Int. J. Engng Ed. Vol. 8, No. 6, pp. 466–471, 1992
Printed in Great Britain.

0742-0269/91 $3.00+0.00

Computer-Aided Linkage Synthesis (CALS) for Planar Mechanisms*

L. PENNER
V. CHEUNG
R. HAN†

Mechanical and Industrial Engineering Department, University of Manitoba, Winnipeg, Manitoba, Canada, R3T 2N2

CALS is a program designed to perform synthesis calculations and animate the resulting linkage. A user-friendly interactive menu system provides a logical method of designing four-bar and slider linkages for path, motion and function generation and teaches the user the basics of linkage synthesis theory. CALS uses the standard form loop-closure equations in complex number notation and the compatibility method to solve for four precision points. The many features of this program make it a useful teaching tool in undergraduate courses in mechanism analysis and synthesis.

SUMMARY OF EDUCATIONAL ASPECTS OF THE PAPER

1. The paper discusses material for a course in: Applied Linkage Synthesis.
2. Students of the following departments are taught in the course: Mechanical and Industrial Engineering.
3. Level of the course: Fourth year (graduating class).
4. Mode of presentation: Lectures, model demonstration and computer simulation.
5. Is the material presented in a regular or in an elective course: Elective.
6. Class hours required to cover the material: Three hours per week.
7. Student homework and revision hours required for the materials: Five to six hours per week.
8. Description of the novel aspects presented in the paper: CALS is an interactive, user-friendly program for linkage synthesis and animation of a four-bar or a slider linkage with four precision points, making it ideal for demonstration and teaching purposes. The user does not need an in-depth knowledge of linkage synthesis to use the program as the program automates most of the difficult tasks.
9. The standard text recommended for the course, in addition to authors' notes: G. N. Sandor and A. G. Erdman, *Advanced Mechanism Design: Analysis and Synthesis*, vol. 2, Prentice Hall, 1984.
10. The material is/is not covered in the text. The discussion in the text is different in the following aspects: No equivalent computer program for linkage synthesis is presented in the text.

INTRODUCTION

COMPUTERS have played an increasingly important role as an educational tool in mechanical engineering, especially in the field of linkage analysis and synthesis. Linkage synthesis is the process of designing a mechanism to execute a desired motion or task. The folding of aircraft landing gear, the sliding motion of a piston connected to a rotating crankshaft, and the folding of a lawn chair are typical examples of employing linkages to achieve certain common tasks. Even though these linkages may have different types and numbers of links, they can be derived from a basic four-bar model. Since this model is the most basic planar mechanism, it can easily be adapted to computer linkage synthesis programs for educational purposes.

The calculations required to design a linkage are long and tedious and do not demonstrte how the linkage will operate under real life constraints. The purpose of the computer program, CALS, is to perform these calculations and animate the resulting linkage. Furthermore, the program has an interactive menu system that provides a logical method for solving design problems and teaches the user about linkage synthesis. This allows the user with

---

* Paper accepted 1 November 1992.

† Ray P. S. Han is an Associate Professor of Mechanical Engineering.

minimal knowledge of the synthesis theory to begin designing different types of linkages and observe their motion.

Some similar software packages such as 'LINCAGES', 'LINKS', 'ADAMS' and 'IMP' are available [1]. 'LINCAGES' and 'LINKS' provide both analysis and synthesis of a linkage for three or more precision points but without the animation. 'SIMUL' [2] is another program developed solely for the animation of a particular type of linkage: an in-line slider linkage. On the other hand, CALS can perform both the synthesis and animation of any four-bar or slider linkage with four precision points. Unlike CALS, these programs are aimed for relatively advanced users who must acquire substantial knowledge of the theory before designing linkages. CALS is specially intended for the educational purpose of putting theory to use in practical applications. A program with a similar purpose is also developed by Bartlett, Smid, and Strang [3]. Their program only deals with four-bar linkages and three precision points while CALS can synthesize and animate both slider and four-bar linkages with four precision points.

The three types of synthesis—path, motion and function generation—require different input parameters which will be explained later. CALS directs the user to respond to the appropriate parameters. As the user synthesizes different types of linkages, he learns the required parameter for each synthesis type. There are a number of features of the program that help to guide the user through the design process. Features like context-sensitive help and the interactive menu system allow the user to return at any point and change the input parameters, making CALS very user friendly to operate. These and other major features will be discussed further in the program features section.

THEORY FOR FOUR PRECISION POINT LINKAGE SYNTHESIS

A typical linkage consists of three links as shown in Fig. 1. The three links are the input link, the coupler, and the output link. The coupler is shown as a ternary link which consists of a rigid triangular frame. In other cases, the coupler can be a binary link similar to the input or the output link. The angular rotations of the input, coupler, and output link, measured from the initial position to a jth position, are represented by the following symbols: β_j, α_j, γ_j respectively.

In general, there are three types of linkage synthesis: path, motion and function generation. For each type of generation, different parameters of linkage motion are prescribed. In path generation, the path traced by a fixed point on the coupler and the input link rotation are prescribed. Similarly, motion generation also prescribes the coupler path, but the coupler rotation is of interest rather than the input link rotation. For function generation, the rotations of the input link and the output link are related by a mathematical function and thus attention is drawn to these two rotations.

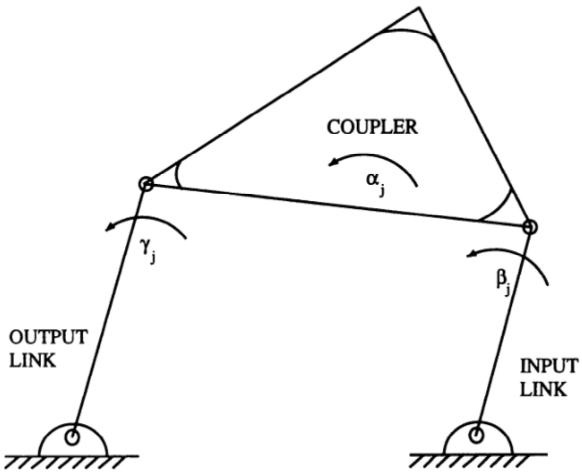


Fig. 1. A four-bar linkage.

There are several graphical and analytical methods of synthesis. This particular program employs the complex number method which represents each link's magnitude and direction by a vector. Figure 2 shows a vector pair called a dyad. This dyad consists of an inut (or output) link and a coupler which are represented by the vectors **W** and **Z** respectively. That means a four-bar linkage can be perceived as two dyads while a slider-crank mechanism is made up of one dyad only.

As a dyad moves from an initial position to a jth position, a closed loop consisting of five vectors is formed as shown in Fig. 3.

The following equation describes this closed loop.

$$\mathbf{W}(e^{i\beta_j}-1)+\mathbf{Z}(e^{i\alpha_j}-1)=\delta_j \qquad (1)$$

where

$\mathbf{W}$ = input link or output link vector
$\mathbf{Z}$ = coupler link vector
δ_j = displacement vector
β = input link rotation
α_j = coupler rotation
j = 2, 3 or 4 for up to four precision points

This equation is known as the standard-form loop-closure equation which is simply the vector sum around the loop formed by one dyad at the initial and the jth position. All the angular rotations are

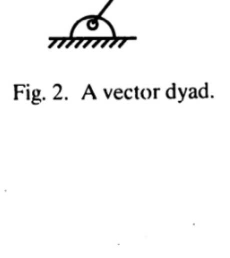


Fig. 2. A vector dyad.

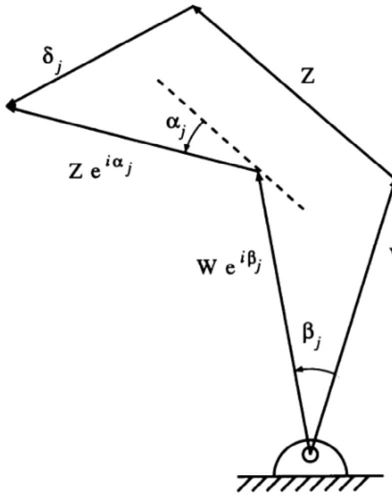


Fig. 3. A vector loop diagram.

measured counterclockwise with respect to the initial position.

For a four precision point synthesis, there are a total of four prescribed positions including the initial one. As a result, only three vector equations which describe the motion of the linkage from the initial to the second, third and fourth positions are generated. These are:

$$\mathbf{W}(e^{i\beta_2}-1)+\mathbf{Z}(e^{i\alpha_2}-1)=\delta_2 \qquad (2)$$

$$\mathbf{W}(e^{i\beta_3}-1)+\mathbf{Z}(e^{i\alpha_3}-1)=\delta_3 \qquad (3)$$

$$\mathbf{W}(e^{i\beta_4}-1)+\mathbf{Z}(e^{i\alpha_4}-1)=\delta_4 \qquad (4)$$

All the symbols bear the same meaning as the standard-form equation.

As mentioned above, different types of synthesis have different prescribed parameters. Depending on the synthesis type, δ_j and either β_j or a_j are known. For instance, in motion-generation, the path traced by the coupler, δ_j and the coupler rotation, a_j are presribed. Since there are three vector equations (six scalar equations), and seven unknowns (both the magnitude and direction of vectors, **W** and **Z**, and the angles β_2, β_3, and β_4), one of the unknowns must be chosen arbitrarily in order to solve for the other six unknowns. In practice, the first input link rotation β_2 is usually picked as an arbitrary choice. Similarly, according to the definitions of path and function generation, δ_j and β_j are prescribed, and α_2 becomes the arbitrary choice.

After choosing one of the unknowns arbitrarily, there are only six unknowns with six scalar equations. Thus, the problem is solvable. Unfortunately, these six scalar equations are non-linear because of the exponential function involved. For example, in motion-generation, the unknown angular rotations, β_3 and β_4 must be solved first to make the equations linear. This is done using the following computability equation.

$$\Delta_2 e^{\beta_2}+\Delta_3 e^{\beta_3}+\Delta_4 e^{\beta_4}+\Delta_1=0 \qquad (5)$$

where

$$\Delta_1=-\Delta_2-\Delta_3-\Delta_4 \qquad (6)$$

$$\Delta_2=(e^{\alpha_3}-1)\delta_4-(e^{\alpha_4}-1)\delta_3 \qquad (7)$$

$$\Delta_3=(e^{\alpha_4}-1)\delta_2-(e^{\alpha_2}-1)\delta_4 \qquad (8)$$

$$\Delta_4=(e^{\alpha_2}-1)\delta_3-(e^{\alpha_3}-1)\delta_2 \qquad (9)$$

The Δ's are known because they contain only the known input data. This compatibility equation can be solved by an analytical solution based on geometric construction. This analytical solution is shown as follows [4]:

$$\Delta=\Delta_1+\Delta_2 e^{\beta_2} \qquad (10)$$

$$\cos\theta_3=\frac{\Delta_4^2-\Delta_3^2-\Delta^2}{2\Delta_3\Delta} \qquad (11)$$

$$\sin\theta_3=|\surd(1-\cos^2\theta_3)|\geqslant 0 \qquad (12)$$

$$\beta_3=arg\Delta+\theta_3-arg\Delta_3 \qquad (13)$$

$$\cos\theta_4=\frac{\Delta_3^2-\Delta_4^2-\Delta^2}{2\Delta_4\Delta} \qquad (14)$$

$$\sin\theta_4=|\surd(1-\cos^2\theta_4)|\geqslant 0 \qquad (15)$$

$$\beta_4=arg\Delta-\theta_4-arg\Delta_4 \qquad (16)$$

After β_3 and β_4 are determined using the above equations, all the unknown angles are defined. Now, both the direction and magnitude of the link vectors, **W** and **Z** can be solved using any two of the standard eqs. (2), (3) and (4).

The above procedure only synthesizes for one dyad. Therefore, to complete the synthesis for the second dyad of a four-bar linkage, this calculation must be repeated. Note that the unknown angular rotations of the first dyad become the known angles for the second dyad and a new arbitrary angular choice must be selected. Since a total of two arbitrary choices are picked to produce one four-bar linkage, there will be two 'infinites' of solution associated with each mechanism synthesis.

PROGRAM IMPLEMENTATION AND FEATURES

CALS runs on an IBM PC or compatible with at least 512 Kbytes of memory. A hard disk is not necessary as the program occupies only 300 Kbytes on a diskette. High quality graphics are displayed on the screen using a VGA card. The program is written using the Borland TURBO C++® programming language. This language was chosen because of its popularity and easy manipulation of graphics.

The user does not require special training to use the program because it is menu-driven and contains many context-sensitive help messages. The emphasis of the program is on teaching the user, who may have little familiarity with the above theory, to design a linkage subject to a given set of constraints. The animation can also assist the user to learn how small changes in input parameters affect the movement of the linkage.

The program starts with the cursor highlighting the 'Introdution' on the menu bar. Figure 4 shows a picture of the screen with the main menu, the introductory pull-down menu and a help message.

In addition to the help given in the 'Introduction' section, context-sensitive help is available for every aspect of the program by pressing the ⟨F1⟩ key. The rest of the main menu is divided into 'Setup Modes', 'Problem Definition', 'Synthesis', 'Output' and 'Program'. The selection is made using the cursor keys or by pressing the highlighted letters, first in the main menu bar then in the pull-down menu. The following is a list of each main menu heading and the options in each pull-down menu. Note that the bold face letters correspond to the highlighted letters that activate each option.

Introduction
  main menu
  useable **k**eys
  general procedure

Setup **M**odes
  angular mode
  decimal places

Problem **D**efinition
  work area
  origin
  o**b**stacles
  linkage type

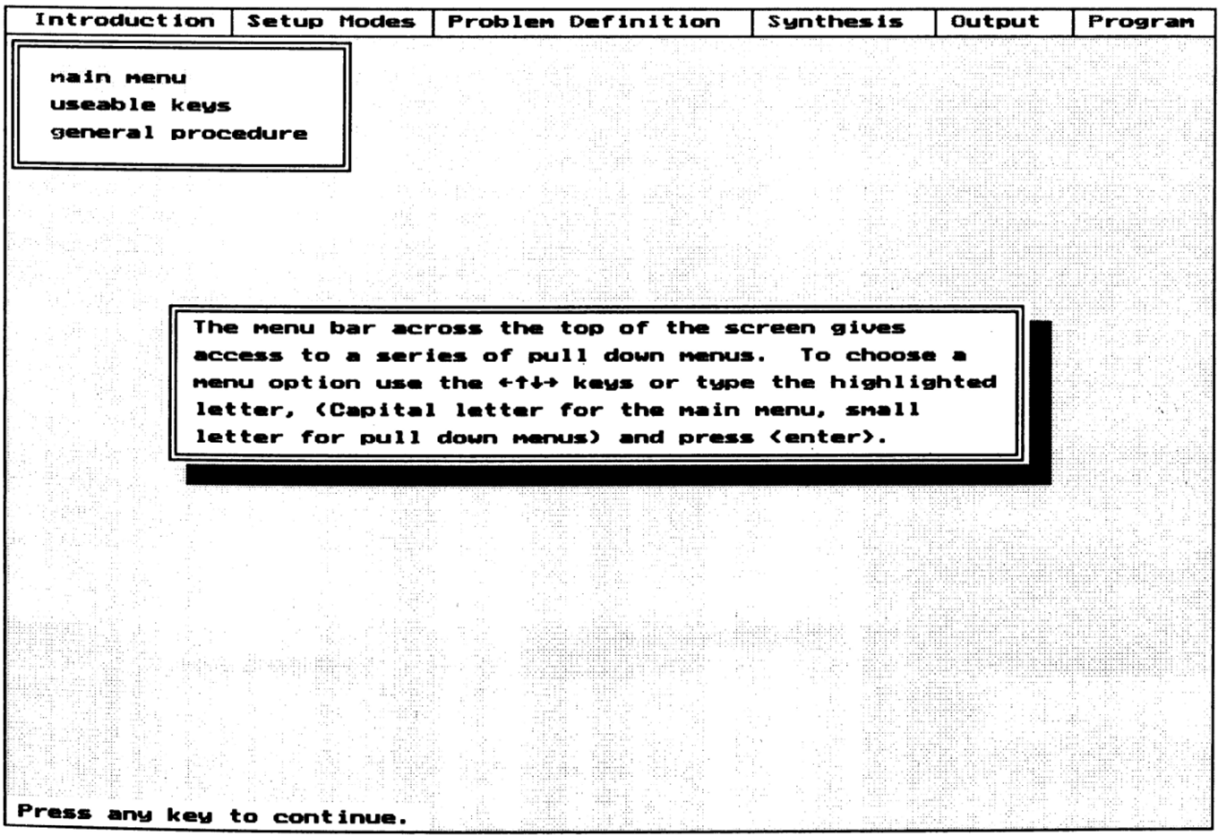
Synthesis
  precision points
  angular **r**otations
  arbitrary choice
  execute synthesis

Output
  refresh screen
  tabular form
  print tables

Program
  save
  load/restart
  new linkage
  quit

The menu system is arranged in such a way so as to encourage a logical approach to problem-solving. First, the user may set up the type of angular measurement and accuracy desired, and then input the data which defines the problem. The 'Problem Definition' includes defining the space constraints (i.e. 'Work Area') and a frame of reference (i.e. 'Origin') within that space. Any obstacles or points will be defined relative to the user-defined origin. The user must also define the type of linkage, slider or four-bar, and the type of synthesis to be performed, path, motion or function generation.

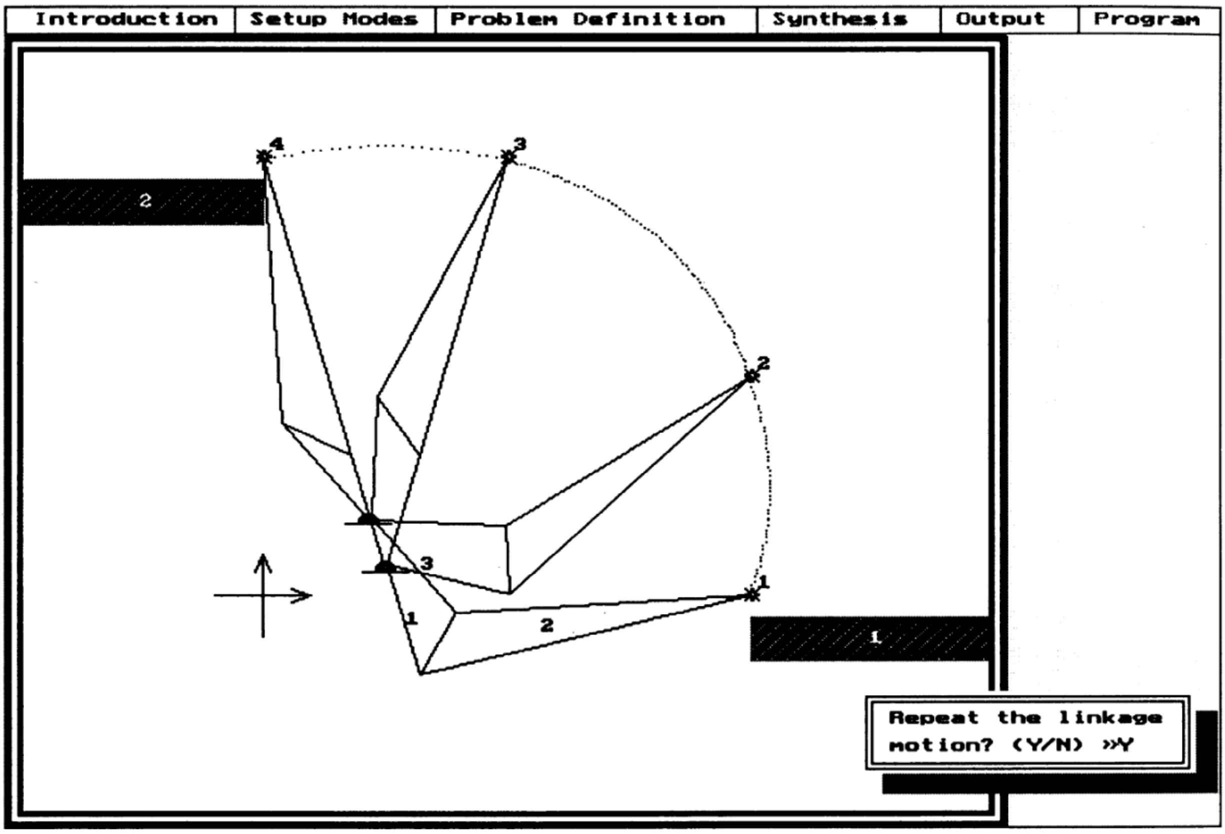
The next step is prescribing the synthesis parameters which are the precision points, angular rotations and the arbitrary choices. These three sets of variables control the size, shape and motion of the resulting mechanism. For example, increasing the amount of angular rotation of a particular link has the effect of shortening its length. After executing synthesis and seeing the linkage in motion, the user may find that linkage needs some changes to satisfy all constraints. An optimum linkage with the desirable link lengths and orientations for a particular task can thus be designed by adjusting the appropriate synthesis parameters. The user may not be able to see the effect of a very small adjustment by looking at the graphical representation. In this case the tabular output



The menu bar across the top of the screen gives access to a series of pull down menus. To choose a menu option use the ←↑↓→ keys or type the highlighted letter, (Capital letter for the main menu, small letter for pull down menus) and press <enter>.

Press any key to continue.

Fig. 4. Main menu with pull-down.



Repeat the linkage motion? (Y/N) »Y

Fig. 5. Graphical representation.

provides a helpful quantitative means of describing the linkage as opposed to qualitative output from the animation.

Figure 5 shows the resulting linkage at the four prescribed positions after animation has completed. This particular linkage would lift an object from a lower conveyer belt to an upper belt while rotating the object exactly 90 degrees and moving along a specific path.

The data storage scheme in the program is based on the ASII format and both intermediate and final linkage data can be saved for future use. In particular, the storage of intermediate data enables the user to restart a problem from where he left, at any time. In the data file, each variable is clearly labelled to provide the user with a readable way of interpreting and modifying the data outside the program itself. The data file can also be used as a means of passing out problems to students and receiving solutions for grading. This feature enables the program to be easily incorporated in mechanism analysis and synthesis courses, as a very useful and efficient teaching tool.

CONCLUSION

The computer program, CALS, was developed to perform the long and tedious calculations required for linkage synthesis and to animate the resulting linkage. Several features of CALS enable it to become a valuable teaching tool. Even users with minimal theory background can start designing linkages with the help of the interactive menu system. The menu-driven system guides the user in entering the required parameters and modifying the resulting linkage to meet the design criteria. Through this process, the user learns the appropriate parameters associated with different synthesis types and how the variables interact. Also, the animation assists the user in observing how a slight change in the parameters can significantly affect the resulting linkage. The 'load' and 'save' features allow the user to modify the data outside the program itself, and provide a means of distributing and collecting assignments in the context of a kinematic or synthesis course. All these features enable CALS to become a valuable teaching tool for undergraduate mechanism analysis and synthesis in mechanical engineering.

REFERENCES

1. G. N. Sandor and A. G. Erdman, *Advanced Mechanism Design: Analysis and Synthesis*, vol. 2, 251–265, 631–634. Prentice-Hall Inc., Englewood Cliffs (1984).
2. W. P. Boyle, Mechanism Kinematics: Simul™ Animation, *Int. J. Appl. Engng Ed.*, **6**, 361–363 (1990).

3. C. J. Bartlett, G. J. Smith, and D. R. Strang, Synthesis of a Four Bar Mechanism for Planar Rigid Body Guidance Using a Microcomputer, *Proceedings of the 9th OSU Applied Mechanisms Conference*, Oklahoma State University, Oct. 28–30 (1985).
4. Ray, P. S. Han, Lecture notes of course 25.467, Applied Linkage Synthesis, Department of Mechanical and Industrial Engineering, University of Manitoba (1991).

Lloyd Penner graduated with a B.Sc. (M.E.) from the University of Manitoba in May 1992 and is now enrolled for graduate studies at the same university. The focus of his M.Sc. research is in the area of computational thermodynamics.

Vivian Cheung graduated with a B.Sc. (M.E.) from the University of Manitoba in May 1992. Currently, she is working in Hong Kong as an engineer.

Dr Ray Han is an associate professor of mechanical engineering at the University of Manitoba. His research interests are in the area of theoretical and computational applied mechanics, mechanism analysis and synthesis, and finite element methods.