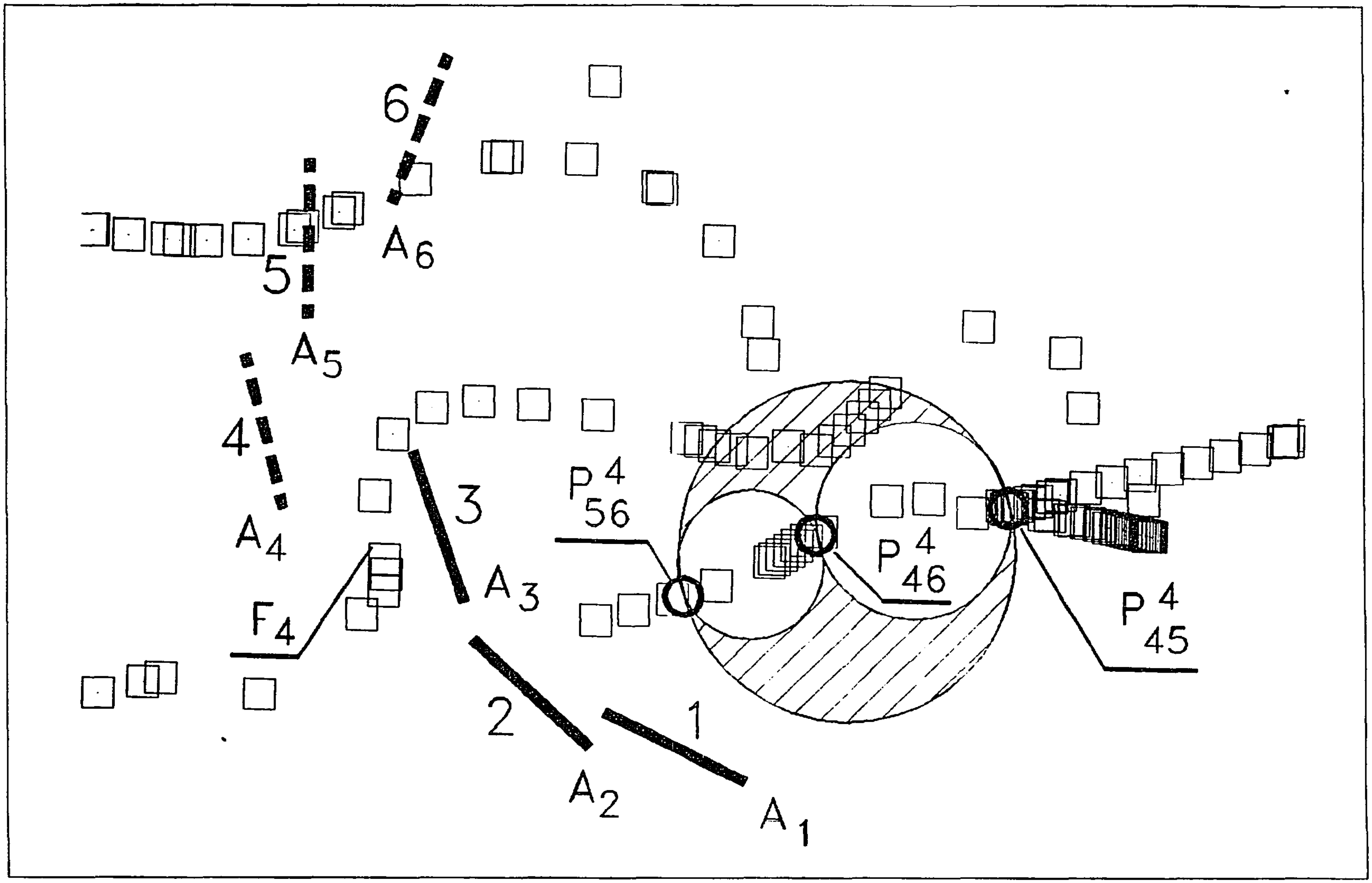

Figure 8.24 The Image Pole Circles for phase 2

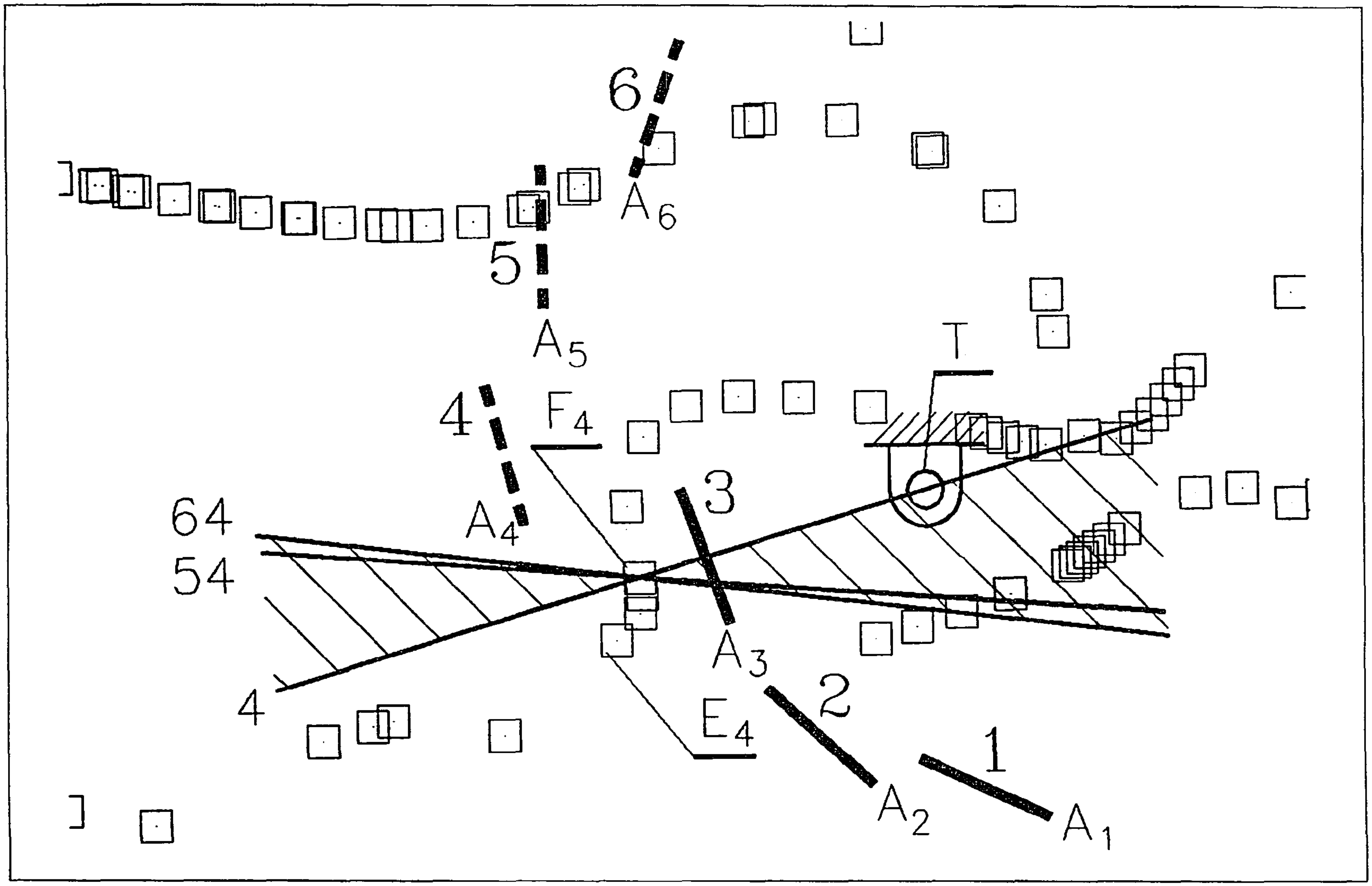

Figure 8.25 The Filemon Construction Lines for phase 2

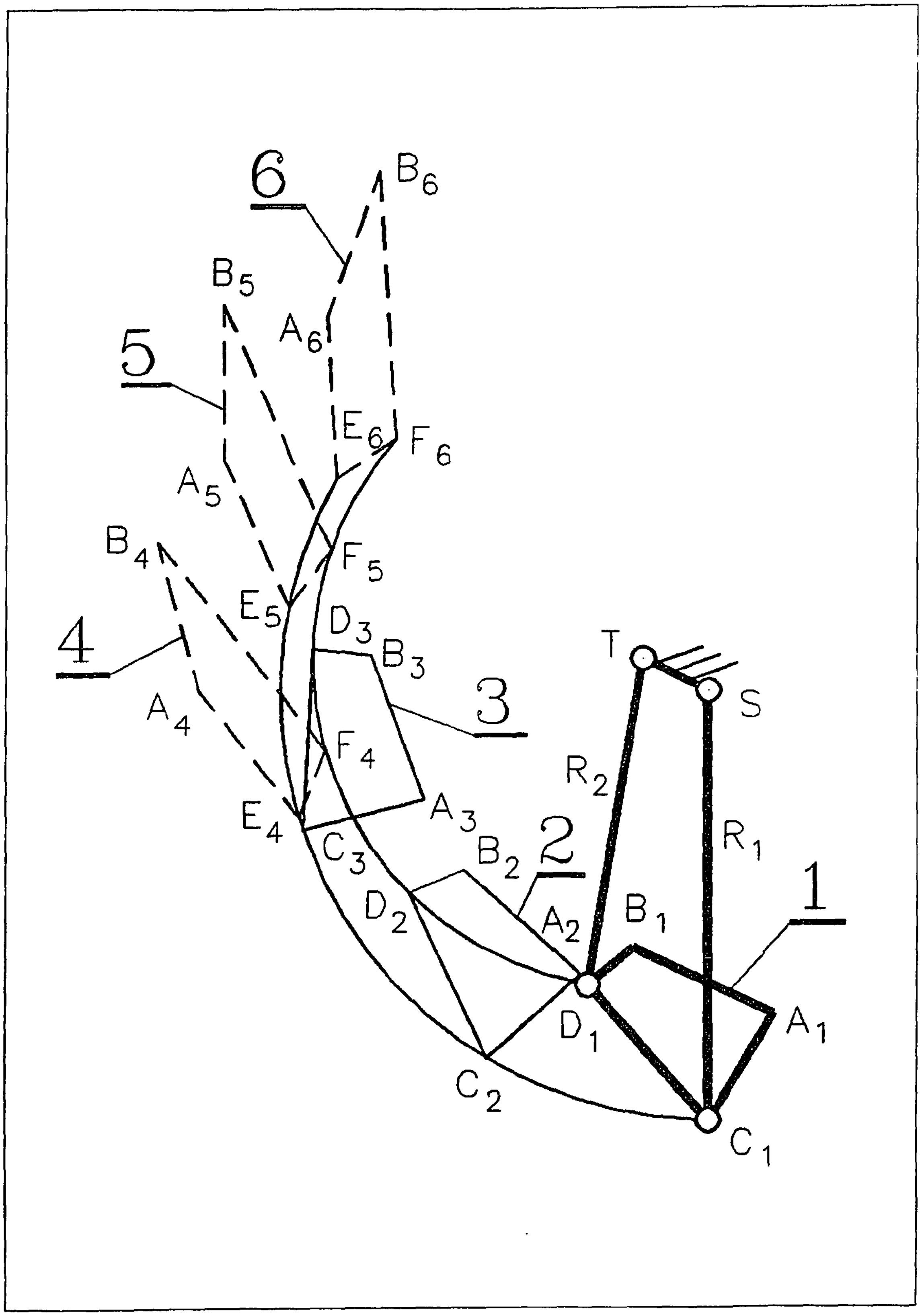

Figure 8.26 The resulting linkage at position 1

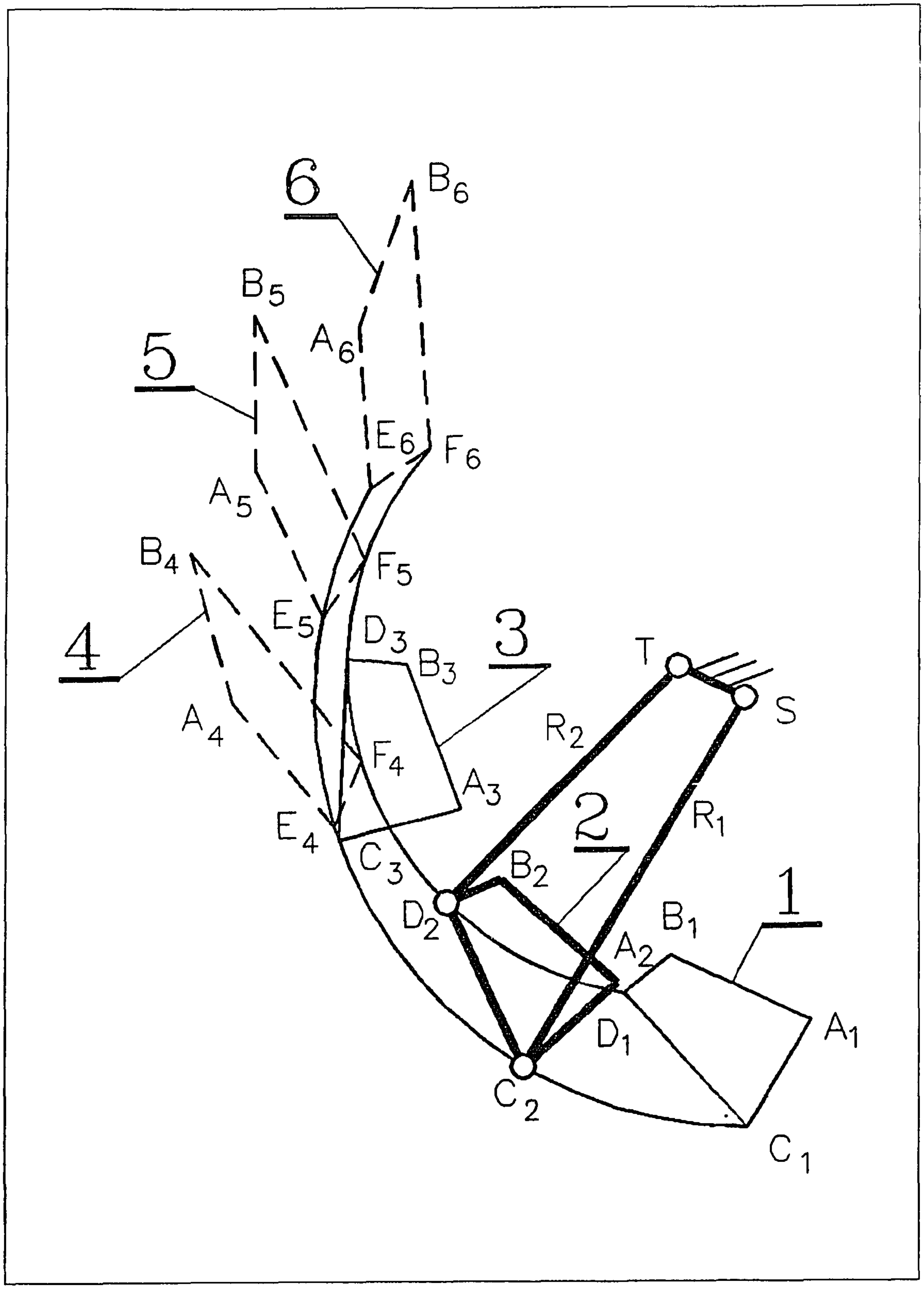

Figure 8.27 The resulting linkage at position 2

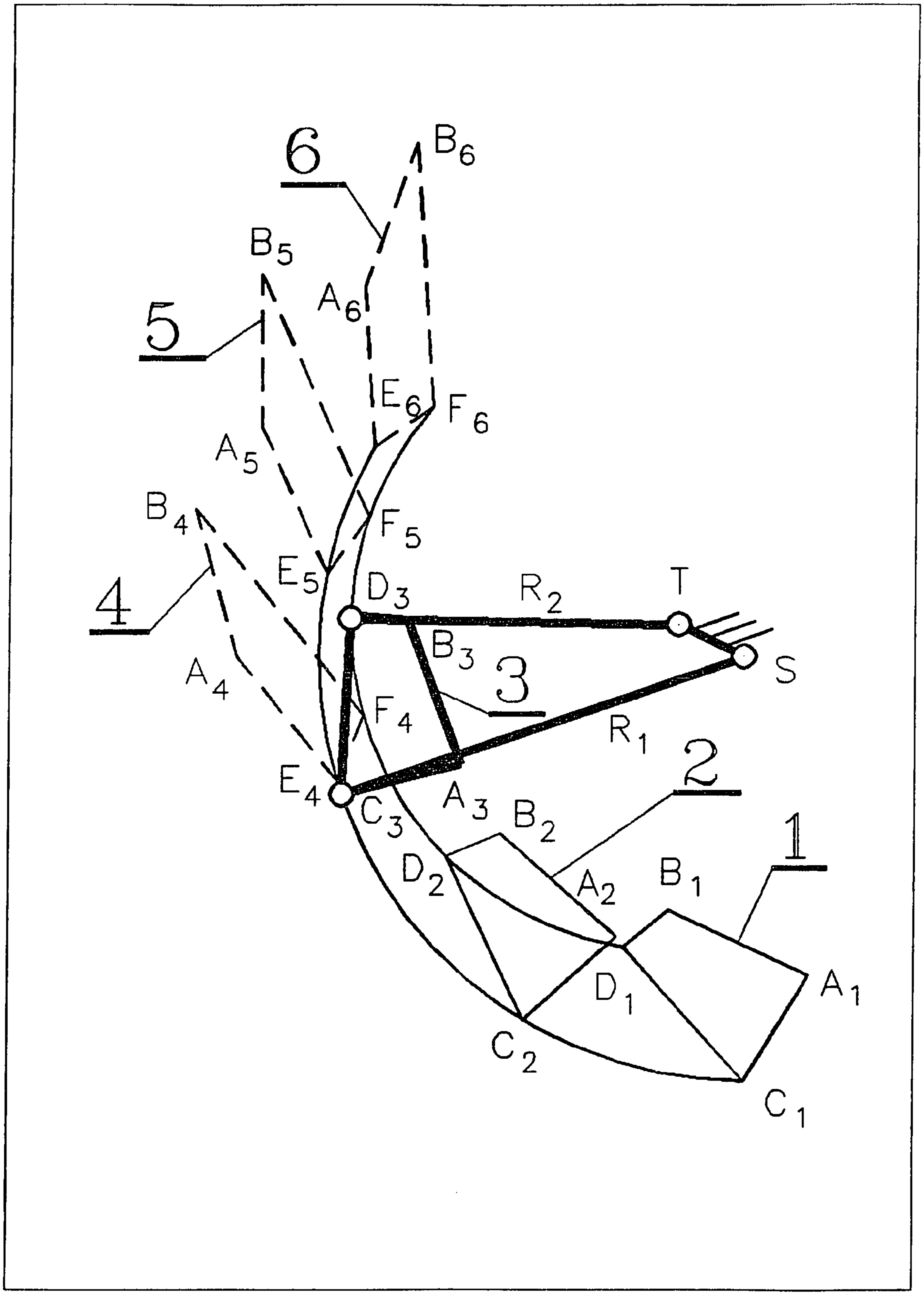

Figure 8.28 The resulting linkage at position 3

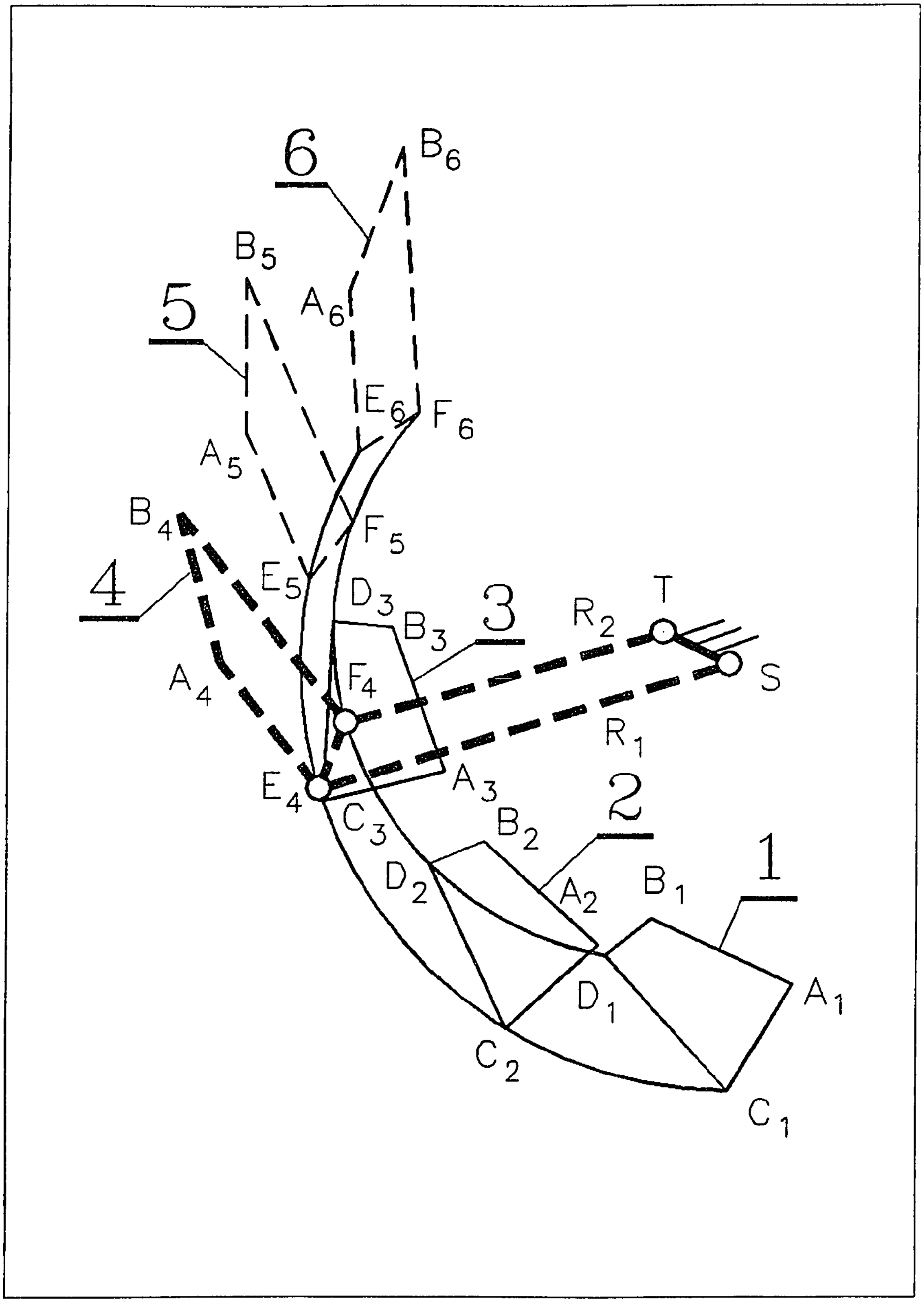

Figure 8.29 The resulting linkage at position 4

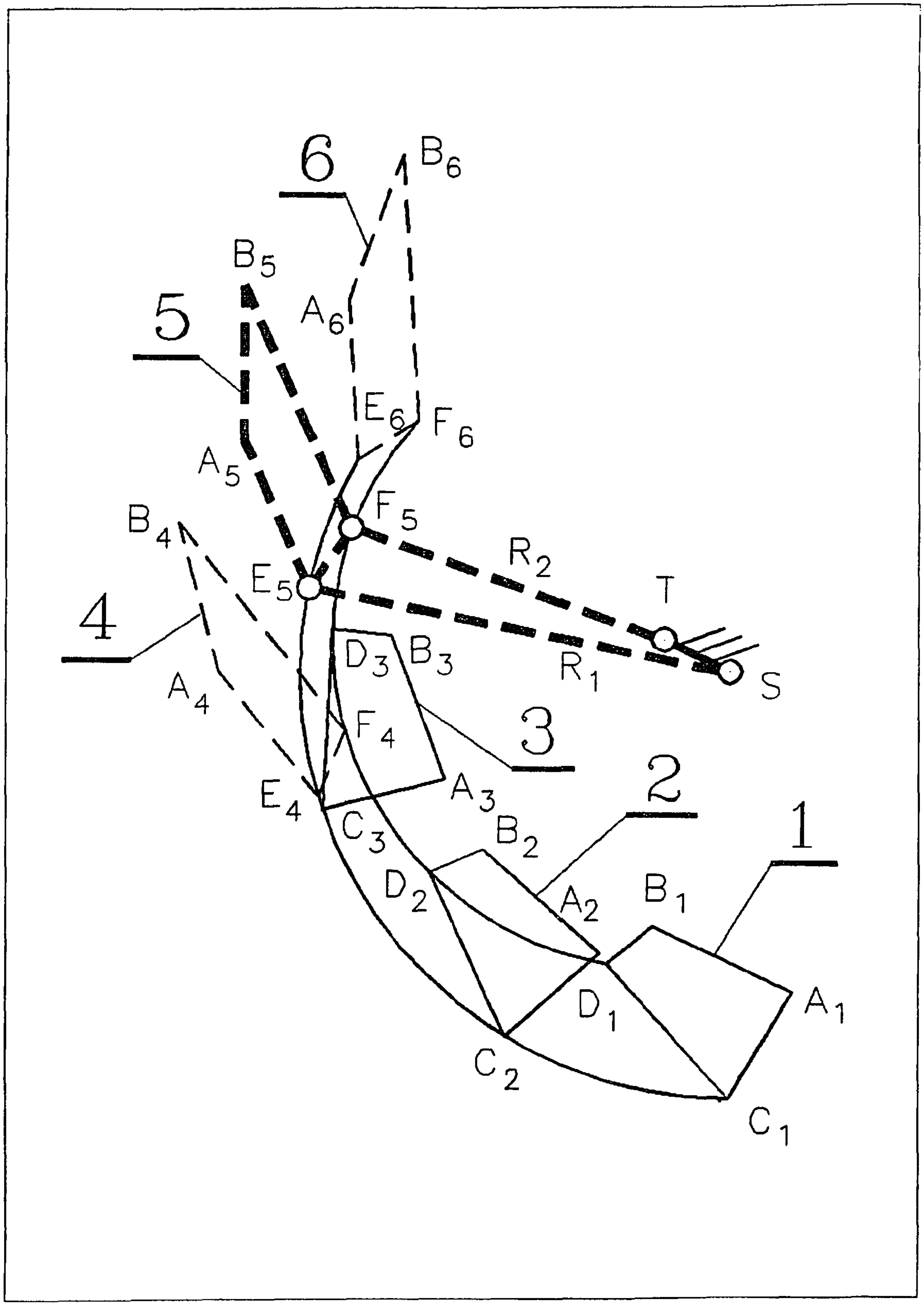

Figure 8.30 The resulting linkage at position 5

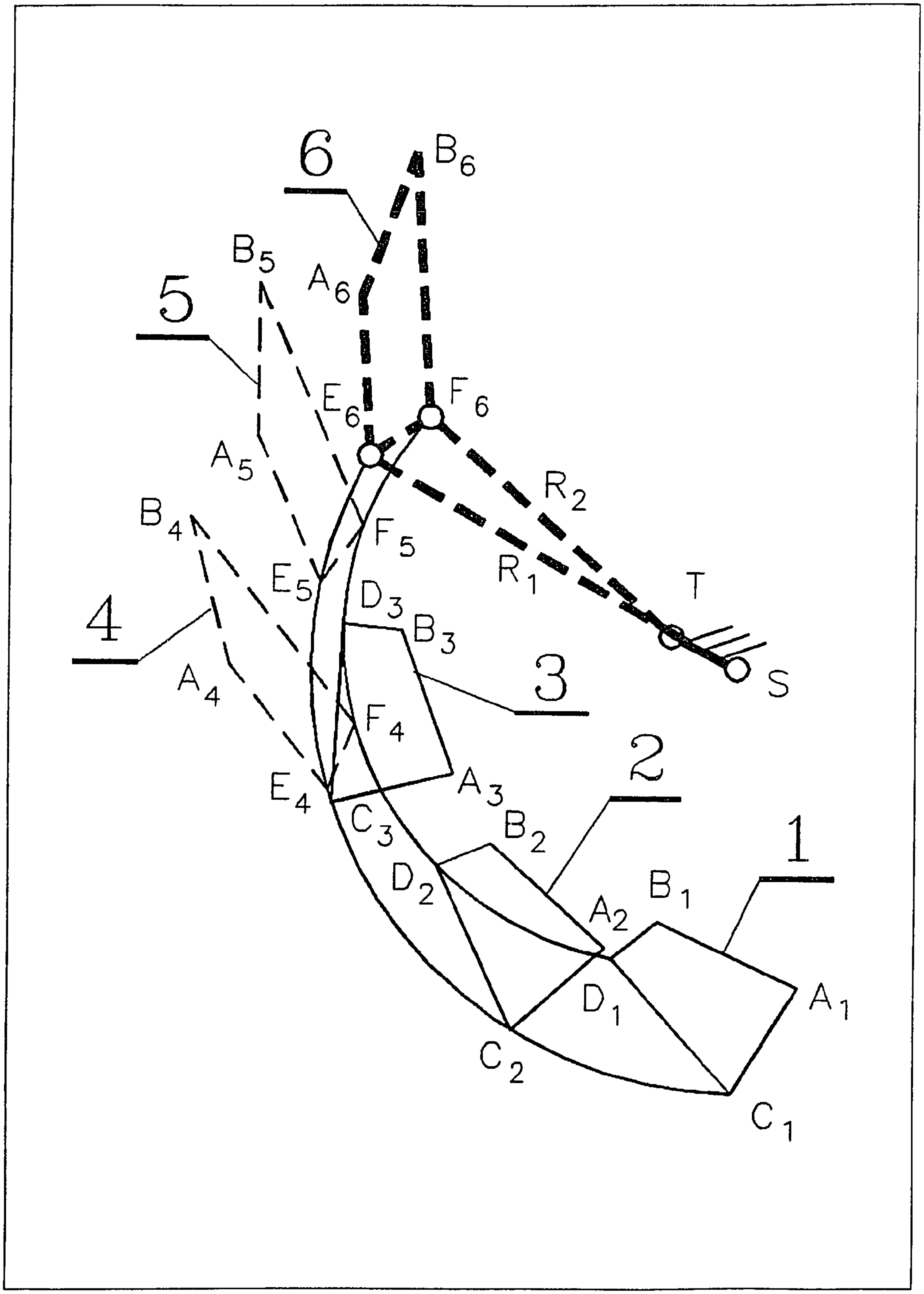

Figure 8.31 The resulting linkage at position 6

Table 8.7 The given data for example 4

Position	X	Y	θ
1	6.1500	2.0000	155.00
2	4.6500	2.3000	138.00
3	3.4500	3.6500	110.00
4	1.7000	4.5000	105.00
5	1.9000	6.3000	90.000
6	2.7000	7.4000	70.000

Table 8.8 The resulting data for example 4

Point	X	Y
S	5.6536	4.4931
C_1	5.6496	1.1671
E_4	2.4739	3.5175
T	5.1460	4.7435
D_1	4.7079	2.2194
F_4	2.6840	4.0350

8.6 Example 5

This problem needs two positions in each of the three phases with no position shared. The given data for this three phase example problem are listed in Table 8.9. Let us try adjustable crank length for both driving and driven side.

Draw six given positions in Figure 8.32 by calling the user-defined AutoLISP function PLOT_POS. Create an input data file CL_2_2_2.DAT and run the Turbo Pascal program CL_2_2_2.PAS. Plot circle point curve 12-34-56, and pick a circle point C_1 on the curve. Locate C_2, C_3, C_4, C_5, and C_6 by calling an user-defined AutoLISP function TRIANG. The bisectors for line segments C_1C_2 and C_5C_6 intersect at point T, which is one of the center point.

R_1, the radius of the circle passing through circle points C_1 and C_2 with center T is the crank length of phase 1. Similarly, R_3, the radius of the circle passing through circle points C_5 and C_6 with center T is the crank length of phase 3. Draw a circle passing through circle point C_2 with center T. As shown in the figure, this circle passes precisely through the circle point C_3. We have an unique circle point C, an unique center point T, and three distinct crank lengths R_1, R_2, and R_3, which satisfies the requirement of the problem. This indicates the validity of both the method and the program CL_2_2_2.PAS.

Another circle point D_1 is then picked on the curve for the other side of the four-bar linkage. Similarly, find circle points D_2 through D_6, the crank lengths R_4, R_5, and R_6, and locate the corresponding center point S.

The resulting four-bar linkage is shown in six consecutive Figures (8.33 through 8.38). It can be seen by inspection that neither a branch defect nor an order problem occurs in the resulting drag-link, which means it is a good solution. In fact, either S or T side of the linkage can be used as the driving side.

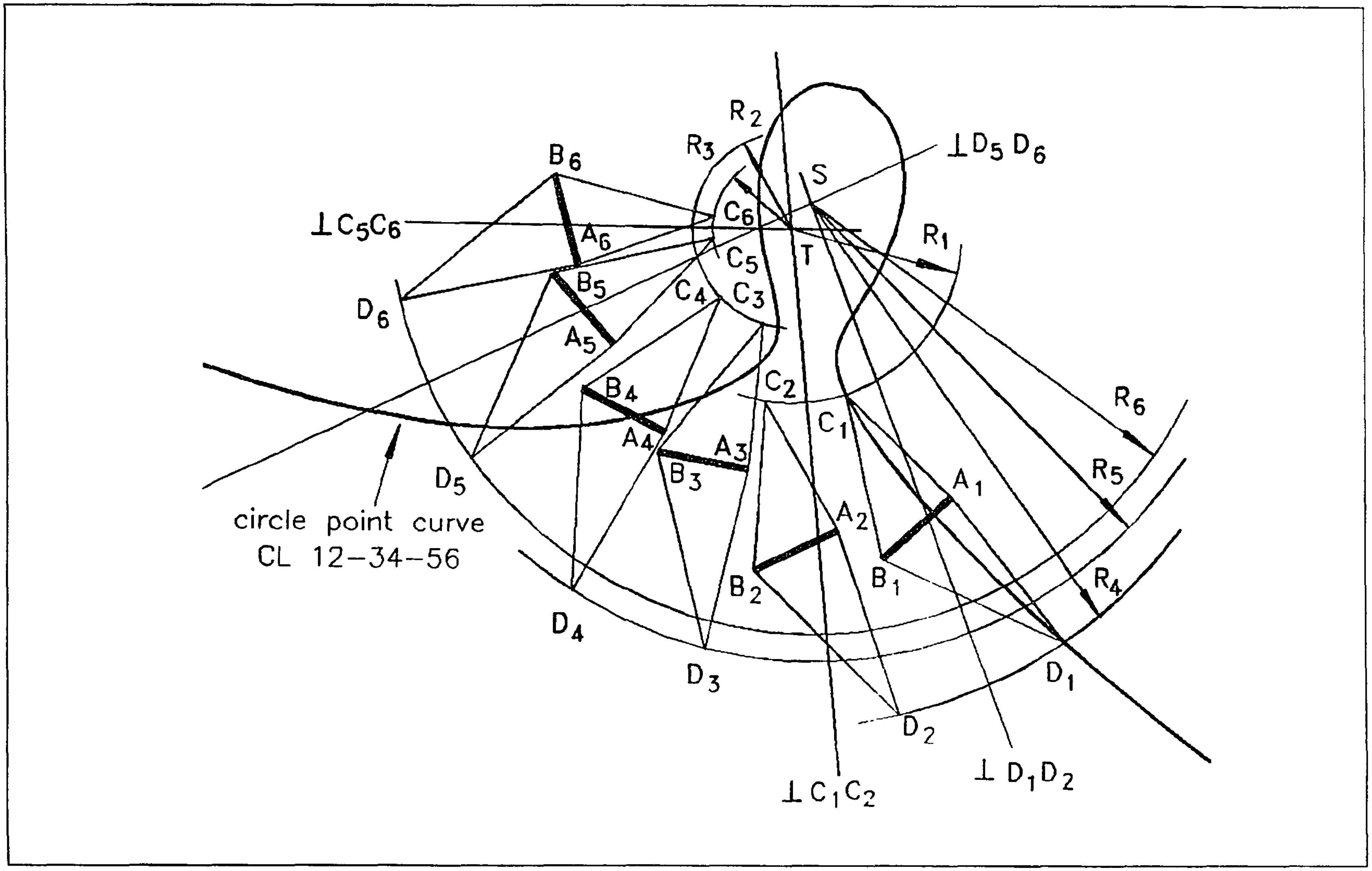

Figure 8.32 The six given positions, the circle point curve, and the moving pivots for example 5

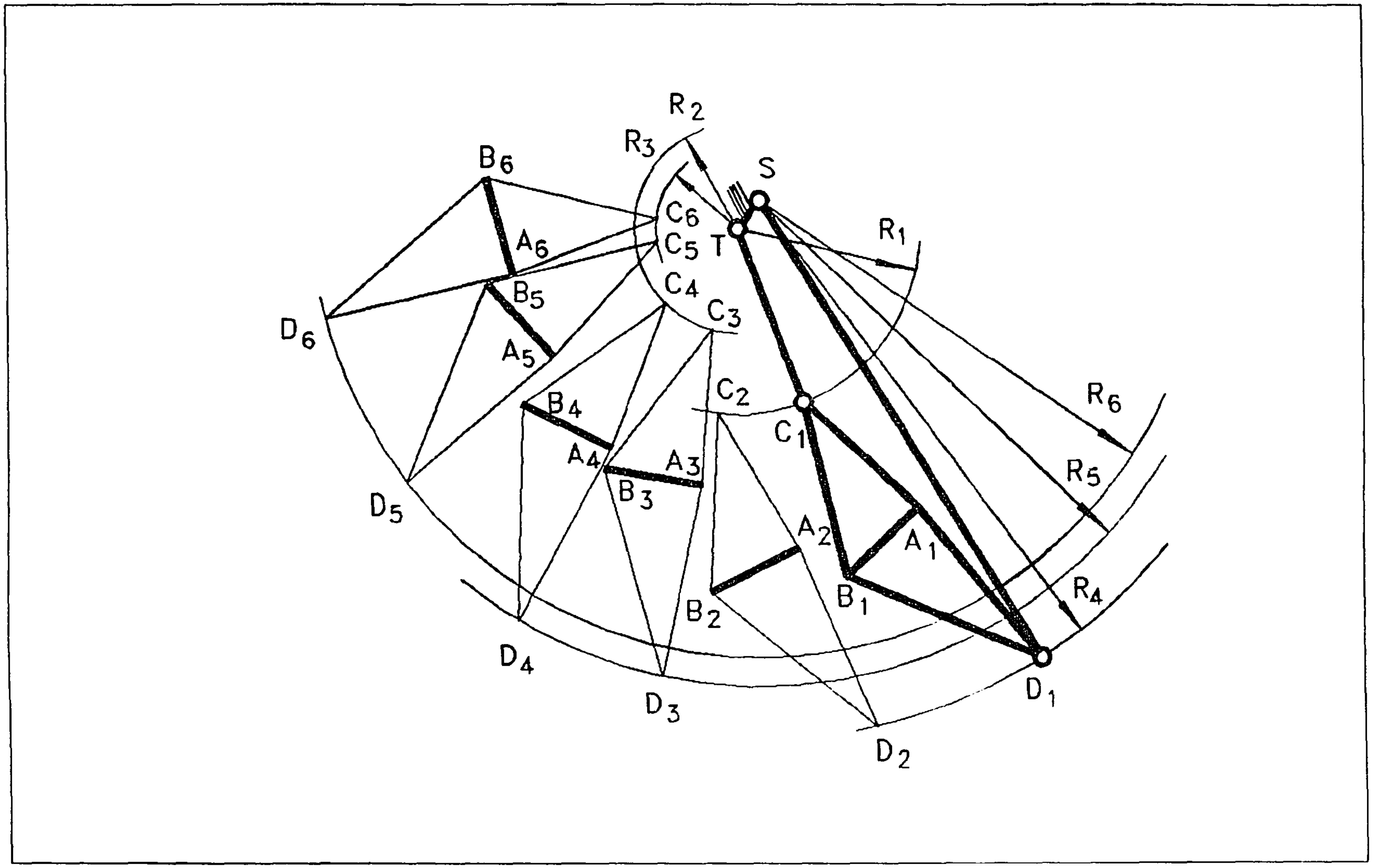

Figure 8.33 The resulting linkage at position 1

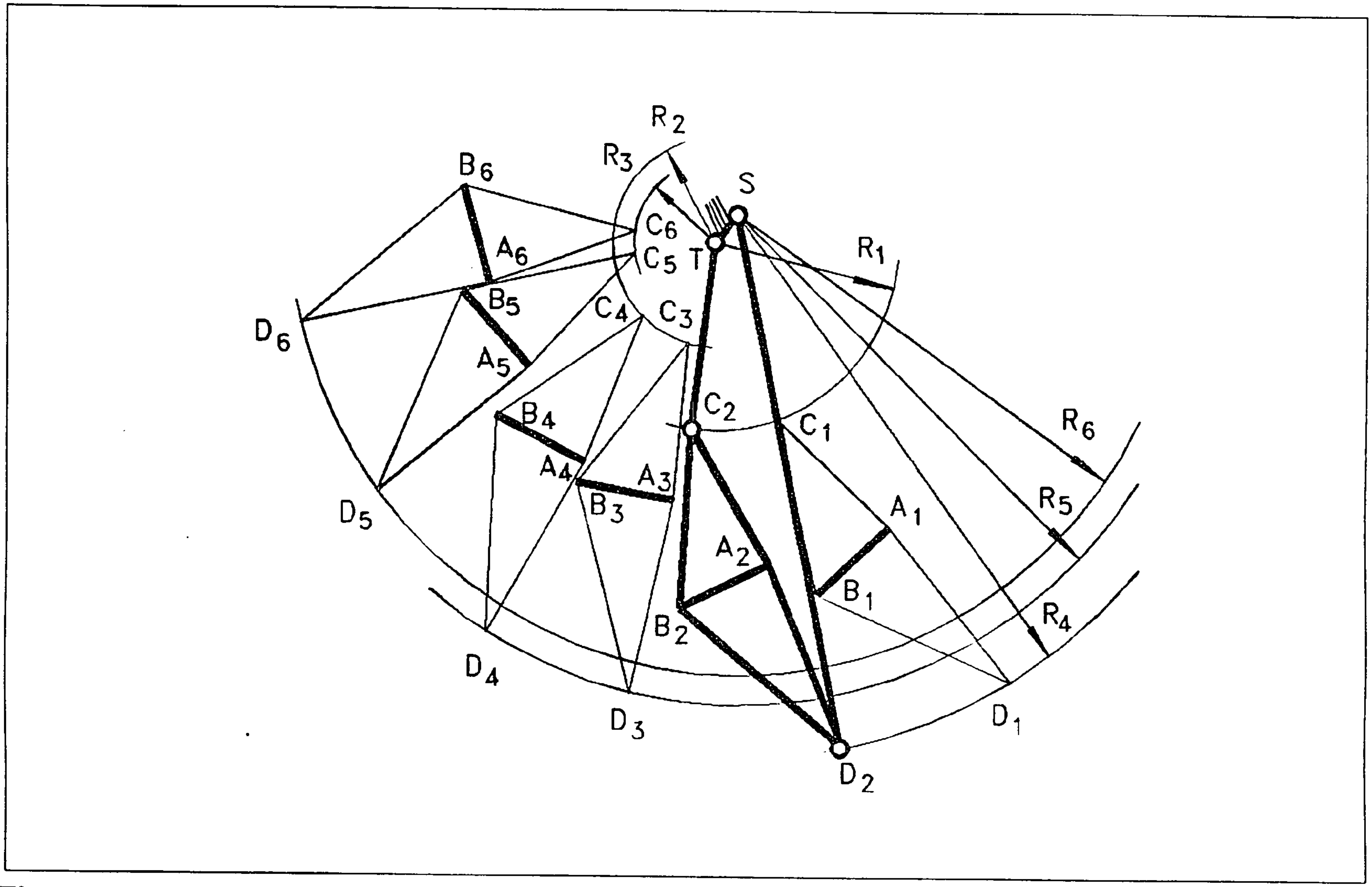

Figure 8.34 The resulting linkage at position 2

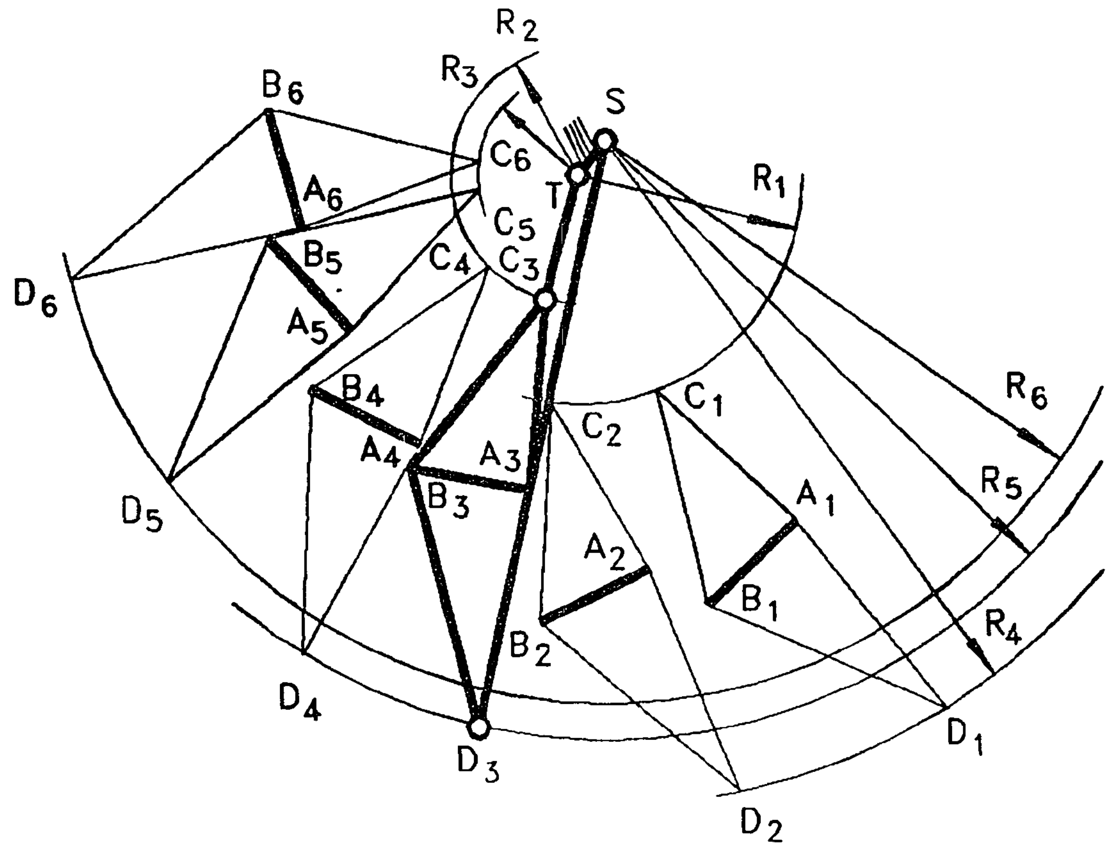

Figure 8.35 The resulting linkage at position 3

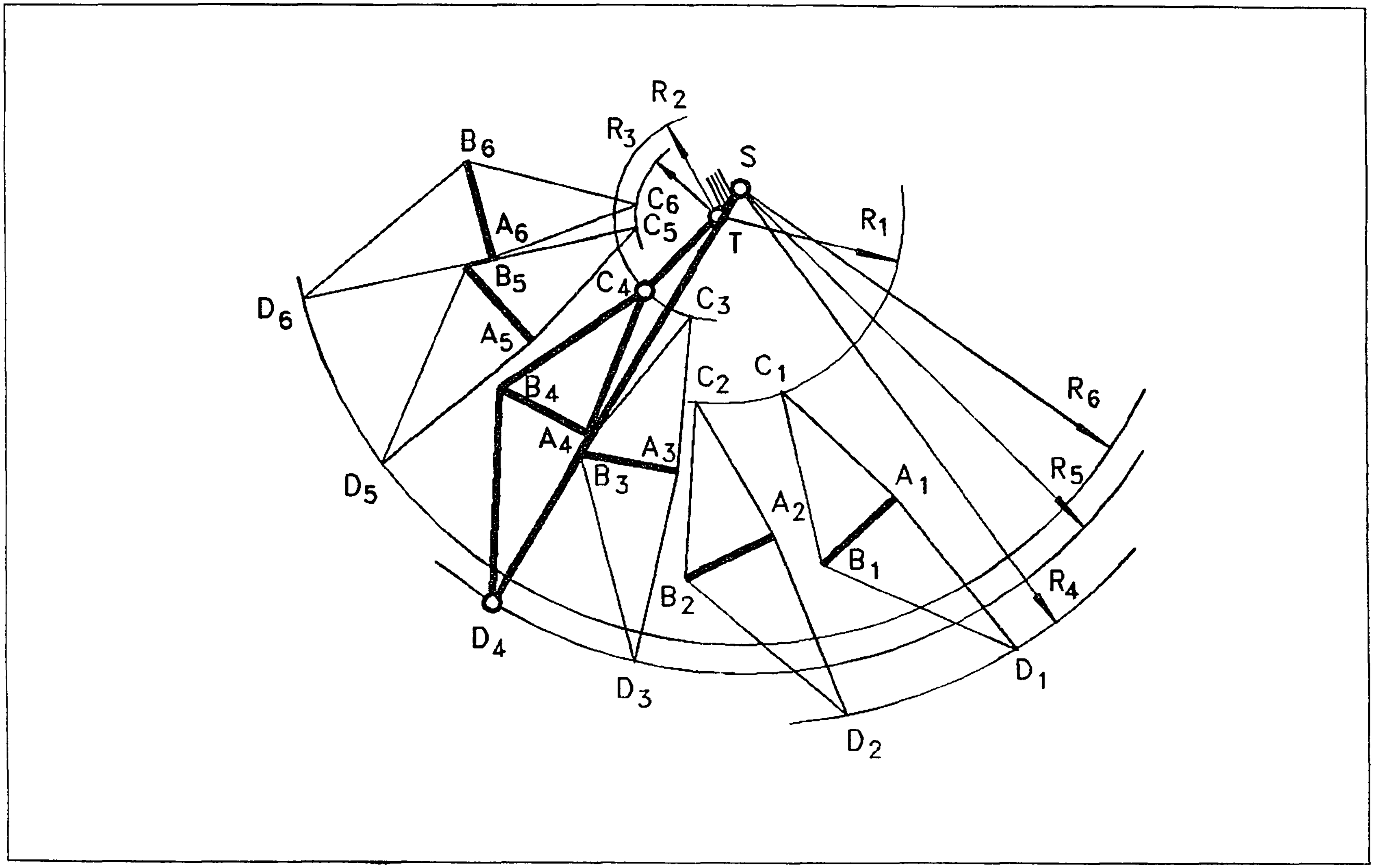

Figure 8.36 The resulting linkage at position 4

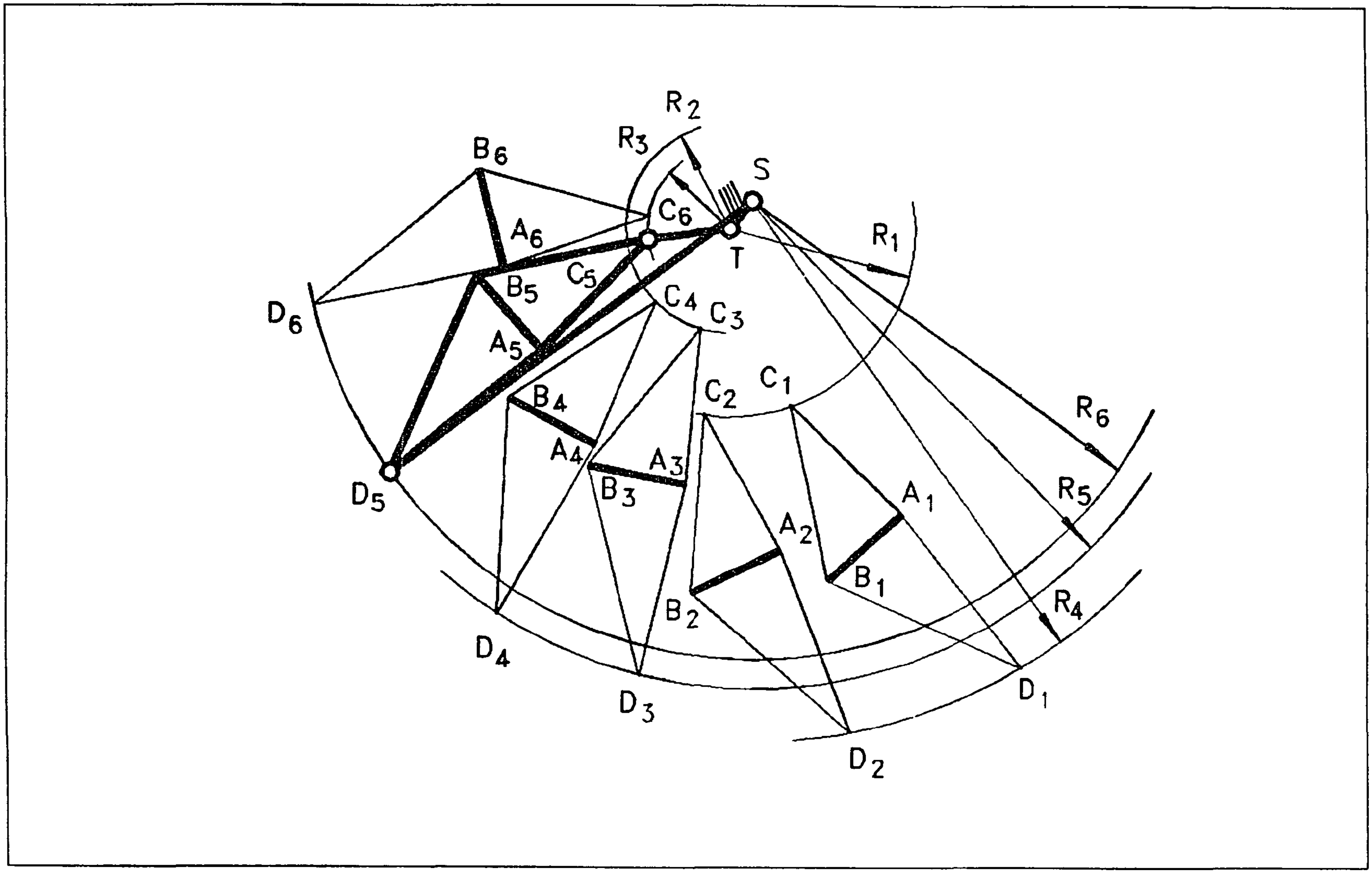

Figure 8.37 The resulting linkage at position 5

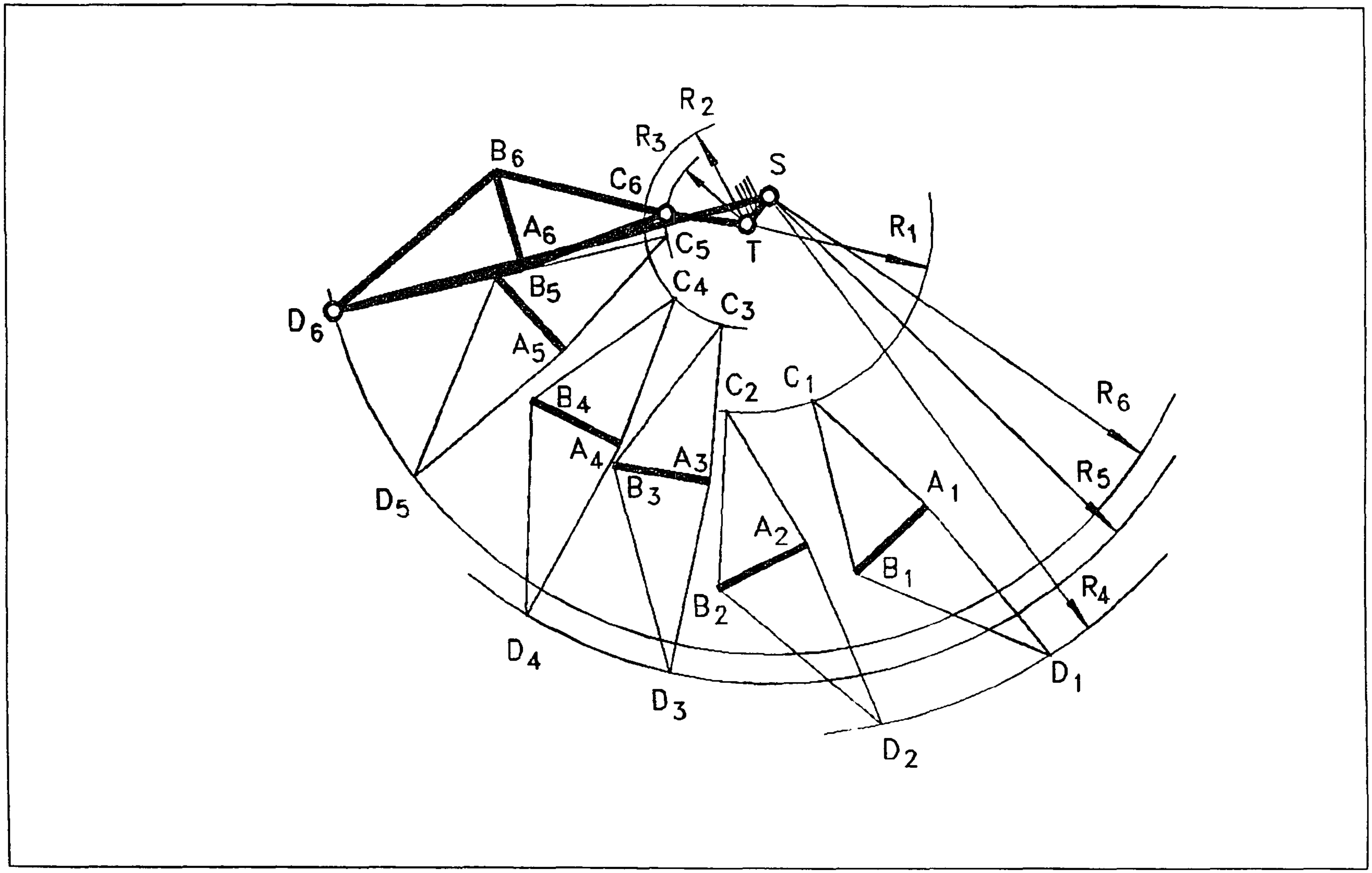

Figure 8.38 The resulting linkage at position 6

Table 8.9 The given data for example 5

Position	X	Y	θ
1	10.100	1.5454	222.25
2	8.2634	0.9779	205.73
3	6.8040	1.9346	170.00
4	5.4400	2.5000	153.00
5	4.6000	3.9000	132.00
6	4.0000	5.1500	105.00

Table 8.10 The resulting data for example 5

Point	X	Y
C_1	8.3847	3.1376
C_3	7.0080	4.2598
C_5	6.1989	5.6008
D_1	11.9606	–0.7320
D_3	6.1424	–0.9308
D_5	2.3145	2.0494
S	7.7734	6.1903
T	7.4283	5.7727

8.7 Example 6

This is the case of four positions in one phase and three in the other phase with no position shared. Since the total number of position is seven, an adjustable linkage is considered. Let us try adjustable moving pivot and crank length on both driving and driven side.

Plot seven positions in Figure 8.39 by means of the user-defined AutoLISP function PLOT_POS. Plot the center point curve for positions 1, 2, 3, and 4 and pick a center point S on it. Invert center point S for positions 2 through 4 into position 1 to get circle points C_1 through C_4 by using the user-defined AutoLISP function INVERT. Circle points C_1, C_2, C_3, C_4, and a circle passing through them are plotted automatically.

Since no order defect has been found for circle points C_1 through C_4, the center point S is inverted again from positions 6 and 7 into position 5, which is the first position of phase 2. The circle points E_5, E_6, and E_7 and a circle passing through them are plotted automatically on the screen and again no order defect has been found in phase 2.

Similarly, choose another center point T on the center point curve for the other side of the linkage (Figure 8.40). Do the same for center point T as that for center point S. The circle points D_1, D_2, D_3, D_4, F_5, F_6, and F_7 are shown in the figures.

Two groups of Filemon Construction Lines are plotted in Figure 8.43 by calling AutoLISP command FILEMON. In order to avoid a branch problem, circle point D_1 should be chosen outside the shaded area of the Filemon Construction Lines passing through point C_1. Similarly, circle point F_5 is outside of the shaded area of the Filemon Construction Lines passing through point E_5.

The resulting linkage is shown in seven consecutive Figures 8.42 through 8.48. By inspection, neither an order problem nor a branch defect occurs in this four-bar linkage. Both the moving pivots and the crank lengths are adjusted in both of the two phases.

In fact, either side of this particular linkage could be used as the driving side. The cranks rotate clockwise for positions 1 through 4 of phase 1, then counterclockwise for positions 5 through 7 of phase 2.

The given and resulting data are listed in Tables 8.11 and 8.12 respectively.

Table 8.11 The given data for example 6

position	X	Y	θ
1	3.3800	2.4800	120.00
2	2.6800	4.0900	100.00
3	2.7400	5.6000	83.000
4	3.3074	6.8723	65.790
5	1.6500	5.3600	79.000
6	1.6300	3.2000	90.000
7	2.5000	1.5300	102.00

Table 8.12 The resulting data for example 6

point	X	Y
C_1	6.3292	3.5641
D_1	5.7931	4.3650
E_5	4.6022	5.4528
F_5	4.2438	6.1891
S	6.8071	4.5241
T	6.7038	5.0031

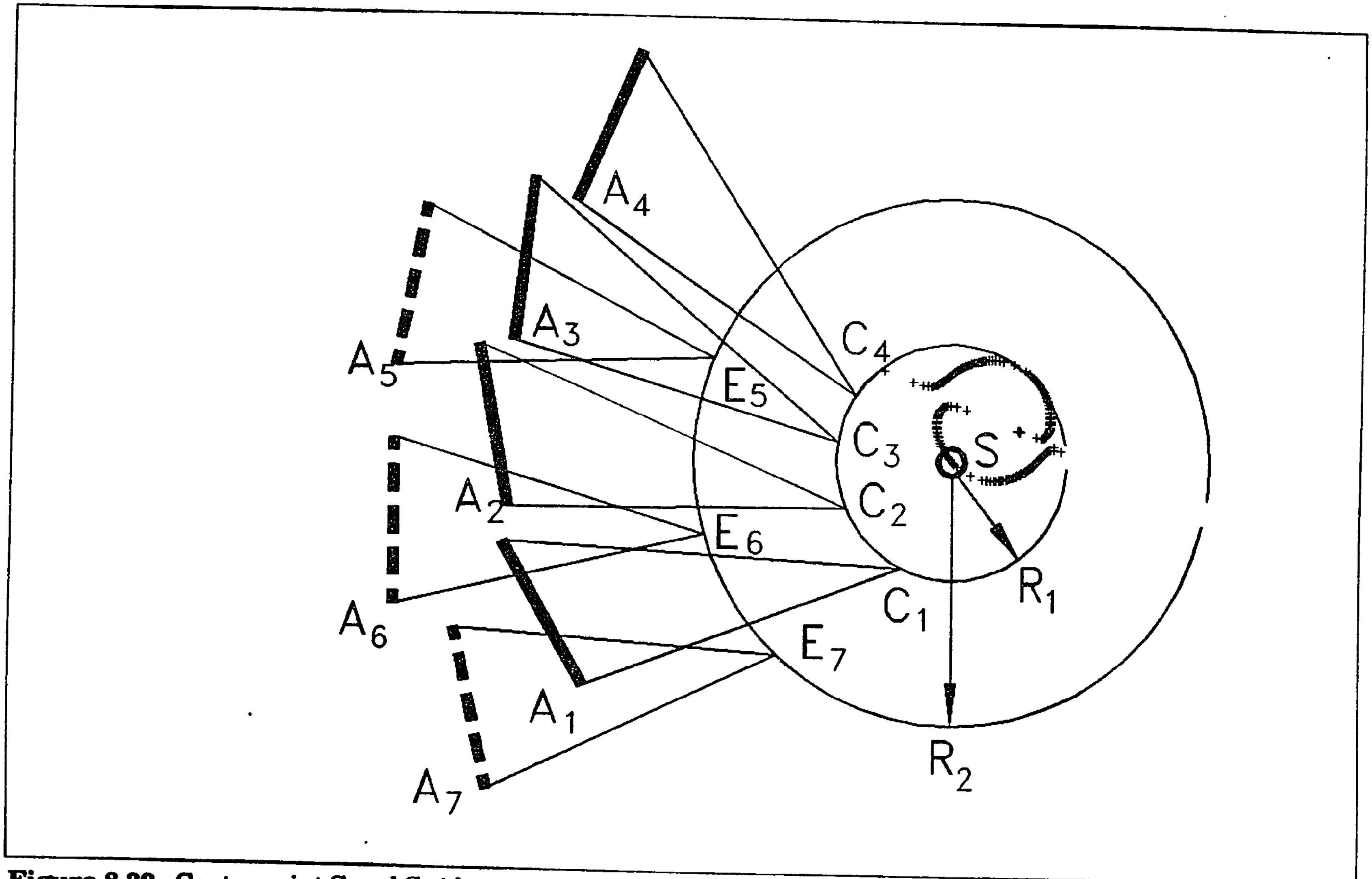

Figure 8.39 Center point S and S side

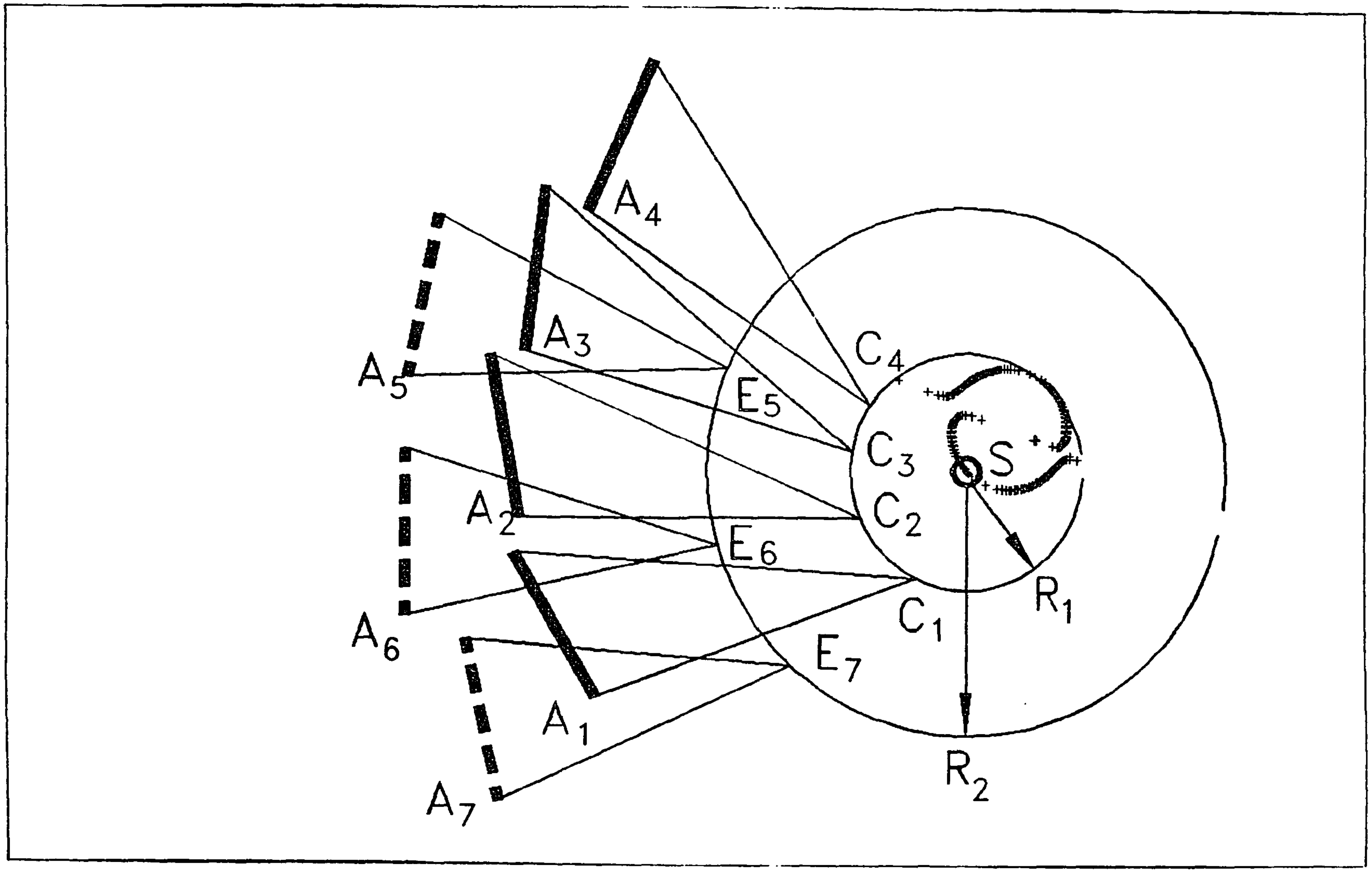

Figure 8.40 Center point T and T side

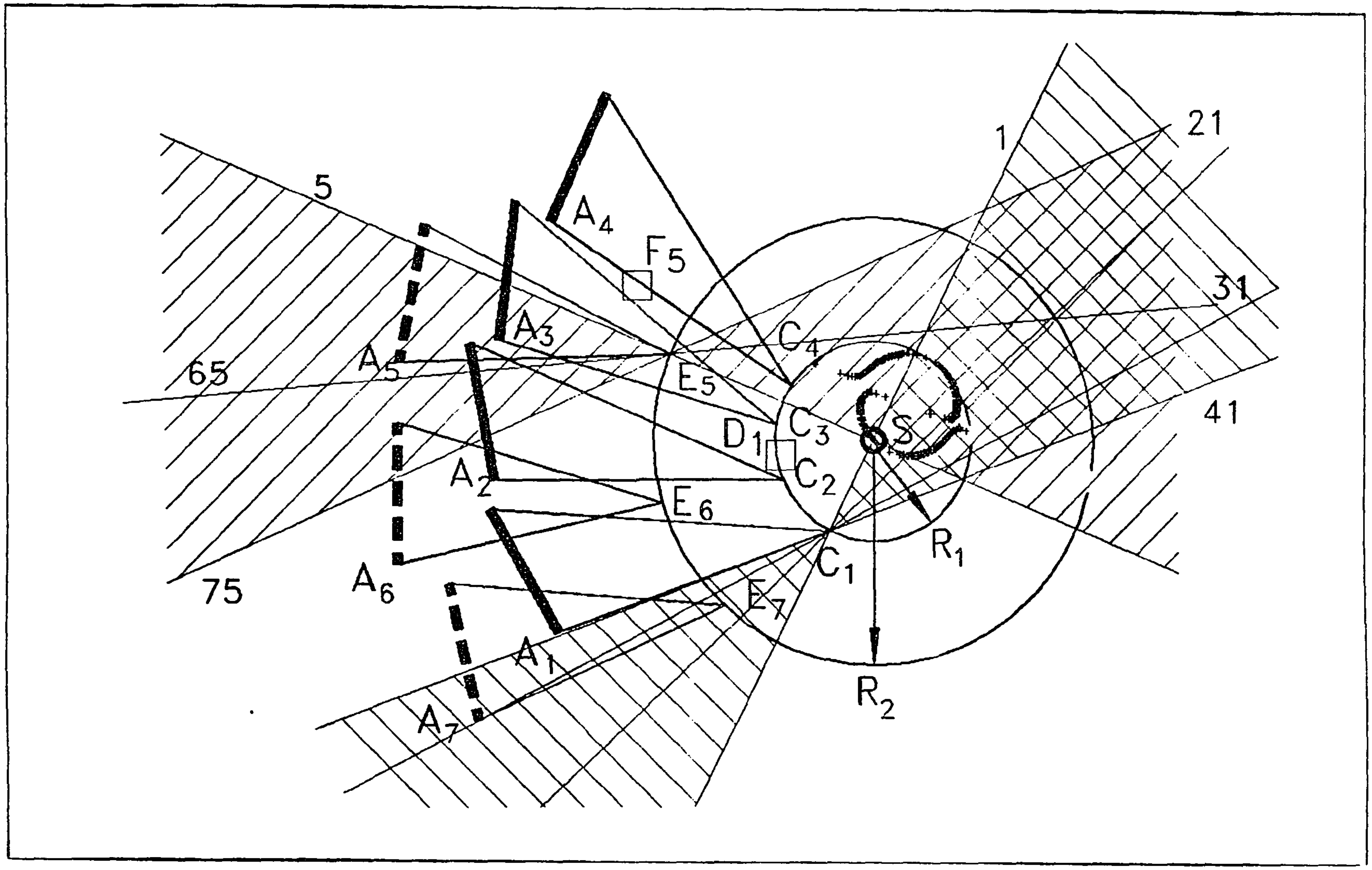

Figure 8.41 Filemon Construction Lines for both phase 1 and phase 2

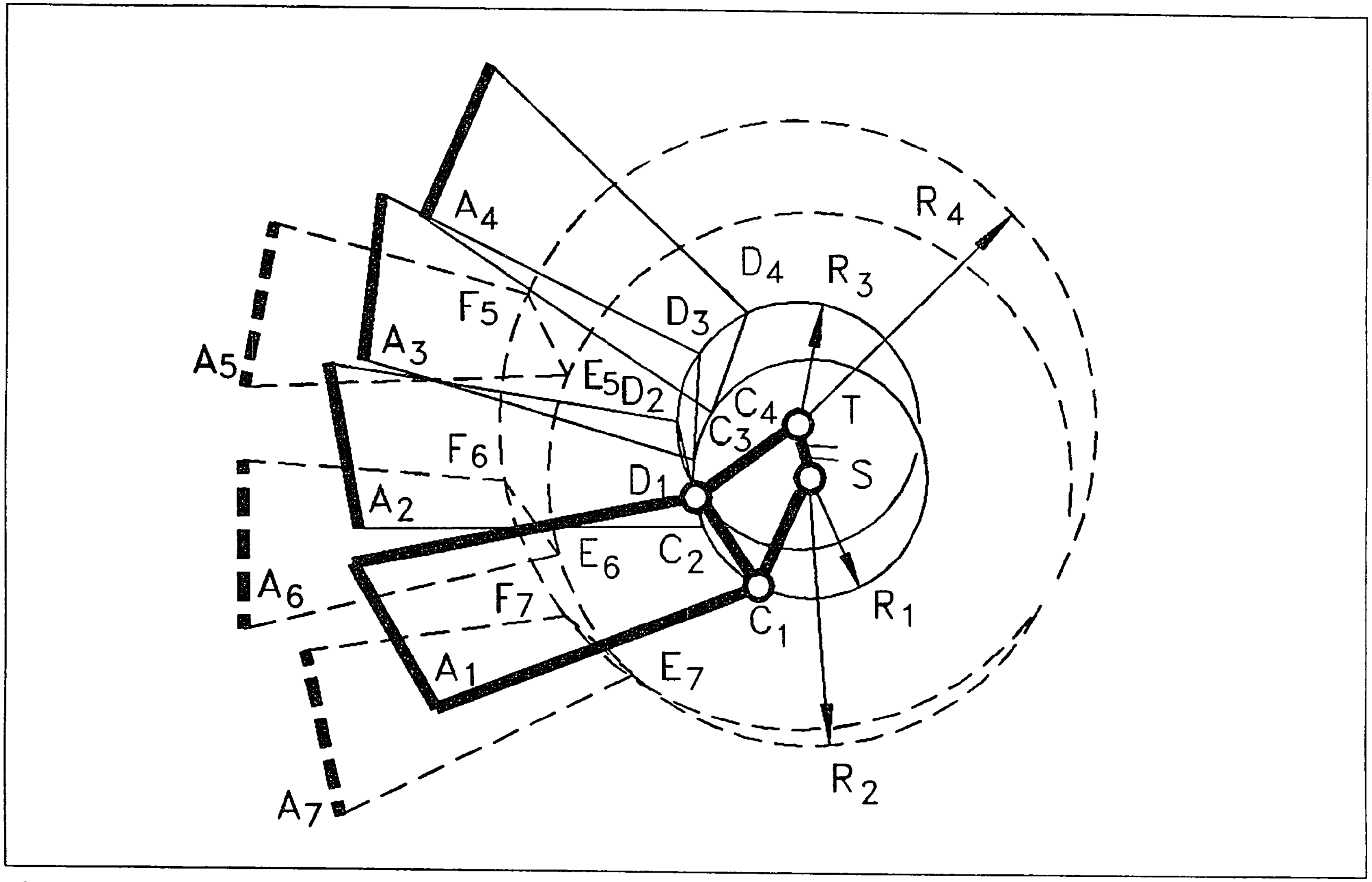

Figure 8.42 The resulting linkage at position 1

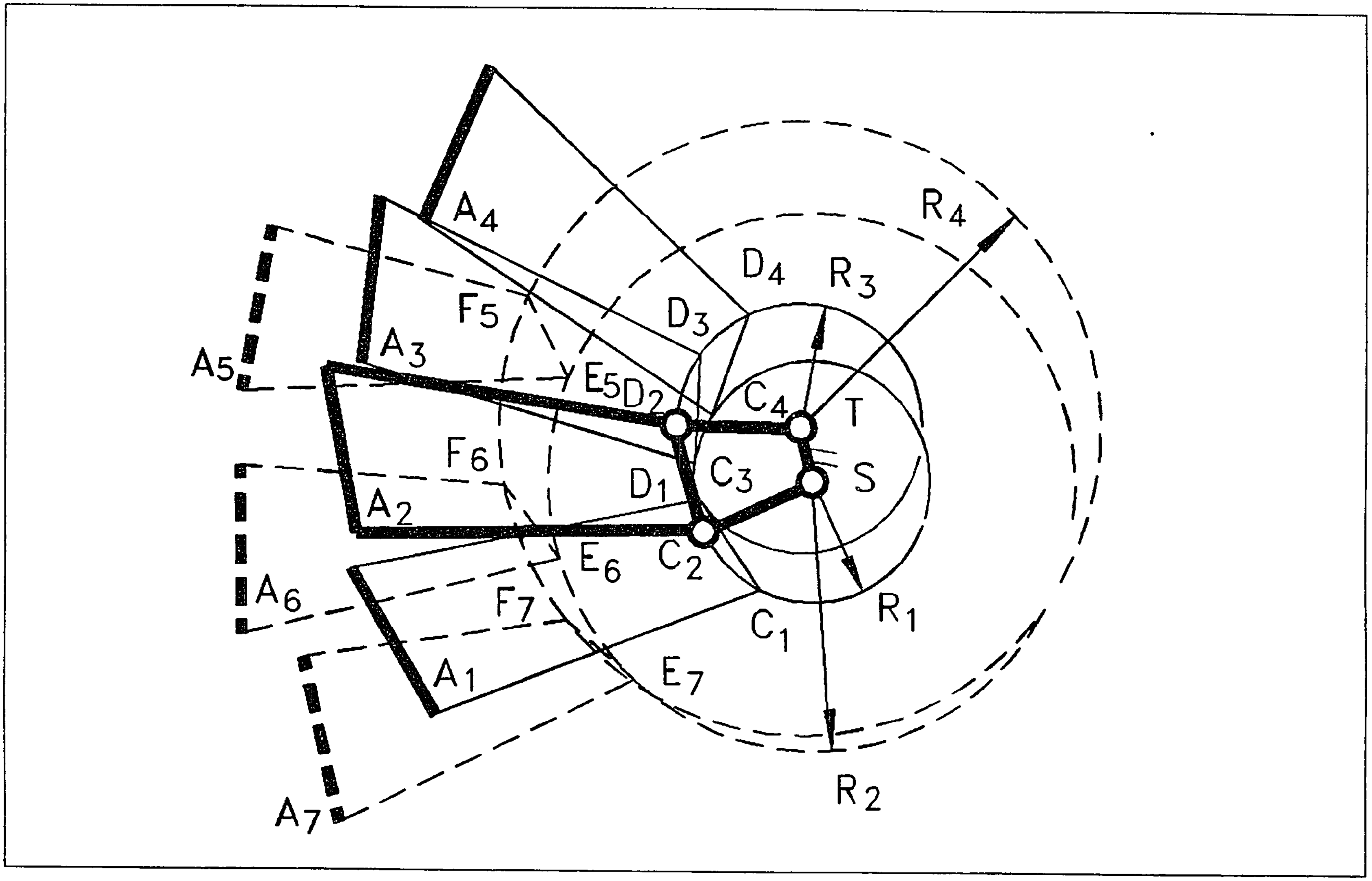

Figure 8.43 The resulting linkage at position 2

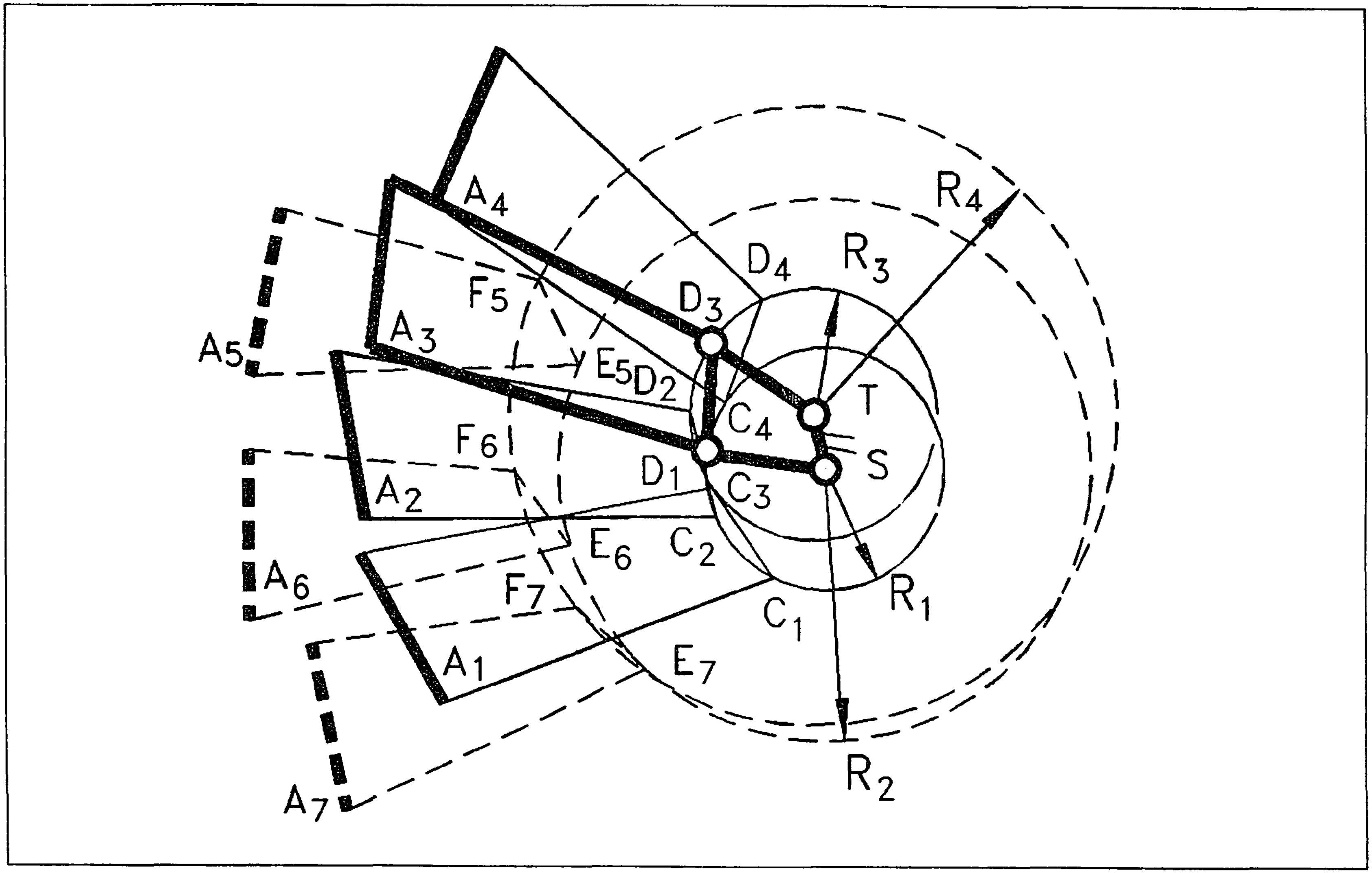

Figure 8.44 The resulting linkage at position 3

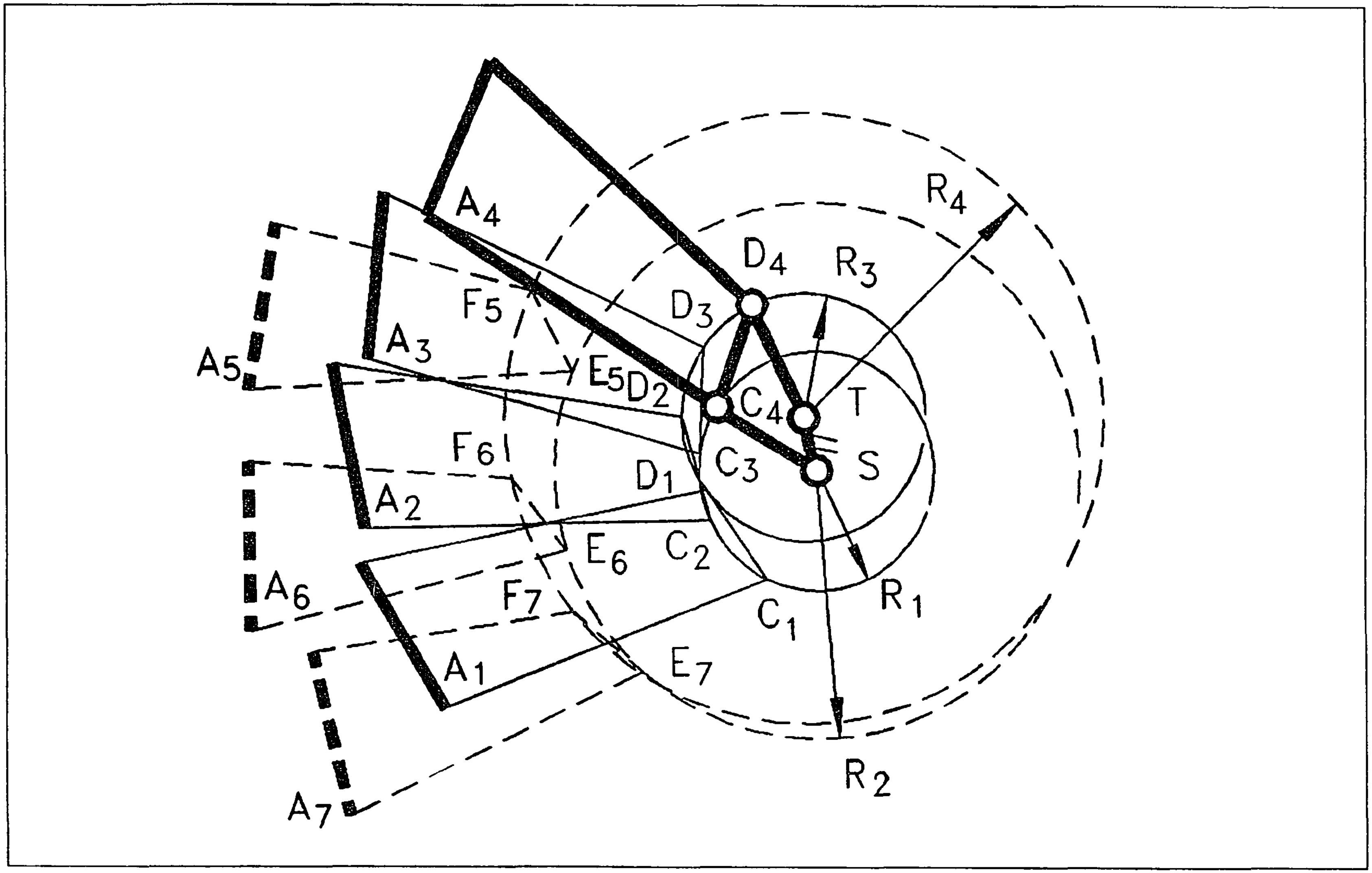

Figure 8.45 The resulting linkage at position 4

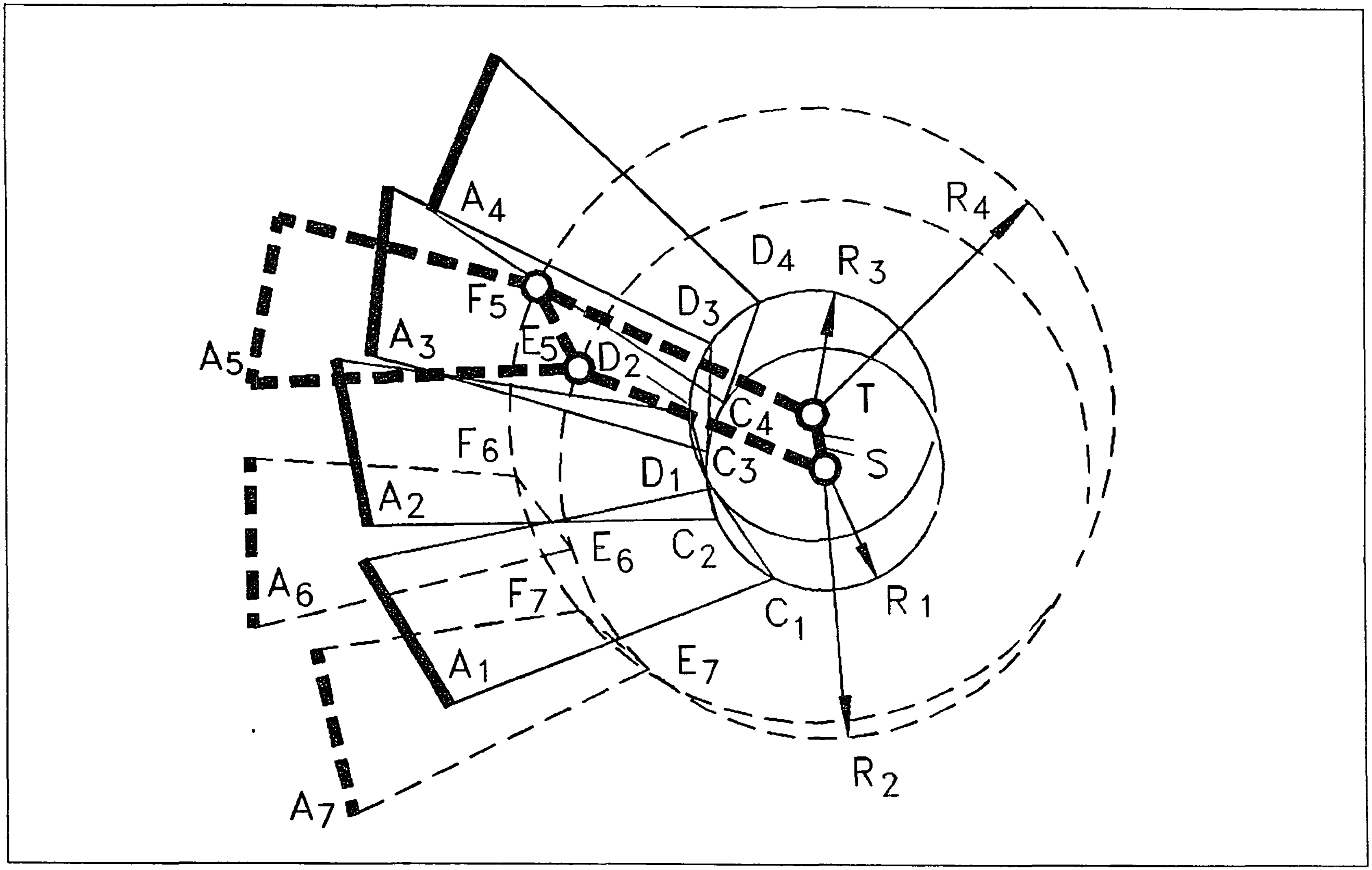

Figure 8.46 The resulting linkage at position 5

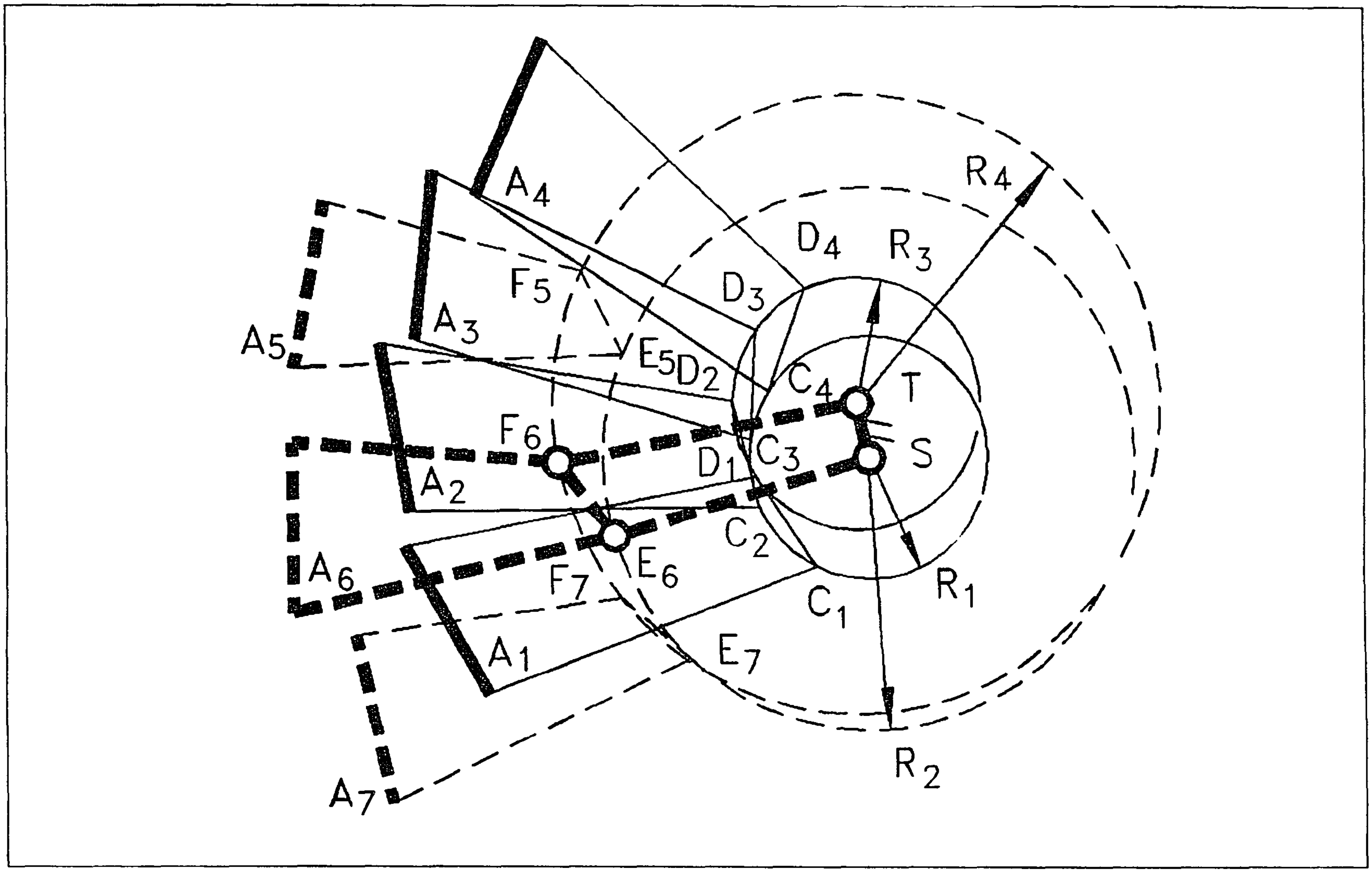

Figure 8.47 The resulting linkage at position 6

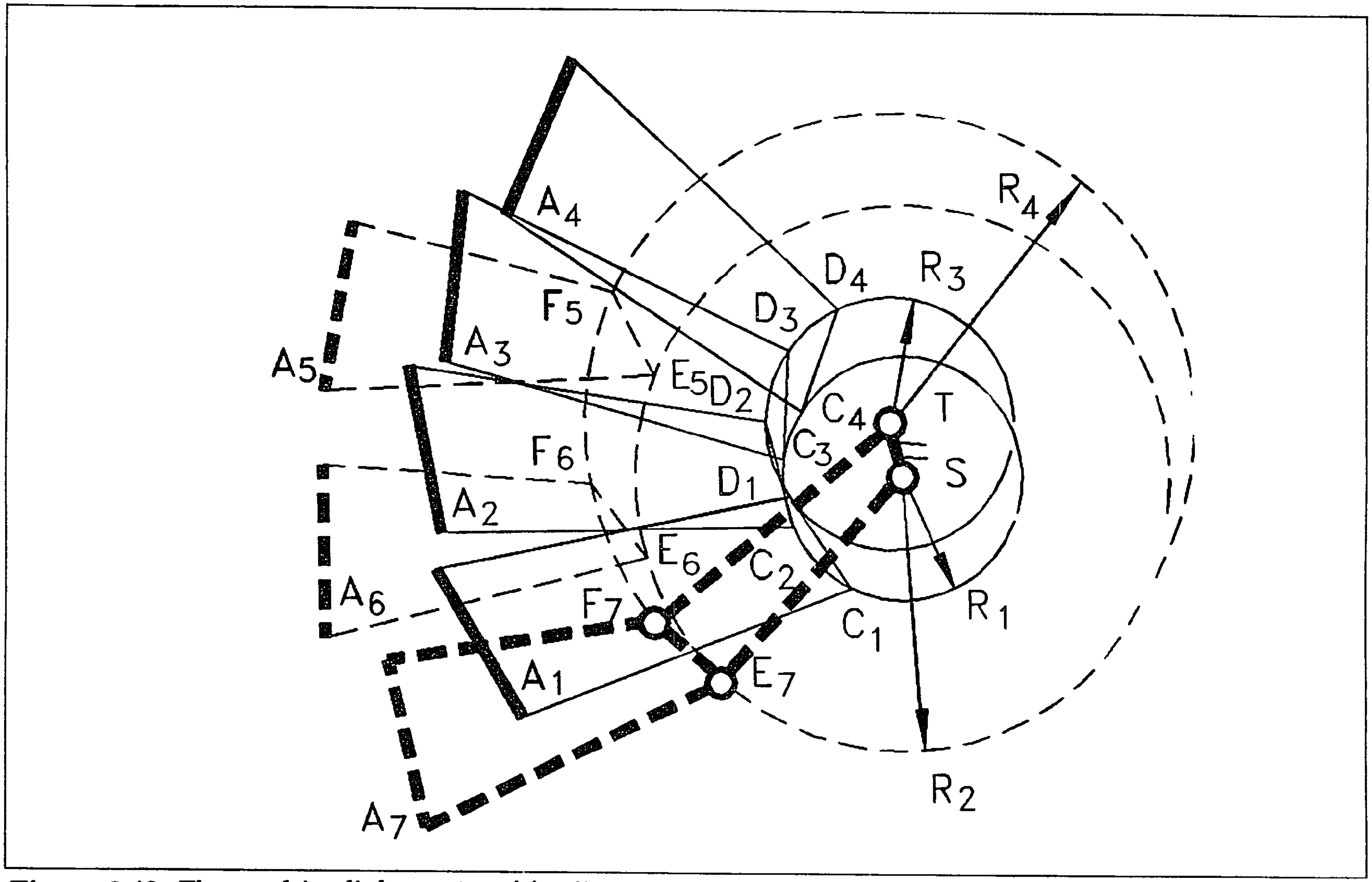

Figure 8.48 The resulting linkage at position 7

Chapter 9
Conclusions

The objective of this study was to solve the kinematic synthesis problems of multi-phase adjustable four-bar linkages for motion generation. The results of this research have demonstrated that this study has successfully developed techniques to solve adjustable moving pivot problems, adjustable moving pivot and crank length problems, and various three phase problems.

Equations are developed for the cases of adjustable moving pivot with three positions in each of the two phases. A numerical method with the solutions at the rotation poles as the initial solutions has been successfully developed to find more solutions for the problems. Center points and their corresponding circle points which satisfy the equations are obtained by the programs MP_3_3.PAS and MP_3_3_1.PAS. The adjustable moving pivot problem of seven given positions can be solved by intersecting two such center point curves.

The method for solving two phase adjustable moving pivot problems can be extended for three phase problems of the same kind. For example, any solution for the two phase problem MP 123–456 is a solution of three phase problem MP 123–456–78, and so do the other combinations of the number of the positions with the total of eight positions. Three phase adjustable moving pivot problems with the total of nine positions can be solved by intersecting center point curves. More points are needed in order to get a much precise solution.

A circle point curve equation is also developed for the case of adjustable crank length problem with two positions in each of the three phases. Circle points satisfying the equations of the problem are obtained by plotting a circle point curve. The adjustable crank length problems of seven given positions are solved by intersecting two circle point curves.

Many two phase adjustable moving pivot and crank length problems are solved by plotting center point curves or intersecting two of such curves.

The three phase adjustable fixed pivot problems are solved by plotting circle point curves, or intersecting two of such curves.

It is well known that the maximum permissible number of prescribed positions for motion generation for a four–bar linkage is five. It has been shown in this study that the maximum number of prescribed positions for an adjustable four–bar linkage varies from seven to nine for the problems involved. Many examples with high number of positions have found a good solution in this study. However, to have one or two free choices of parameters for an adjustable four–bar linkage usually gets better results than solutions without any free choice of parameters. This is because the design of a linkage has to satisfy not only the basic equations but also some other conditions, such as branch problem, order problem, the transmission angle and efficiency, etc.

Appendices

A. AutoLISP Programs

```
; Plot given positions to ACAD screen, good to any number of
positions.
(defun plot_pos  (n)
  (setq f1(open "input.dat" "r")
        d(getdist "\nEnter a length:")
  )
  (repeat  n
      (setq x(atof (read-line f1))
            y(atof (read-line f1))
            th1d(atof (read-line  f1))
            th1(/ (* th1d pi) 180.0)
            a1(list x y)
            b1(polar a1 th1 d)
      )
      (command "pline" a1 b1 "")
  )
  (close f1)
  (princ)
)


; This lisp function has been used by mp_3_3 and mp_3_3_1 to
plot points.
(defun C:plot_pts(/ f t)
  (setq  f(open  "bus.dat" "r"))
  (setvar "pdmode" 0)
  (while (/= t "nil")
         (setq  t(read-line  f))
         (command "point"  t)
  )
  (close f)
)


(defun pts_+(s / f t p1 p2 p3 p4 x1 x2 y1 y2 x y)
  (setq  f(open  "bus.dat" "r"))
  (while (/= t "nil")
         (setq  t(read-line  f)
                x(atof     t)
                y(atof(read-line  f))
                x2(+  x  s)
                x1(-  x  s)
                y2(+  y  s)
                y1(-  y  s)
                p1(list x1 y)
                p2(list x2 y)
                p3(list x  y2)
                p4(list x  y1)
         )
```

```
        )
        (command "pline"  p1  p2  "")
        (command "pline"  p3  p4  "")
  )
  (close f)
)


(defun pts_diam(s / f t p1 p2 p3 p4 x1 x2 y1 y2 x y)
  (setq  f(open  "bus.dat" "r"))
  (while (/= t "nil")
        (setq  t(read-line  f)
               x(atof     t)
               y(atof(read-line  f))
               x2(+  x  s)
               x1(-  x  s)
               y2(+  y  s)
               y1(-  y  s)
               p1(list x1 y)
               p2(list x2 y)
               p3(list x  y2)
               p4(list x  y1)
        )
        (command "point" (list x y))
        (command "pline"  p1 p3 p2 p4 "c")
  )
  (close f)
)


(defun pts_x(s / f t p1 p2 p3 p4 x1 x2 y1 y2 x y)
  (setq  f(open  "bus.dat" "r"))
  (while (/= t "nil")
        (setq  t(read-line  f)
               x(atof     t)
               y(atof(read-line  f))
               x2(+  x  s)
               x1(-  x  s)
               y2(+  y  s)
               y1(-  y  s)
               p1(list x2 y2)
               p2(list x1 y1)
               p3(list x1 y2)
               p4(list x2 y1)
        )
        (command "pline"  p1  p2  "")
        (command "pline"  p3  p4  "")
  )
  (close f)
)


(defun pts_sq(s / f t p1 p2 p3 p4 x1 x2 y1 y2 x y)
  (setq  f(open  "bus.dat" "r"))
```

```
  (while (/= t "nil")
         (setq  t(read-line  f)
                x(atof     t)
                y(atof(read-line  f))
                x2(+  x  s)
                x1(-  x  s)
                y2(+  y  s)
                y1(-  y  s)
                p1(list x2 y2)
                p2(list x1 y1)
                p3(list x1 y2)
                p4(list x2 y1)
         )
         (command "point"  (list x y))
         (command "pline"  p1 p3 p2 p4 "c")
  )
  (close f)
)


(defun pts_circ(r / f t c x y)
  (setq  f(open  "bus.dat" "r"))
  (while (/= t "nil")
         (setq  t(read-line  f)
                x(atof     t)
                y(atof(read-line  f))
                c(list x y)
         )
         (command "circle"  c  r)
         (command "donut"  0.0  0.005  c "")
  )
  (close f)
)


(defun C:pick_pt(/ f k)
       (setq  f(open  "pick_pt.out"  "w")
              k(getpoint "\nPick a point by mouse: "))
       (write-line "The point you just picked is:"  f)
       (write-line (rtos (car  k)  2  4)  f)
       (write-line (rtos (cadr k)  2  4)  f)
       (write-line (rtos (car  k)  2  4)  )
       (write-line (rtos (cadr k)  2  4)  )
       (close f)
       (princ)
)


(defun pick_pt(n / f k)
       (setq  f(open  "pick_pt.out"  "w"))
      ( repeat n
       (setq  k(getpoint  "\nPick a point by mouse: "))
       (write-line (rtos (car  k)  2  4)  f)
       (write-line (rtos (cadr k)  2  4)  f)
```

```
        (write-line (rtos (car  k)  2  4)  )
        (write-line (rtos (cadr k)  2  4)  )
       )
        (close f)
        (princ)
)


(defun C:pick_ang()
        (setq  f(open  "pick_ang.out"  "w")
               th(getangle "\nPick 2 points to show the
angle: ")
               thd(/ (* th 180.0) pi )
        )
        (write-line (rtos thd  2  4)  f)
        (write-line (rtos thd  2  4)  )
        (close f)
        (princ)
)


(defun pick_ang(n / f th thd)
        (setq  f(open  "pick_ang.out"  "w"))
        (repeat n
           (setq    th(getangle "\nPick 2 points to show the
angle: ")
                    thd(/ (* th 180.0) pi )
           )
           (write-line (rtos thd  2  4)  f)
           (write-line (rtos thd  2  4)  )
        )
        (close f)
        (princ)
)



; Bisector version 1 -- the interactive version, A command.
; Global: k1 k2 pt1 pt2
(defun bisect (/ k3 k4 k5 k6 k7 pt1x pt1y pt2x pt2y)
  (setq k1 (getpoint "\nEnter first point (key in or by
mouse): ")
        k2 (getpoint "\nEnter second point (key in or by
mouse): ")
        k3 (/ (+ (car  k1) (car  k2)) 2)
        k4 (/ (- (cadr k2) (cadr k1)) (distance k1 k2))
        k5 (/ (+ (cadr k2) (cadr k1)) 2)
        k6 (/ (- (car  k2) (car  k1)) (distance k1 k2))
        k7 (getdist (list k3 k5) "\nEnter length: ")
        pt1x (- k3 (* k4 k7))
        pt1y (+ k5 (* k6 k7))
        pt2x (+ k3 (* k4 k7))
        pt2y (- k5 (* k6 k7))
        pt1 (list pt1x pt1y)
```

```
        pt2 (list pt2x pt2y)
   )
   (setq f1(open "bisect.out" "w"))
   (write-line "The 1st point entered for function BISECT
is:"  f1)
   (write-line (rtos (car  k1)  2  4)  f1)
   (write-line (rtos (cadr k1)  2  4)  f1)
   (write-line "The 2nd point entered for function BISECT
is:"  f1)
   (write-line (rtos (car  k2)  2  4)  f1)
   (write-line (rtos (cadr k2)  2  4)  f1)
   (close f1)
   (command "pline" pt1 pt2 "")
   (princ)
)


; Bisector version 2 -- to be called by the function
invert(),
; and other routines. k1, k2, pt1, and pt2 are global
variables.
; Values should be assigned to k1 & k2 prior to calling this
function.
; This version is from Wilhelm[12]

(defun bisector (/ k3 k4 k5 k6 k7 pt1x pt1y pt2x pt2y)
  (setq k3 (/ (+ (car  k1) (car  k2)) 2)
        k4 (/ (- (cadr k2) (cadr k1)) (distance k1 k2))
        k5 (/ (+ (cadr k2) (cadr k1)) 2)
        k6 (/ (- (car  k2) (car  k1)) (distance k1 k2))
       k7 2      ; assign arbitrary length
      ;remove the following line when not display the
bisectors
      ;and put the line above into the program
;         k7 (getdist (list k3 k5) "\nEnter length: ")
        pt1x (- k3 (* k4 k7))
        pt1y (+ k5 (* k6 k7))
        pt2x (+ k3 (* k4 k7))
        pt2y (- k5 (* k6 k7))
        pt1 (list pt1x pt1y)
        pt2 (list pt2x pt2y)
  )
;  (command "pline" pt1 pt2 "")    ;delete this line, while
not to display bisectors
  (princ)
)


;Intersection point of 2 bisectors
;k1: global var of function BISECT
;
(defun C:biinters (/ pt3 pt4)
       (bisect)
```

```
        (setq f(open "biinters.out" "w"))
        (write-line "The 1st point entered for function
 BISECT is:"  f)
        (write-line (rtos (car  k1)  2  4)  f)
        (write-line (rtos (cadr k1)  2  4)  f)
        (write-line "The 2nd point entered for function
 BISECT is:"  f)
        (write-line (rtos (car  k2)  2  4)  f)
        (write-line (rtos (cadr k2)  2  4)  f)
        (setq pt3 pt1)
        (setq pt4 pt2)
        (setq k k1)
        (bisect)
        (write-line "The 3rd point entered for function
 BISECT is:"  f)
        (write-line (rtos (car  k1)  2  4)  f)
        (write-line (rtos (cadr k1)  2  4)  f)
        (write-line "The 4th point entered for function
 BISECT is:"  f)
        (write-line (rtos (car  k2)  2  4)  f)
        (write-line (rtos (cadr k2)  2  4)  f)
        (setq pt(inters pt1 pt2 pt3 pt4 nil))
        (command "point" pt)
        (setq R1(distance pt k))
        (setq R2(distance pt k1))
        (write-line "Intersection point of 2 bisectors:"  f)
        (write-line (rtos (car  pt)  2  4)  f)
        (write-line (rtos (cadr pt)  2  4)  f)
        (write-line "The distance (R1): " f)
        (write-line (rtos  R1  2  4)  )
        (write-line (rtos  R1  2  4)  f)
        (write-line "The distance (R2): " f)
        (write-line (rtos  R2  2  4)  )
        (write-line (rtos  R2  2  4)  f)
        (close  f)
        (princ)
)


;  Triangle version 1 -- interactive version
;  Max number of positions:  no limit.
;
(defun triang  (n / pt1 pt2 pt3 x y th thd d1 d2 phi phid
psi)
        (setq f1(open "input.dat" "r")
              f2(open "circ.dat" "w")
              pt3(getpoint "\nEnter a point for plotting
triangles(key in or by mouse)")
              x(atof (read-line f1))
              y(atof (read-line f1))
              thd(atof (read-line  f1))
              th(/ (* thd pi) 180.0)
              pt1(list x y)
```

```
              d1(getdist "\nEnter a length for side 1 of the
triangles:")
              pt2(polar pt1 th d1)
              phi(angle pt1 pt3)
              phid(/ (* phi 180.0) pi )
              psi(- th phi)
              d2(distance pt1 pt3)
       )
       (write-line(rtos (car pt3) 2 6) f2)
       (write-line(rtos (cadr pt3) 2 6) f2)
       (command "pline" pt1 pt2 "")
       (command "pline" pt2 pt3 pt1 "")
       (setq n(1- n))
       (repeat  n
            (setq x(atof (read-line  f1))
                  y(atof (read-line  f1))
                  thd(atof (read-line  f1))
                  th(/ (* thd pi) 180.0)
                  pt1(list x y)
                  pt2(polar pt1 th d1)
                  pt3(polar pt1 (- th psi) d2)
            )
            (write-line(rtos (car pt3) 2 6) f2)
            (write-line(rtos (cadr pt3) 2 6) f2)
            (command "pline" pt1 pt2 "")
            (command "pline" pt2 pt3 pt1 "")
       )    ; end of repeat
       close (f1)
       close (f2)
       (princ)
 )


; Triangle version 2 -- to be called by invert() and other
routines.
;
(defun triangle  ()
       (command "pline" a1 b1 ""
                "pline" b1 c1 a1 ""
       )
       (setq phi1(angle a1 c1)
             psi(- th1 phi1)
             dist(distance a1 c1)
             c2(polar a2 (- th2 psi) dist)
             c3(polar a3 (- th3 psi) dist)
       )
       (command "pline" a2 b2 ""
                "pline" b2 c2 a2 ""
                "pline" a3 b3 ""
                "pline" b3 c3 a3 ""
       )
       (princ)
 )
```

```
(defun invert (n)
  (setq f1(open "input.dat" "r")
        f2(open "output.dat" "w")
;        f3(open "test.dat" "w")
        f(getpoint "\nEnter a center point for inversion(key
in or by mouse)")
        x(atof (read-line f1))
        y(atof (read-line f1))
        th1d(atof (read-line  f1))
        th1(/ (* th1d pi) 180.0)
        a1(list x y)
        d(getdist "\nEnter a length for the 1st side of
triangles:")
        b1(polar a1 th1 d)
        phi1(angle a1 f)
        phi1d(/ (* phi1 180.0) pi )
)
  (setq x(atof (read-line  f1))
        y(atof (read-line  f1))
        th2d(atof (read-line  f1))
        th2(/ (* th2d pi) 180.0)
        a2(list x y)
        b2(polar a2 th2 d)
        dist2(distance a2 f)
        phi2(angle a2 f)
        phi2d(/ (* phi2 180.0) pi)
        beta2(+ phi2 (- th1  th2))
        f2p(polar a1 beta2 dist2)
)
(if (= n 2)
  (progn
   (command "pline" a1 b1 "")
   (command "pline" b1 f a1 "")
   (command "pline" a1 b1 "")
   (command "pline" b1 f2p a1 "")
  )
  (progn
  (setq x(atof (read-line  f1))
        y(atof (read-line  f1))
        th3d(atof (read-line  f1))
        th3(/ (* th3d pi) 180.0)
        a3(list x y)
        b3(polar a3 th3 d)
        dist3(distance a3 f)
        phi3(angle a3 f)
        phi3d(/ (* phi3 180.0) pi)
        beta3(+ phi3 (- th1  th3))
        f3p(polar a1 beta3 dist3)
)

;   (command "pline" a1 b1 "")
;   (command "pline" b1 f a1 "")
```

```
;   (command "pline" b1 f2p a1 "")
;   (command "pline" b1 f3p a1 "")

  (setq t1(car f)
        t2(cadr f)
        t3(car f2p)
        t4(cadr f2p)
        t5(car f3p)
        t6(cadr f3p)
        t7(abs  (- t1 t3))
        t8(abs  (- t2 t4))
        t9(abs  (- t1 t5))
        t10(abs (- t2 t6))
        t11(abs (- t3 t5))
        t12(abs (- t4 t6))
  )
  (if (and (<= t7  0.0001) (<= t8  0.0001))      ;for P12
      (write-line "You have entered a pole, a bad point to
invert.") ;exit
      (progn
        (if (and (<= t9 0.0001) (<= t10 0.0001))
;for P13
          (write-line "You have entered a pole, a bad point
to invert.") ;exit
          (progn
            (if (and (<= t11 0.0001) (<= t12 0.0001))
;for P23
               (write-line "You have entered a pole, a bad
point to invert.") ;exit
               (progn
  (setq k1 f
        k2 f2p
  )
      (bisector)
  (setq pt3 pt1
        pt4 pt2
        k1 f3p
  )              ; keep k2=f2p

  (bisector)
  (setq c1(inters pt1 pt2 pt3 pt4 nil)) ; c1--moving pivot
if f is a cent pt
;                         triangles 1, 2 and 3
  (command "pline" a1 b1 "")
  (command "pline" b1 c1 "")
  (command "pline" c1 a1 "")
  (setq phi1(angle a1 c1)
      psi(- th1 phi1)
      dist(distance a1 c1)
      c2(polar a2 (- th2 psi) dist)
  )
  (command "pline" a2 b2  "")
  (command "pline" b2 c2 "")
  (command "pline" c2 a2 "")
```

```
   (setq c3(polar a3 (- th3 psi) dist))
   (command "pline" a3 b3 "")
   (command "pline" b3 c3 "")
   (command "pline" c3 a3 "")
;   (write-line "stop 1"  f3)
;   (write-line(rtos n) f3)
;                           for position 4,5,6..
   (setq n(- n 3))
;   (write-line "stop 2"  f3)
;   (write-line(rtos n) f3)
   (while (> n 0)
      (progn
;           (write-line "stop 3"  f3)
;           (write-line(rtos n) f3)
           (setq x(atof (read-line  f1))
                 y(atof (read-line  f1))
                 th4d(atof (read-line  f1))
                 th4(/ (* th4d pi) 180.0)
                 a4(list x y)
                 b4(polar a4 th4 d)
                 dist4(distance a4 f)
                 phi4(angle a4 f)
                 phi4d(/ (* phi4 180.0) pi)
                 beta4(+ phi4 (- th1  th4))
                 f4p(polar a1 beta4 dist4)
           )
;           (write-line "stop 4"  f3)
;           (write-line (rtos n) f3)
;          (command "pline" a4 b4 f a4 "")      ;to be deleted
;                    triangle 4,5,6....
           (setq c4(polar a4 (- th4 psi) dist))
           (write-line (rtos (car a4)) f2)
           (write-line (rtos (cadr a4)) f2)
           (write-line (rtos (car b4)) f2)
           (write-line (rtos (cadr b4)) f2)
           (write-line (rtos (car c4)) f2)
           (write-line (rtos (cadr c4)) f2)
           (command "pline" a4 b4 "")
           (command "pline" b4 c4 "")
           (command "pline" c4 a4 "")
           (setq n(1- n))
      )    ; end progn
  )    ; end if or while
;                circle
  (command "circle" f (distance  f c1))
  ))))))))     ;end of 4 ifs
  (close f1)
  (close f2)
;  (close f3)

  (princ)
)
```

```
(defun C:filemon (/ ph pos k1 k2 s d gd gr gr1 t1 t2)
      (command)
      (command "shell")
      (command "filemon")
         (setq f(open  "bus.dat" "r")
               ph(atoi(read-line f))    ;num of phases
         )
         (repeat ph
            (setq pos(atoi(read-line  f)))    ;num of pos
               (setq k1(atof(read-line f))
                     k2(atof(read-line f))
                     s (list k1 k2)             ;source point
                     d(atof(read-line  f))    ;half length
               )
            (repeat pos
               (setq gd(atof(read-line f))
                     gr(/ (* gd pi) 180.0)
                     gr1(+ gr pi)
                     t1(polar s gr d)
                     t2(polar s gr1 d)
               )
               (command "pline" s t1 "")
               (command "pline" s t2 "")
            )
         )
         (close f)
(princ)
)


(defun C:ipol_circ(/ t1 t2 t3 k1 k2 k3 k4 k5 k6 c1 c2 c3 d1
d2 d3 d11 d22 d33 f n)
;   (setvar "pdmode" 64)
    (setq f(open "input.dat" "r")
          n(getint"\nHow many phases to plot the Ipole
Circles?")
    )
    (repeat  n
       (setq k1(atof(read-line f))
             k2(atof(read-line f))
             k3(atof(read-line f))
             k4(atof(read-line f))
             k5(atof(read-line f))
             k6(atof(read-line f))
             t1(list k1 k2)
             t2(list k3 k4)
             t3(list k5 k6)
             c1(polar t1 (angle t1 t2) (/ (distance t1 t2)
2))
             c2(polar t1 (angle t1 t3) (/ (distance t1 t3)
2))
```

```
                c3(polar t2 (angle t2 t3) (/ (distance t2 t3)
2))
        )
;       (command "point" t1 )
;       (command "point" t2 )
;       (command "point" t3 )
        (setq d1(distance t1 t2)
              d2(distance t1 t3)
              d3(distance t2 t3)
              d11(+ d1 0.01)
              d22(+ d2 0.01)
              d33(+ d3 0.01)
        )
        (command "donut" d1 d11 c1 "")
        (command "donut" d2 d22 c2 "")
        (command "donut" d3 d33 c3 "")
    )             ; end repeat
    (close  f)
(princ)
)
```

B. Program Filemon

{This program plots Filemon lines for up to 12 lines in 5 groups.}

```
Program  Filemon;

Const
     in_fil = 'input.dat';
     out_fil = 'bus.dat';    {trans data to alisp command
"filemon" to plot}
Var
     inf, out                              :text;
     total_gro, total_pos,i,j,n,len        :integer;
     x, y, thr, thd                        :array[1..12] of
real;
     pos_num                               :array[1..3,1..5]
of integer;
     xs,ys,xc,yc                           :array[1..5] of
real;
     x1,y1,thd1,thr1,x2,y2,thd2,thr2       :real;

gam1,gamd1,alpha1,beta,alpha2,d,c2x,c2y,gam2,del2,del2d,s,t
:real;

Function atan (u,v :real):real;  {u-Numerator, v-
Denominator}
begin
     if      (((u>=0) and (v<0)) or ((u<0) and (v<0)))  then
             atan := arctan(u/v) + pi
     else if ((u>0)and (v=0))    then atan := pi/2.
     else if ((u<0)and(v=0))     then atan:= pi*3./2.
     else if ((u=0)and(v=0))     then writeln('Bad
argument: 0/0 ')
     else    atan := arctan(u/v) ;
end;

BEGIN
     assign (inf,in_fil);
     reset (inf);
     assign (out,out_fil);
     rewrite(out);

     write('Total number of positions: ');
     readln(total_pos);
     write('How many groups of lines? ');
     readln(total_gro);
     writeln(out,total_gro);

     for i := 1 to total_pos do   begin
        readln(inf,x[i]);
        readln(inf,y[i]);
        readln(inf,thd[i]);
        thr[i] := thd[i] * pi/180;
```

```
      end;
      close(inf);

      for i := 1 to total_gro   do   begin
         write('How many positions in line group ',i,'? ');
         readln(n);
         writeln(out,n);
         write('X coord. of center pt.: ');
         readln(xc[i]);
         write('Y coord. of center pt.: ');
         readln(yc[i]);
         write('X coord. of circle (line source) pt.: ');
         readln(xs[i]);
         writeln(out,xs[i]:10:4);
         write('Y coord. of circle (line source) pt.: ');
         readln(ys[i]);
         writeln(out,ys[i]:10:4);

         for  j := 1 to n   do   begin
           writeln('Enter position numbers in line group
','i',' in order');
           writeln('(enter reference position first):');
           readln(pos_num[i,j]);
         end;

         write('The half length of lines to plot: ');
         readln(len);
         writeln(out,len);

         x1 := x[pos_num[i,1]];   {for reference position of
a group of lines}
         y1 := y[pos_num[i,1]];
         thd1 := thd[pos_num[i,1]];    thr1 := thd1
*pi/180.0;

                                   {calc at reference pos.}
         d       := sqrt(sqr(ys[i]-y1)+sqr(xs[i]-x1));
         s       := ys[i]-yc[i];
         t       := xs[i]-xc[i];
         gam1    := atan(s,t);
         s       := ys[i]-y1;
         t       := xs[i]-x1;
         alpha1  := atan(s,t);
         beta    := alpha1 - thr[i];
         gamd1   := gam1 * 180.0/pi;
         writeln(out,gamd1:10:4);

         for j := 2 to n do   begin
             x2 := x[pos_num[i,j]];
             y2 := y[pos_num[i,j]];
             thd2 := thd[pos_num[i,j]];    thr2 := thd2
*pi/180.0;

             alpha2 := thr2 + beta;
```

```
            C2x     := x2 + d*cos(alpha2);
            C2y     := y2 + d*sin(alpha2);
            s       := C2y-yc[i];
            t       := C2x-xc[i];
            gam2    := atan(s,t);
            del2    := gam2 + (thr1-thr2);
            del2d   := del2 * 180./pi;
            writeln(out,del2d:10:4);
        end;
     end;
     close(out);
END.
```

C. Program MP_3_3.PAS

```
Program mp_3_3;
   {This program calculates a center point curve for 123-
456,}
   {by choosing solutions at pole as initial solutions}

Const
  In_fil_name  = 'MP_3_3.DAT';
  Layer_name = 'centpt';
  Out_fil_name = 'centpt.dxf';
  Out_fil_1 = 'MP_3_3.o1';  {center pts for plotting in
ACAD}
  Out_fil_2 = 'MP_3_3.o2';  {complete Info, detail at every
pt}
  Out_fil_3 = 'MP_3_3.o3';  {center pts, L<=Maxlen, detail
info at every pt}
  Out_fil_4 = 'MP_3_3.o4';  {center pts, L<=Maxlen, but
coord. of pts only}
  Out_fil_5 = 'MP_3_3.o5';  {data for plotting ipole-
circles}
  Out_fil_6 = 'MP_3_3.o6';  {all circle pts at pos 1, for
Filemon plotting}
  Out_fil_7 = 'MP_3_3.o7';  {all circle pts at pos 4, for
Filemon plotting}
  Out_fil_8 = 'MP_3_3.o8';  {centpts, circpts at pos1,
circpts at pos2}
  Out_fil_9 = 'MP_3_3.o9';  {plot curves with diff. appear.
of points }
  positions = 6;
  EPS = 0.0001;
  maxcount = 100;
  maxpoints = 200;

Var
  In_fil, o1, o2, o3, o4, o5, o6, o7, o8, o9, out_fil
:text;
  x,   y,   xx, yy, th         :array[1..6] of real;
  thr, Bx,  By, c,  s          :array[1..6] of real;
  Px,  Py,  Ix, Iy             :array[1..6,1..6] of real;
  ii,  jj,  kk, ll, mm, nn, oo:integer;
  i,   j,   j1, j2, k,  phase :integer;
  P,   Q,   p0, q0, R,  R0     :real;
  dR,  dR0, p1, q1, p2, q2     :real;
  d1,d4,maxlen                 :real;
  cc,dd,ee,ff                  :real;
  pp, qq                       :array[1..2,1..2] of real;
  x1,y1,x2,y2                  :array[1..6] of real;
                                 {abs. coord. for mp
obtained}
  R_check                      :array[1..6] of real;
                                 {crank lengths for checking}
  done, large, abort, tired    :boolean;
  pole_OK                      :boolean;
```

```
  G,   D                          :array[1..6] of real;
  V                               :array [1..6,1..6] of real;
{$I POLE.PAS}   {[12]}
{$I C_MP_R.PAS}


Procedure Get_data;
  {gets position information}
  {and calculates position cosines and sines}
  Var
    i : integer;
  Begin
    Assign (in_fil,in_fil_name);
    Reset  (in_fil);
    for i := 1 to 6 do
      begin
        Readln (in_fil, x[i]);
        Readln (in_fil, y[i]);
        Readln (in_fil, th[i]);
        thr[i] := th[i] * pi/180;
        c[i] := cos(thr[i]); s[i] := sin(thr[i]);
      end;
    close (in_fil);
  End;

Procedure Load_array;
  {loads the partial derivative array and the function}
  {array for 123-456 problem}
  Var
    T1, T2 :real;
    i :integer;
  begin
    for i := 1 to 3 do  {load partial array}
    begin
      T1 := x[i] + p1*c[i] - q1*s[i] - P;
      T2 := y[i] + p1*s[i] + q1*c[i] - Q;
      V[i,1] := -2 * T1;
      V[i,2] := -2 * T2;
      V[i,3] :=  2 * c[i] * T1 + 2 * s[i] * T2;
      V[i,4] := -2 * s[i] * T1 + 2 * c[i] * T2;
      V[i,5] := 0;
      V[i,6] := 0;
      G[i] := R*R - T1*T1 - T2*T2;
    end;
    for i := 4 to 6 do  {load partial array}
    begin
      T1 := x[i] + p2*c[i] - q2*s[i] - P;
      T2 := y[i] + p2*s[i] + q2*c[i] - Q;
      V[i,1] := -2 * T1;
      V[i,2] := -2 * T2;
      V[i,3] := 0;
      V[i,4] := 0;
      V[i,5] :=  2 * c[i] * T1 + 2 * s[i] * T2;
      V[i,6] := -2 * s[i] * T1 + 2 * c[i] * T2;
```

```
      G[i] := R*R - T1*T1 - T2*T2;
    end;
  end;

{$I CALC.PAS}

Procedure Open_fil;
Begin
  assign(o1,Out_fil_1);
  rewrite(o1);
  assign(o2,Out_fil_2);
  rewrite(o2);
  assign(o3,Out_fil_3);
  rewrite(o3);
  assign(o4,Out_fil_4);
  rewrite(o4);
  assign(o5,Out_fil_5);
  rewrite(o5);
  assign(o6,Out_fil_6);
  rewrite(o6);
  assign(o7,Out_fil_7);
  rewrite(o7);
  assign(o8,Out_fil_8);
  rewrite(o8);
  assign(o9,Out_fil_9);
  rewrite(o9);
End;

Procedure Close_fil;
Begin
  close(o1);
  close(o2);
  close(o3);
  close(o4);
  close(o5);
  close(o6);
  close(o7);
  close(o8);
  close(o9);
End;

Procedure Output;
          {test number, pole number at which calculation is
carrying on,}
          {input data (given data), poles & Ipoles, output
good points  }
          {are included in this file.}
Var
   i:  integer;
Begin
writeln(o2,'------------------------------------------------
---------');
writeln(o2,'Output file #2 (MP_3_3.o2) for adj. moving pivot
123-456.');
```

```
writeln(o2,'------------------------------------------------
---------');
writeln(o2,'[Input data:]');
for  i := 1 to 6 do
     begin
               write(o2,x[i]:10:4);
               write(o2,y[i]:10:4);
               writeln(o2,th[i]:10:4);
     end;
writeln(o3,'------------------------------------------------
-----------------');
writeln(o3,'The following selected center points satisfies
the length');
writeln(o3,'    requirement: length <=',maxlen:8:2);
writeln(o3,'------------------------------------------------
-----------------');
writeln(o4,'------------------------------------------------
-----------------');
writeln(o4,'The circle points of the following selected
center points ');
writeln(o4,'satisfies the length requirement: length
<=',maxlen:8:2);
writeln(o4,'------------------------------------------------
-----------------');
writeln(o8,'------------------------------------------------
-----------------');
writeln(o8,'   The center points    The circle points    The
circle points ');
writeln(o8,'                         (ABS)at position 1
(ABS)at position 4');
writeln(o8,'------------------------------------------------
-----------------');
End;

BEGIN
  Get_data;
  Open_data;
  Beg_poly;
  Open_fil;
  Pole(6);
  Print_pol;
  Ipole(6);
  Print_ipol;
  writeln('Enter a length (The maximum desired distance
between a circle');
  write('  point and the given points (x[1],y[1]), and
(x[4],y[4])):');
  readln(maxlen);
  write('Processing......Please Wait');
  Output;
  for phase := 0 to 1 do   begin
      ll := 5-3*phase;
      mm := 6-3*phase;
      nn := 4-3*phase;
```

```
      oo := 1+3*phase;
      for i := 1 to 2 do   begin
        ii := i+3*phase;
        j1 := ii + 1;   j2 := 3 + 3*phase;
        for jj := j1 to j2 do   begin
          kk := 5 + 6*phase -jj;
 {           writeln(o2,'     ii     jj     kk     ll     mm
nn      oo');}
 {           writeln(o2,ii:7, jj:7, kk:7, ll:7, mm:7, nn:7,
oo:7);}

          Calc_at_pole;
          if pole_OK then   begin write(' ->'); Cpoints;
end;
        end;   {jj}
      end;  {i}
  end;    {phase}
  writeln(o1,'nil');
  writeln(o6,'nil');
  writeln(o7,'nil');
  End_poly;
  Close_data;
  close_fil;
END.
{This program is created by modifying program FP_3_3.PAS[12]
on the basis of equations developed in chapter 3}
```

D. Program C_MP_R.PAS

```
Procedure Trans_to_abs;
{transforms p1,q1,p2,q2 calculated by numerical method to  }
{its abs. coord. for checking and plotting}
Var
  i:  integer;

Begin
     for i := 1 to 3 do begin
       x1[i] := x[i] + p1*c[i] - q1*s[i];
       y1[i] := y[i] + p1*s[i] + q1*c[i];
     end;    {end for}
     for i := 4 to 6 do begin
       x2[i] := x[i] + p2*c[i] - q2*s[i];
       y2[i] := y[i] + p2*s[i] + q2*c[i];
     end;    {end for}
End;

Procedure Distance;
Begin
     d1 := sqrt(sqr(x[1]-x1[1])+sqr(y[1]-y1[1]));
     d4 := sqrt(sqr(x[4]-x2[4])+sqr(y[4]-y2[4]));
End;

Procedure Calc_r;
 {calculates crank lengths for each moving pivot generated}
 {by numerical method for positions 1,2,3 & 4,5,6}
 {for checking }
 Var
   i:  integer;

Begin
      for i := 1 to 3 do
        R_check[i] := sqrt(sqr(x1[i]-P)+sqr(y1[i]-Q));
      for i := 4 to 6 do
        R_check[i] := sqrt(sqr(x2[i]-P)+sqr(y2[i]-Q));

End;

Procedure Mp_out(points: integer);
 Var
   i:  integer;
Begin
                Data_out(P,Q);
                writeln(o1,P:10:4,',',Q:10:4);
                writeln(o9,P:10:4);
                writeln(o9,Q:10:4);
                write(o8,P:10:4,',',Q:10:4);
                writeln(o2,'');

writeln(o2,'**********************************************')
;
                writeln(o2,'[Good point ',points+1,']');
```

```
                write(o2,'     P          Q         p1
q1         p2');
                writeln(o2,'         q2         R');

writeln(o2,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);
                Trans_to_abs;
                Calc_r;
       writeln(o2,'-----------------------------------------
----');
       writeln(o2,'position#    x1 or x2    y1 or y2    R');
       writeln(o2,'-----------------------------------------
----');

      for  i := 1 to 3 do
        begin
          writeln(o2,'      ',i,'
',x1[i]:10:4,y1[i]:10:4,R_check[i]:10:4);
        end;
      for  i := 4 to 6 do
        begin
          writeln(o2,'      ',i,'
',x2[i]:10:4,y2[i]:10:4,R_check[i]:10:4);
        end;
        writeln(o6,x1[1]:10:4,',',y1[1]:10:4);
        writeln(o7,x2[4]:10:4,',',y2[4]:10:4);

writeln(o8,x1[1]:10:4,',',y1[1]:10:4,x2[4]:10:4,',',y2[4]:10
:4);
        Distance;
        if (d1 <= maxlen)and(d4 <= maxlen)  then
        begin
        writeln(o3,'');

writeln(o3,'*********************************************')
;
                writeln(o3,'[Good point ',points+1,']');
                write(o3,'     P          Q         p1
q1         p2');
                writeln(o3,'         q2         R');

writeln(o3,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);
       writeln(o3,'-----------------------------------------
----');
       writeln(o3,'position#    x1 or x2    y1 or y2    R');
       writeln(o3,'-----------------------------------------
----');
       writeln(o4,P:10:4,',',Q:10:4);
      for  i := 1 to 3 do
        begin
          writeln(o3,'      ',i,'
',x1[i]:10:4,y1[i]:10:4,R_check[i]:10:4);
        end;
```

```
      for  i := 4 to 6 do
        begin
          writeln(o3,'      ',i,'
',x2[i]:10:4,y2[i]:10:4,R_check[i]:10:4);
        end;
        end;
End;

Procedure Mp_out_pole(i3:integer);
 Var
   i:  integer;
Begin

     {the initial solution point--the pole }
            Data_out(P,Q);
            writeln(o9,P:10:4);
            writeln(o9,Q:10:4);
            writeln(o1,P:10:4,',',Q:10:4);
            write(o8,P:10:4,',',Q:10:4);
            writeln(o2,'------------------------------------
-------');
            write(o2,'[At P',ii,jj,': Branch #',i3,',
dR=',dR:6:2,', ');
            writeln(o2,'Starting a branch of solutions:]');
            writeln(o2,'------------------------------------
-------');
            write(o2,'     P          Q          p1         q1
p2');
            writeln(o2,'          q2         R');

writeln(o2,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);
            writeln(o2,'------------------------------------
-------');
            writeln(o2,'[Good center points:]');
        Trans_to_abs;
        Calc_r;
       writeln(o2,'---------------------------------------
----');
       writeln(o2,'position  x1(p1)|x2(p2)   y1(q1)|y2(q2)
R_check');
       writeln(o2,'---------------------------------------
----');

      for  i := 1 to 3 do
        begin
          write(o2,'   P',ii,jj,'   ',x1[i]:10:4,'       ');
          writeln(o2,y1[i]:10:4,'   ',R_check[i]:10:4);
        end;
      for  i := 4 to 6 do
        begin
          write(o2,'   P',ii,jj,'   ',x2[i]:10:4,'       ');
          writeln(o2,y2[i]:10:4,'   ',R_check[i]:10:4);
        end;
```

```
        writeln(o2,'---------------------------------------
---');
        writeln(o6,x1[1]:10:4,',',y1[1]:10:4);
        writeln(o7,x2[4]:10:4,',',y2[4]:10:4);

writeln(o8,x1[1]:10:4,',',y1[1]:10:4,x2[4]:10:4,',',y2[4]:10
:4);
        Distance;
        if (d1 <= maxlen)and(d4 <= maxlen)  then
        begin
                {the initial solution point--the pole }
             writeln(o3,'-------------------------------------
-------');
             write(o3,'At P',ii,jj,': Branch #',i3,',
dR=',dR:6:2,', ');
             writeln(o3,'Starting a branch of solutions:');
             writeln(o3,'-------------------------------------
-------');
             write(o3,'     P         Q         p1         q1
p2');
             writeln(o3,'         q2         R');

writeln(o3,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);

        writeln(o3,'---------------------------------------
----');
        writeln(o3,'position  x1(p1)|x2(p2)   y1(q1)|y2(q2)
R_check');
        writeln(o3,'---------------------------------------
----');
        writeln(o4,P:10:4,',',Q:10:4);
       for  i := 1 to 3 do
         begin
           write(o3,'   P',ii,jj,'   ',x1[i]:10:4,'      ');
           writeln(o3,y1[i]:10:4,'   ',R_check[i]:10:4);
         end;
       for  i := 4 to 6 do
         begin
           write(o3,'   P',ii,jj,'   ',x2[i]:10:4,'      ');
           writeln(o3,y2[i]:10:4,'   ',R_check[i]:10:4);
         end;
        writeln(o3,'---------------------------------------
---');
        end;
End;
```

E. Program CALC.PAS

```
Procedure Calc_at_pole;
  {finds solution set to the adjustable moving pivot
problem,}

  var
    K1, K2, K3, arg :real;
    L, M, N         :array [1..6,1..6] of real;
    i, j, j1, pos1  :integer;
  Begin
      {calculate at a pole which is chosen as a center
point}
      {P,Q--abs. coord. of the center points}
      P := Px[ii,jj];  Q := Py[ii,jj];

            {calculates L,M,Ns for indexes [1,2], [1,3],
[4,5], [4,6]}
      i := 1;
      for j := 2 to 3 do
      begin
        L[i,j] := P*(-c[i]+c[j])+Q*(-s[i]+s[j])
                  +x[i]*c[i]-x[j]*c[j]+y[i]*s[i]-y[j]*s[j];
        M[i,j] := P*(s[i]-s[j])+Q*(-c[i]+c[j])
                  -x[i]*s[i]+x[j]*s[j]+y[i]*c[i]-y[j]*c[j];
        N[i,j] := P*(-x[i]+x[j])+Q*(-y[i]+y[j])
                  +(x[i]*x[i]+y[i]*y[i]-x[j]*x[j]-
y[j]*y[j])/2;
      end; {for j}

      i := 4;
      for j := 5 to 6 do
      begin
        L[i,j] := P*(-c[i]+c[j])+Q*(-s[i]+s[j])
                  +x[i]*c[i]-x[j]*c[j]+y[i]*s[i]-y[j]*s[j];
        M[i,j] := P*(s[i]-s[j])+Q*(-c[i]+c[j])
                  -x[i]*s[i]+x[j]*s[j]+y[i]*c[i]-y[j]*c[j];
        N[i,j] := P*(-x[i]+x[j])+Q*(-y[i]+y[j])
                  +(x[i]*x[i]+y[i]*y[i]-x[j]*x[j]-
y[j]*y[j])/2;
      end; {for j}

      { solve for p2, q2 ( pp[2, ], qq[2, ] )when p12, p13,
p23 is picked }
      { solve for p1, q1 ( pp[2, ], qq[2, ] )when p45, p46,
p56 is picked }
      { p2,q2--the relative coord. of the moving pivot for
ph 2.}
      { p1,q1--the relative coord. of the moving pivot for
ph 1.}

    pp[2,1] := (M[nn,ll]*N[nn,mm] - M[nn,mm]*N[nn,ll])
             / (L[nn,ll]*M[nn,mm] - L[nn,mm]*M[nn,ll]);
    qq[2,1] := (L[nn,mm]*N[nn,ll] - L[nn,ll]*N[nn,mm])
```

```
              / (L[nn,ll]*M[nn,mm] - L[nn,mm]*M[nn,ll]);
     pp[2,2] := pp[2,1];
     qq[2,2] := qq[2,1];

    {crank length}
     xx[nn] := x[nn]+pp[2,1]*c[nn]-qq[2,1]*s[nn];
     yy[nn] := y[nn]+pp[2,1]*s[nn]+qq[2,1]*c[nn];
     R := Sqrt (sqr(xx[nn]-P) + sqr(yy[nn]-Q));


    { solve for p1,q1}
      cc := c[oo]+L[oo,kk]*s[oo]/M[oo,kk];
      dd := X[oo]+N[oo,kk]*s[oo]/M[oo,kk];
      ee := s[oo]-L[oo,kk]*c[oo]/M[oo,kk];
      ff := Y[oo]-N[oo,kk]*c[oo]/M[oo,kk];

     K1 := sqr(cc)+sqr(ee);
     K2 := 2*cc*(dd-P)+2*ee*(ff-Q);
     K3 := sqr(dd-P)+sqr(ff-Q)-sqr(R);
     arg := K2*K2 - 4*K1*K3;
     if arg >= 0 then
     begin
       pole_OK := true;
       pp[1,1] :=( -K2 + sqrt(arg)) / (2*K1);
       pp[1,2] :=( -K2 - sqrt(arg)) / (2*K1);
       qq[1,1] := - (L[oo,kk]*pp[1,1] + N[oo,kk])/M[oo,kk];
       qq[1,2] := - (L[oo,kk]*pp[1,2] + N[oo,kk])/M[oo,kk];
writeln(o2,'----------------------------------------');
writeln(o2,'[2 sets of p1, q1, p2, q2 at Pole
P',ii,jj,':]');
writeln(o2,'    pp[1,1]    ','qq[1,1]    ','pp[2,1]
','qq[2,1]' );
writeln(o2,
pp[1,1]:10:4,qq[1,1]:10:4,pp[2,1]:10:4,qq[2,1]:10:4 );
writeln(o2,'    pp[1,2]    ','qq[1,2]    ','pp[2,2]
','qq[2,2]' );
writeln(o2,
pp[1,2]:10:4,qq[1,2]:10:4,pp[2,2]:10:4,qq[2,2]:10:4 );
writeln(o2,'----------------------------------------');
     end
       else begin
          writeln(o2,'arg < 0, pole_OK := false, at Pole:
P',ii,jj);
          pole_OK := false;
       end;
  End;


Procedure Cpoints;
  {calculates center points for the adjustable moving pivot
problem.}

  var
```

```
    z, h, i, j, k, i1, i2, j2,i3,j3, count, times, points
:integer;
    XP, YP, PL, QL, p1L, q1L, p2L, q2L, RR :real;
  begin
    RR := R;   {original value of R saved here.}
      for i3 := 1 to 2 do      begin     {2 branches of
solutions}
        for j3 := 1 to 2 do     begin  { R inc. or dec. }
          P := Px[ii,jj];  Q := Py[ii,jj];  R := RR;  {back
to starting values}
          if   phase = 0  then     begin
             p1 := pp[1,i3];   q1 := qq[1,i3];
             p2 := pp[2,i3];   q2 := qq[2,i3];
          end
          else  begin
             p2 := pp[1,i3];   q2 := qq[1,i3];
             p1 := pp[2,i3];   q1 := qq[2,i3];
          end;       { if phase}       { end of backing to
starting value. }
writeln(o2,'######################');
writeln(o2,'[Numerical method applied to the following set
of data:]');
writeln(o2,'     p1        q1        p2        q2        P
Q');
writeln(o2,p1:10:4,q1:10:4,p2:10:4,q2:10:4,P:10:4,Q:10:4 );

          dR := 0.1;
          if j3 = 2 then    dR := -0.1;

           PL := P;   p1L := p1;   p2L := p2;
           QL := Q;   q1L := q1;   q2L := q2;

         Mp_out_pole(i3);

         times := 0;   points := 0;   tired := false;
         Repeat {until tired}
           R := R + dR;
           count := 0;
           Repeat {until done or large}
             large := false;
             Load_Array;
             done := true;

             for i := 1 to 6 do

               if (abs(G[i]) >eps) then done := false;
             if not done then
             begin
               Sim_Eq;
               P  := P  + D[1];   Q  := Q  + D[2];
               p1 := p1 + D[3];   q1 := q1 + D[4];
               p2 := p2 + D[5];   q2 := q2 + D[6];
             end
             else    begin
```

```
              done := true;

              Mp_out(points);

              PL := P;   p1L := p1;   p2L := p2;
              QL := Q;   q1L := q1;   q2L := q2;
              points := points + 1;
            end;
            if abort then count := maxcount;
            if count >= maxcount then large := true;
            count := count +1;
          Until done or large;
          if (not done) and large then
          begin
            times := times + 1;
            if times = 3 then
              tired := true
            else
            begin
              points := 0;  {use maxpoints at each dR}
              R := R - dR;   {back to last good point}
              dR := dR / 10; {try again with smaller dR}
              writeln(o2,'');
              writeln(o2,'dR=dR/10= ',dR:8:4,'  starting
at here:');
              P := PL;   p1 := p1L;   p2 := p2L;
              Q := QL;   q1 := q1L;   q2 := q2L;
            end;   {if times}
          end; {if large}
          if points >= maxpoints then tired := true;
        Until tired;
      end;    {for j3}
     end;     {for i3}
  End;
{ This program is created by modifying FP_PROCS.PAS[12] on
the basis of the equations developed for the synthesis
problem MP_3_3}
```

F. Program MP_3_3_1.PAS

```
Program mp_3_3_1;
   {This program calculates a center points for 123-145}

Const
   In_fil_name  = 'mp_3_3_1.DAT';
   Out_fil_1 = 'mp_3_3_1.o1';
   Out_fil_2 = 'mp_3_3_1.o2';
   out_fil_9 = 'mp_3_3_1.o9';
   positions = 6;
   EPS = 0.0001;
   maxcount = 200;
   maxpoints = 150;

Var
   Out_fil, In_fil, o1, o2, o9       :text;
   x,    y,    xx, yy, th           :array[1..6] of real;
   thr, Bx,   By, c,  s             :array[1..6] of real;
   Px,  Py,   Ix, Iy                :array[1..6,1..6] of real;
   ii,  jj,   kk, ll, mm, nn, oo:integer;
   i,   j,    j1, j2, k,  phase :integer;
   P,   Q,    p0, q0, R,  R0        :real;
   dR,  dR0, p1, q1, p2, q2         :real;
   cc,dd,ee,ff                      :real;
   pp, qq                           :array[1..2,1..2] of real;
   x1,y1,x2,y2                      :array[1..6] of real;
                                      {abs. coord. for mp
obtained}
   R_check                          :array[1..6] of real;
                                      {crank lengths for checking}
   done, large, abort, tired        :boolean;
   pole_OK                          :boolean;
   G,   D                           :array[1..6] of real;
   V                                :array [1..6,1..6] of real;

{$I POLES5.PAS}
{$I C_MP_R_1.PAS}

Procedure Get_data;
   {gets position information}
   {and calculates position cosines and sines}
   Var
     i : integer;
   Begin
     Assign (In_fil,In_fil_name);
     Reset  (In_fil);
     for i := 1 to 5 do
       begin
         Readln (In_fil, x[i]);
         Readln (In_fil, y[i]);
         Readln (In_fil, th[i]);
         thr[i] := th[i] * pi/180;
         c[i] := cos(thr[i]); s[i] := sin(thr[i]);
```

```
      end;
    Close (In_fil);
  End;

procedure Load_Array;
  {loads the partial derivative array and the function}
  {array for 123-456 problem}
  var
    T1, T2 :real;
    i :integer;
  begin
    for i := 1 to 3 do  {load partial array}
    begin
      T1 := x[i] + p1*c[i] - q1*s[i] - P;
      T2 := y[i] + p1*s[i] + q1*c[i] - Q;
      V[i,1] := -2 * T1;
      V[i,2] := -2 * T2;
      V[i,3] :=  2 * c[i] * T1 + 2 * s[i] * T2;
      V[i,4] := -2 * s[i] * T1 + 2 * c[i] * T2;
      V[i,5] := 0;
      V[i,6] := 0;
      G[i] := R*R - T1*T1 - T2*T2;
    end;
    for i := 4 to 5 do  {load partial array}
    begin
      T1 := x[i] + p2*c[i] - q2*s[i] - P;
      T2 := y[i] + p2*s[i] + q2*c[i] - Q;
      V[i,1] := -2 * T1;
      V[i,2] := -2 * T2;
      V[i,3] := 0;
      V[i,4] := 0;
      V[i,5] :=  2 * c[i] * T1 + 2 * s[i] * T2;
      V[i,6] := -2 * s[i] * T1 + 2 * c[i] * T2;
      G[i] := R*R - T1*T1 - T2*T2;
    end;
                              { for position 1 of the 2nd
phase}
      T1 := x[1] + p2*c[1] - q2*s[1] - P;
      T2 := y[1] + p2*s[1] + q2*c[1] - Q;
      V[6,1] := -2 * T1;
      V[6,2] := -2 * T2;
      V[6,3] := 0;
      V[6,4] := 0;
      V[6,5] :=  2 * c[1] * T1 + 2 * s[1] * T2;
      V[6,6] := -2 * s[1] * T1 + 2 * c[1] * T2;
      G[6] := R*R - T1*T1 - T2*T2;
  end;

{$I CALC331.PAS}

Procedure output;
          {test number, pole number at which calculation is
carrying on,}
```

```
          {input data (given data), poles & Ipoles, output
good points  }
          {are included in this file.}
Var
   i:  integer;
Begin
writeln(o2,'---------------------------------------');
writeln(o2,'mp_3_3_1  for adj. moving pivot problem 123-
145.');
writeln(o2,'---------------------------------------');
writeln(o2,'Input data:');
for  i := 1 to 6 do
     begin
               write(o2,x[i]:10:4);
               write(o2,y[i]:10:4);
               writeln(o2,th[i]:10:4);
     end;
     writeln(o2,'--------------------------');
End;

BEGIN
  Get_data;

  Assign(o1,Out_fil_1);
  rewrite(o1);
  Assign(o9,Out_fil_9);
  rewrite(o9);
  Assign(o2,Out_fil_2);
  rewrite(o2);

  output;

  Poles5(5);

  oo := 1;  nn := 1;
  for phase := 0 to 1 do   begin
     ll := 4-2*phase;
     mm := 5-2*phase;
     for i := 1 to 2 do   begin
        ii := i;
        if phase = 1  then ii := i * i;
        j1 := ii + 1;   j2 := 3;
        if phase = 1  then   begin  j1 := 3+i; j2 := 5;
end;
        for jj := j1 to j2 do   begin
          kk := 5 + 4*phase -jj;
          writeln(o2,'     ii      jj      kk      ll      mm
nn      oo');
          writeln(o2,ii:7, jj:7, kk:7, ll:7, mm:7, nn:7,
oo:7);

          Calc_at_pole;
          if pole_OK then    Cpoints;
        end;   {jj}
```

```
      end;  {i}
  end;    {phase}
  writeln(o1,'nil');
  writeln(o9,'nil');
  close(o1);
  close(o2);
  close(o9);
END.
{This program is created by modifying program
FP_3_3_1.PAS[12] on the basis of the equations developed for
the synthesis problem MP_3_3_1}
```

G. Program C_MP_R_1.PAS

```
Procedure Trans_to_abs_1;
    {transforms p1,q1,p2,q2 calculated by numerical method
to  }
    {its abs. coord. for checking and plotting}
    {for 123-145 , 1 shared at 1 or 3}
Var
  i:  integer;

Begin
                                      {for pos. 1,2,3 of phase 1}
     for i := 1 to 3 do begin
       x1[i] := x[i] + p1*c[i] - q1*s[i];
       y1[i] := y[i] + p1*s[i] + q1*c[i];
     end;    {end for}
                                      {for pos. 1 of phase 2}
     x2[4] := x[1] + p2*c[1] - q2*s[1];
     y2[4] := y[1] + p2*s[1] + q2*c[1];
                                      {for pos. 4 ane 5 of phase
2}
                    {note that index of x2[5], y2[5], x2[6],
y2[6] are 5 & 6}
     for i := 5 to 6 do begin
       x2[i] := x[i-1] + p2*c[i-1] - q2*s[i-1];
       y2[i] := y[i-1] + p2*s[i-1] + q2*c[i-1];
     end;    {end for}
End;

Procedure Calc_r_1;
 {calculates crank lengths for each moving pivot generated}
 {by numerical method for positions 1,2,3 & 4,5,6}
 {for checking }
 Var
   i:  integer;

Begin
      for i := 1 to 3 do            {for pos.1,2 3 of ph. 1}
        R_check[i] := sqrt(sqr(x1[i]-P)+sqr(y1[i]-Q));
                                    {for pos. 1 of ph. 2}
        R_check[4] := sqrt(sqr(x2[4]-P)+sqr(y2[4]-Q));
                                    {for pos. 4,5 of ph. 2}
                                    {note that i = 5, 6 }
      for i := 5 to 6 do
      R_check[i] := sqrt(sqr(x2[i]-P)+sqr(y2[i]-Q));
End;

Procedure Mp_out;
 Var
   i:  integer;
Begin
      writeln(o2,'-----------------------------------------
----');
      writeln(o2,'position#    x1 or x2    y1 or y2    R');
```

```
        writeln(o2,'-----------------------------------------
----');

      for  i := 1 to 3 do
        begin
          writeln(o2,'      ',i,'
',x1[i]:10:4,y1[i]:10:4,R_check[i]:10:4);
        end;
      for  i := 4 to 6 do   {for pos. 1, 4, 5 of ph. 2,
note: i=4,5,6}
        begin
          writeln(o2,'      ',i,'
',x2[i]:10:4,y2[i]:10:4,R_check[i]:10:4);
        end;
End;

Procedure Mp_out_pole;
 Var
   i:  integer;
Begin
       writeln(o2,'-----------------------------------------
----');
       writeln(o2,'position  x1(p1)|x2(p2)   y1(q1)|y2(q2)
R_check');
       writeln(o2,'-----------------------------------------
----');

      for  i := 1 to 3 do
        begin
          write(o2,'P',ii,jj,'        ',x1[i]:10:4,'    ');
          writeln(o2,y1[i]:10:4,'    ',R_check[i]:10:4);
        end;
      for  i := 4 to 6 do
        begin
          write(o2,'P',ii,jj,'        ',x2[i]:10:4,'    ');
          writeln(o2,y2[i]:10:4,'    ',R_check[i]:10:4);
        end;
End;
```

H. Program CALC331.PAS

```
procedure Calc_at_pole;
  {finds solution set to the adjustable moving pivot
problem}

  var
    K1, K2, K3, arg :real;
    L, M, N         :array [1..6,1..6] of real;
    i, j, j1, pos1  :integer;
  Begin
            {calculate at a pole which is chosen as a center
point}
            {P,Q--abs. coord. of the center points}
      P := Px[ii,jj];  Q := Py[ii,jj];

            {calculates L,M,Ns for indexes [1,2], [1,3],
[1,4], [1,5]}
      i := 1;
      for j := 2 to 5 do
      begin
        L[i,j] := P*(-c[i]+c[j])+Q*(-s[i]+s[j])
                  +x[i]*c[i]-x[j]*c[j]+y[i]*s[i]-y[j]*s[j];
        M[i,j] := P*(s[i]-s[j])+Q*(-c[i]+c[j])
                  -x[i]*s[i]+x[j]*s[j]+y[i]*c[i]-y[j]*c[j];
        N[i,j] := P*(-x[i]+x[j])+Q*(-y[i]+y[j])
                  +(x[i]*x[i]+y[i]*y[i]-x[j]*x[j]-
y[j]*y[j])/2;
      end; {for j}

      { solve for p2, q2 (represented by pp[2, ], qq[2, ]
)when p12, p13, p23 is picked }
      { solve for p1, q1 (also represented by pp[2, ], qq[2,
] )when p14, p15, p45 is picked }
      { p2,q2--the relative coord. of the moving pivot for
ph 2.}
      { p1,q1--the relative coord. of the moving pivot for
ph 1.}
    pp[2,1] := (M[nn,ll]*N[nn,mm] - M[nn,mm]*N[nn,ll])
             / (L[nn,ll]*M[nn,mm] - L[nn,mm]*M[nn,ll]);
    qq[2,1] := (L[nn,mm]*N[nn,ll] - L[nn,ll]*N[nn,mm])
             / (L[nn,ll]*M[nn,mm] - L[nn,mm]*M[nn,ll]);
    pp[2,2] := pp[2,1];
    qq[2,2] := qq[2,1];

            {crank length}
            {nn always = 1 in this 123-145 problem}
            {nn = 4(phase 0), 1(phase 1) in 123-456 problem}
    xx[nn] := x[nn]+pp[2,1]*c[nn]-qq[2,1]*s[nn];
    yy[nn] := y[nn]+pp[2,1]*s[nn]+qq[2,1]*c[nn];
    R := Sqrt (sqr(xx[nn]-P) + sqr(yy[nn]-Q));

                    { solve for p1,q1 when p12,p13,p23 is
picked}
```

```
                    { solve for p2,q2 when p14,p15,p45 is
picked}
     cc := c[oo]+L[oo,kk]*s[oo]/M[oo,kk];
     dd := x[oo]+N[oo,kk]*s[oo]/M[oo,kk];
     ee := s[oo]-L[oo,kk]*c[oo]/M[oo,kk];
     ff := y[oo]-N[oo,kk]*c[oo]/M[oo,kk];

    K1 := sqr(cc)+sqr(ee);
    K2 := 2*cc*(dd-P)+2*ee*(ff-Q);
    K3 := sqr(dd-P)+sqr(ff-Q)-sqr(R);
    arg := K2*K2 - 4*K1*K3;
    if arg >= 0 then   begin
      pole_OK := true;
      pp[1,1] :=( -K2 + sqrt(arg)) / (2*K1);
      pp[1,2] :=( -K2 - sqrt(arg)) / (2*K1);
      qq[1,1] := - (L[oo,kk]*pp[1,1] + N[oo,kk])/M[oo,kk];
      qq[1,2] := - (L[oo,kk]*pp[1,2] + N[oo,kk])/M[oo,kk];
    end
    else      pole_OK := false;
writeln(o2,'stop1: pp[1,1]   ','qq[1,1]   ','pp[2,1]
','qq[2,1]' );
writeln(o2,'    ',
pp[1,1]:10:4,qq[1,1]:10:4,pp[2,1]:10:4,qq[2,1]:10:4 );
writeln(o2,'stop2: pp[1,2]   ','qq[1,2]   ','pp[2,2]
','qq[2,2]' );
writeln(o2,'    ',
pp[1,2]:10:4,qq[1,2]:10:4,pp[2,2]:10:4,qq[2,2]:10:4 );
  End;


procedure Cpoints;
  {calculates center points for the adjustable moving}
  {pivot problem 123-145}
  var
    z, h, i, j, k, i1, i2, j2,i3,j3, count, times, points
:integer;
    XP, YP, PL, QL, p1L, q1L, p2L, q2L, RR :real;
  begin
     RR := R;   {original value of R saved here.}
        for i3 := 1 to 2 do      begin     {2 branches of
solutions}
          for j3 := 1 to 2 do     begin  { R inc. or dec. }
                                 {P,Q and R back to starting
values}
             P := Px[ii,jj];  Q := Py[ii,jj];  R := RR;
             if   phase = 0  then     begin
                p1 := pp[1,i3];   q1 := qq[1,i3];
                p2 := pp[2,i3];   q2 := qq[2,i3];
             end
             else  begin
                p2 := pp[1,i3];   q2 := qq[1,i3];
                p1 := pp[2,i3];   q1 := qq[2,i3];
```

```
                end;      { if phase}      { end of backing to
starting value. }
writeln(o2,'stop3:       p1        q1        p2        q2
P        Q');
writeln(o2,'
',p1:10:4,q1:10:4,p2:10:4,q2:10:4,P:10:4,Q:10:4 );

                Trans_to_abs_1;   { for pole point }
                Calc_r_1;

                dR := 0.2;
                if j3 = 2 then    dR := -0.2;

                PL := P;   p1L := p1;   p2L := p2;
                QL := Q;   q1L := q1;   q2L := q2;

                       {the initial solution point--the pole
}
               writeln(o1,P:8:4,',',Q:8:4);
               writeln(o9,P:8:4);
               writeln(o9,Q:8:4);
               writeln(o2,'---------------------------------------
-------');
               writeln(o2,'At P',ii,jj,':  Branch ',i3,',
dR=',dR:10:4);
               writeln(o2,'---------------------------------------
-------');
               writeln(o2,'Starting a branch of solution:');
               writeln(o2,'---------------------------------------
-------');
               write(o2,'     P          Q         p1         q1
p2');
               writeln(o2,'        q2         R');

writeln(o2,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);

               Mp_out_pole;
               writeln(o2,'---------------------------------------
-------');
               writeln(o2,'Good center points:');

               times := 0;   points := 0;   tired := false;

               Repeat {until tired}
                  R := R + dR;
                  count := 0;

                  Repeat {until done or large}
                    large := false;
                    Load_Array;
                    done := true;
                    for i := 1 to 6 do
                      if (abs(G[i]) >eps) then done := false;
```

```
                if not done then    begin
                    Sim_Eq;
                    P  := P  + D[1];   Q  := Q  + D[2];
                    p1 := p1 + D[3];   q1 := q1 + D[4];
                    p2 := p2 + D[5];   q2 := q2 + D[6];
                end
                else    begin
                    done := true;

            writeln(o1,P:8:4,',',Q:8:4);
            writeln(o9,P:8:4);
            writeln(o9,Q:8:4);
            writeln(o2,'');

writeln(o2,'**********************************************')
;
            writeln(o2,'Good point ',points+1);
            write(o2,'     P          Q         p1
q1         p2');
            writeln(o2,'         q2          R');

writeln(o2,P:10:4,Q:10:4,p1:10:4,q1:10:4,p2:10:4,q2:10:4,R:1
0:4);

            Trans_to_abs_1;
            Calc_r_1;
            Mp_out;

            PL := P;   p1L := p1;   p2L := p2;
            QL := Q;   q1L := q1;   q2L := q2;
            points := points + 1;
          end;
          if abort then count := maxcount;
          if count >= maxcount then large := true;
          count := count +1;
        Until done or large;
          if (not done) and large then
          begin
            times := times + 1;
            if times = 3 then
               tired := true
            else  begin
               points := 0;  {use maxpoints at each dR}
               R := R - dR;   {back to last good point}
               dR := dR / 10; {try again with smaller
dR}
               writeln(o2,'');
               writeln(o2,'dR=dR/10= ',dR:8:4,'
starting at here:');

               P := PL;   p1 := p1L;   p2 := p2L;
               Q := QL;   q1 := q1L;   q2 := q2L;
            end;   {if times}
          end; {if large}
```

```
              if points >= maxpoints then tired := true;
           Until tired;
       end;    {for j3}
      end;     {for i3}
  End;
{ This program is created by modifying FP_PROCS.PAS [12] on
the basis of the equations developed for the synthesis
problem MP_3_3_1}
```

I. Program CL_2_2_2.PAS

```
Program cl_2_2_2;
   {program calculates a circle point curve for}
   {an adjustable crank length for a 4-bar linkage}
   {The 1st phase: positions 1, 2}
   {The 2nd phase: positions 3, 4}
   {The 3rd phase: positions 5, 6}

Const
  In_fil_name   = 'CL_2_2_2.DAT';
  Layer_name    = 'CL_2_2_2';
  Out_fil_name = 'CL_2_2_2.DXF';
  out2_name = 'circpt.dat';
Type
  vector3 = array[1..3] of real;
  vector4 = array[1..4] of real;
  vector5 = array[1..5] of real;
  vector6 = array[1..6] of real;

Var
  Out_fil, In_fil, out2 : text;
  x, y, th   : vector6;
  bp, cp, dp, ep, fp, gp, hp : real;
  alpha, beta, asymp : real;
  i, j, k, nroots : integer;
  vl, vu : vector3;

{$I POLYLINE.PAS}

{$I CPOINTS.PAS}

{$I LIMITS.PAS}

Procedure Get_data;
  {get data for 6 positions 1,2,3,4,5,and 6}
  Var
    i : integer;
  Begin
    Assign (In_fil,In_fil_name);
    Reset  (In_fil);
    for i := 1 to 6 do
      begin
        Readln (In_fil, x[i]);
        Readln (In_fil, y[i]);
        Readln (In_fil, th[i]);
      end;
    Close (In_fil);
  End;

Procedure Calc_con;
  {calculate constants for a circle point curve equation}
  {for 6 positions with 2 on each phase}
  Var
```

```
  i, j :integer;
  a, b : real;
  theta, costh, sinth : real;
  aa,  bb, cc, dd, ee, ff, gg, bp1 : real;
  aa2, bb2 : real;
  a1, a2, a3 : vector3;
  b1, b2, b3 : vector3;
  c1, c2, c3 : vector3;
  q1, q2, q3 : vector5;
  r1, r2, r3 : vector5;
  s1, s2, s3 : vector5;
Begin
  for i := 1 to 5 do            {for positions 2-6}
  begin
    j := i + 1;
    theta := (th[j] - th[1]) * pi /180.;
    costh := cos(theta);
    sinth := sin(theta);
    a := x[j] - x[1];
    b := y[j] - y[1];
    q1[i]:= 1-costh; q2[i]:= -sinth;
    r1[i]:= sinth;   r2[i]:= 1-costh;
    s1[i]:= -a;      s2[i]:= -b;
    q3[i]:= a*costh + b*sinth;
    r3[i]:= b*costh - a*sinth;
    s3[i]:= (a*a + b*b) / 2.;
  end;

    a1[1] := q1[1]; a2[1] := q2[1]; a3[1] := q3[1];
    b1[1] := r1[1]; b2[1] := r2[1]; b3[1] := r3[1];
    c1[1] := s1[1]; c2[1] := s2[1]; c3[1] := s3[1];

  for i := 2 to 3 do
  begin
  k := 2*(i-1);
  a1[i] := q1[k]-q1[k+1]; a2[i] := q2[k]-q2[k+1];
  b1[i] := r1[k]-r1[k+1]; b2[i] := r2[k]-r2[k+1];
  c1[i] := s1[k]-s1[k+1]; c2[i] := s2[k]-s2[k+1];
  a3[i] := q3[k]-q3[k+1];
  b3[i] := r3[k]-r3[k+1];
  c3[i] := s3[k]-s3[k+1];
  end;
  aa := det (a1, a2, a3);
  bb := det (b1, b2, b3);
  cc := det (a1, c2, b3) + det (b1, c2, a3);
  cc := det (c1, a2, b3) + det (c1, b2, a3) + cc;
  dd := det (a1, a2, c3) + det (a1, c2, a3)
      + det (c1, a2, a3);
  ee := det (b1, b2, c3) + det (b1, c2, b3)
      + det (c1, b2, b3);
  ff := det (a1, c2, c3) + det (c1, a2, c3)
      + det (c1, c2, a3);
  gg := det (b1, c2, c3) + det (c1, b2, c3)
      + det (c1, c2, b3);
```

```
    hp := det (c1, c2, c3);
    bp1 := aa*aa + bb*bb;
    bp := sqrt (bp1);
    cp := cc * (bb*bb - aa*aa) + 2 * aa * bb * dd;
    cp := (cp - 2 * aa * bb * ee) / bp1;
    dp := (ee * aa*aa + dd * bb*bb - aa * bb * cc) / bp1;
    ep := (dd * aa*aa + ee * bb*bb + aa * bb * cc) / bp1;
    fp := (bb * ff - aa * gg) / bp;
    gp := (aa * ff + bb * gg) / bp;
    alpha := bb / bp;
    beta  := aa / bp;
    asymp := - dp / bp;
  End;

BEGIN
  Get_data;
  Calc_con;
  Limits;
  Open_data;
  Cpoints;
  Close_data;
END.
{This program is created by modifying program CL_3_2.PAS[12]
on the basis of the equations developed for the synthesis
problem CL_2_2_2.}
```

References

1. Tao, D. C., and L. G. Amos. "A Four-Bar Linkage Adjustable for Variable Straight Line Motion." *Journal of Engineering for Industry, Transactions of the ASME*, (August 1965): 287–290.

2. Tao, D. C., and S. Krishnamoorthy. "Linkage Mechanism Adjustable for Variable Symmetrical Coupler Curves With a Double Point." *Mechanism and Machine Theory*. 13 (1978): 585–591.

3. Tao, D. C., and H. S. Yan. "Technology Transfer in the Design of Adjustable Linkages." *Journal of Mechanical Design, Transactions of the the ASME,* (July 1979): 495–498.

4. Beaudrot, C. B. "Synthesis of 4–Bar Linkages Adjustable for Several Approximate Straight–Line Motions of a Coupler Point." *Journal of Engineering forIndustry, Transactions of the ASME,* (Feb. 1969): 172–178.

5. Bonnell, R. D., and J. A. Cofer. "Kinematic Synthesis of Adjustable Four–Bar Linkages." *Journal of Applied Mechanics, Transactions of the ASME,* (Mar. 1966): 221.

6. Bonnell, R. D. "Graphical Synthesis of an Adjustable Four–Bar Linkage." *Design News,* (Mar. 18, 1964)

7. Hoekzema, R. J., N. H. Littlejohn, G. W. Pickard, and D. C. Tao. "Synthesis of a Four–Bar Linkage Adjustable for Variable Radius of Curvature of a Coupler Curve." *ASME Paper* No.70–Mech–30.

8. McGovern, Joseph F., and George N. Sandor. "Kinematic Synthesis of Adjustable Mechanisms Part 1: Function Generation." *Journal of Engineering for Industry, Transactions of the ASME,* (May, 1973): 417–422.

9. McGovern, Joseph F., and George N. Sandor. "Kinematic Synthesis of Adjustable Mechanisms Part 2: Path Generation." *Journal of Engineering for Industry, Transactions of the ASME,* (May, 1973): 423–429.

10. Ahmad, Anees and Kenneth J. Waldron. "Synthesis of Adjustable Planar 4–Bar Mechanisms." *Mechanism and Machine Theory.* 14 (1979): 405–411.

11. Ahmad, Anees. "Synthesis of Adjustable Planar Four–Bar Mechanisms." *Ph.D Dissertation, University of Houston.* (1979).

12. Wilhelm, Andrew J. "Kinematic Synthesis of Adjustable Linkages for Motion Generation." *Ph.D Dissertation, The Wichita State University.* (1989).

13. Waldron, K. J. "Graphical Solution of the Branch and Order Problems of Linkage Synthesis for Multiply Separated Positions." *Journal of Engineering for Industry, Transactions of the ASME,* (Aug. 1977): 591–597.

14. Waldron, K. J. "Elimination of the Branch Problem in Graphical Burmester Mechanisms Synthesis for Four Finitely Separated Positions."*Journal of Engineering for Industry, Transactions of the ASME,* (Feb. 1976): 176–182.

15. Waldron, K. J. "The Order Problem of Burmester Linkage Synthesis." *Journal of Engineering for Industry, Transactions of the ASME,* (1975): 1405–1406.

16. Filemon, E. "In Addition to the Burmester Theory." *Proceedings of Third World Congress for Theory of Machines and Mechanisms,* Kupari, Yugoslavia, Vol. D, (1971): 63–78.

17. Waldron, K. J. "Graphical Burmester Design of Four–Bar Linkages for a Specified Position Order and With All Positions Reachable Without Disconnection. "*Proceedings of the 4th OSU Applied Mechanisms Conference,* (1975).

18. Wilhelm, Andrew J., and Raj S. Sodhi. "Burmester Synthesis for Four–Bar Linkages on a Microcomputer CAD System. "*Proceedings of the 10th Applied Mechanisms Conference,* New Orleans, LA., (Dec. 1987).

19. Freudenstein, Ferdinand, and George N. Sandor. "On the Burmester Points of a plane. "*Journal of Applied Mechanics, Transactions of the ASME,* (Mar. 1961): 41–49.

20. Kramer S. N., and G. N. Sander. "Selective Precision Synthesis – A General Method of Optimization for Planar Mechanisms." *Journal of Engineering for Industry, Transactions of the ASME,* (May 1975): 689–701.

21. Sandor, George N., and Arthur G. Erdman. "Advanced Mechanism Design: Analysis and Synthesis." Vol. 2, Prentice–Hall Inc. (1984).