KINEMATIC PROGRAMS BASED ON MATLAB'S GUI
To Supplement the Textbook

Kinematics, Dynamics, and Design of Machinery, 2^{nd} Ed.

By

K. J. Waldron and G. L. Kinzel

Department of Mechanical Engineering

THE OHIO STATE UNIVERSITY

Table of Contents

Section **Page**

Section **Page**

MATLAB PROGRAMS BASED ON GUI

to Supplement the Textbook

Kinematics, Dynamics, and Design of Machinery, 2nd Ed.

1.0 Introduction

In the first edition of the textbook entitled *Kinematics, Dynamics, and Design of Machinery* by K.J. Waldron and G. L. Kinzel, a set of MATLAB programs were written to supplement the textbook. These programs were written so that the input was command driven. This means that the user must input information in response to prompts. These original programs are included in a separate folder on this CD. The programs are written using a fairly simple programming structure, and either the students or the instructor can modify them easily. The original programs will work with version 5.0 or higher or MATLAB.

While the original programs generally work well, they are more difficult to use then mouse driven programs. Therefore, most of the programs were rewritten to incorporate a graphical user interface (GUI) that is mouse driven. The new programs are much easier to use than the original ones; however, the programming structure is much more complex than the original programs, and considerable MATLAB programming expertise is required to make modifications in them. Therefore, in this user's manual, we have not attempted to define the internal structure of the programs. We will only explain how to use them. However, the source code for the programs is provided on the disk for those who are experienced in programming using the MATLAB GUI.

This manual gives a description of the MATLAB programs written to support the textbook. The program descriptions are presented in the order in which they appear in the program main menu. **The programs require version 6.0 or higher of MATLAB**. Either the full version or student version of MATLAB may be used. Some of the routines available in MATLAB tend to change with the version number, and the previous versions of MATLAB do not have some of the routines that are employed by the new version of the kinematic programs.

In general, the descriptions consist of a brief overview of the purpose of the program followed by a description of the input and output windows. The programs are menu driven so the inputs can be changed interactively.

1.1 Types of Programs Available

A brief description of each type of program (in alphabetical order) is given in the following:

Cam Design	Program for cam design with axial cylindrical-faced follower, axial flat-faced follower, oscillating cylindrical-faced follower, and oscillating flat-faced follower
Centrode Plot	Program for computing the centrode for a four-bar linkage
Cognate Drawing	Programs for computing the cognate linkages for a four-bar linkage
Coupler Curve Gen.	Programs for computing the coupler curves of a four-bar linkages and slider cranks
Crank Rocker Design	Program for crank rocker design

Four Bar Analysis	Program for the analysis of a four-bar linkage with either a crank or the coupler as driver
Gear Drawing	Programs for drawing gears given the geometry of the cutter
Inflection Circle Prog.	Program for computing the inflection circle of a four bar linkage
Rigid Body Guidance	Program for design of linkage for three positions for rigid body guidance
SC Shaking Force	Program for computing the shaking force for a slider-crank mechanism
Six Bar Mechanism	Program for analyzing a six-bar linkage
Slider Crank Analysis	Program for the analysis of a slider-crank linkage with the slider, coupler, or crank as driver

1.2 Program Installation

To install the programs, simply copy the folder entitled "*GUI Based Kinematic Programs*" from the CD to a folder on your hard disk. (The programs can be run directly from the CD, but they will be slower than if they were copied to the hard disk). After the programs are copied to the hard disk, open MATLAB and set the MATLAB path to the folder where the programs reside.

1.3 Running the Programs

To run the programs open MATLAB, and at the command prompt type *mainmenu*. The screen shown in Fig. 1.1 will appear. Click on the *Continue* button, and the screen in Fig. 1.2 will appear. As indicated in Fig. 1.2, the programs are arranged under four general headings: Linkage Design, Cam Design, Rigid Body Guidance Design, and Gear Design.

Fig. 1.1 Main screen after typing *mainmenu*

Fig. 1.2 Menu of program types

The plus (+) sign before each of the topics indicates that there are subtopics. Note that the *Run* button on the bottom right-hand side of the window cannot be actuated until one of the subtopics is selected. To select a program, first click on the topic you want and then click on the desired program. To run the program, click on the *Run* button. Note that you cannot run the programs by simply double clicking on them. When you are done with the programs, click on the *Close* button to terminate the program.

1.3.1 Programs Under *LinkageDesign*

Under *Linkage Design*, there are 10 subtopics as shown in Fig. 1.3. These subtopics corresponding to individual programs are:

1) Crank Rocker Design (CRDesign)
2) Cognate determinations of a four bar linkage (CognateAnalysis)
3) Design of double rocker linkage (DoubleRockerDesign)
4) Simple four bar linkage analysis (FourbarAnalysis)
5) Simple slider crank linkage analysis (SliderCrankAnalysis)
6) Six bar analysis program (SixbarAnalysis)
7) Simulation of Hrones & Nelson coupler curve atlas for four-bar linkages (HR CrankRockerAnalysis)
8) Simulation of Hrones & Nelson coupler curve atlas for slider-crank linkages (HRSliderCrankAnalysis)
9) Display of four-bar linkage centrode curves (CentrodeDesign)
10) Display of four-bar linkage inflection circle and calculation of center of curvature (Inflection4barAnalysis)
11) Analysis of shaking forces in slider-crank mechanism (ShakeAnalysis).

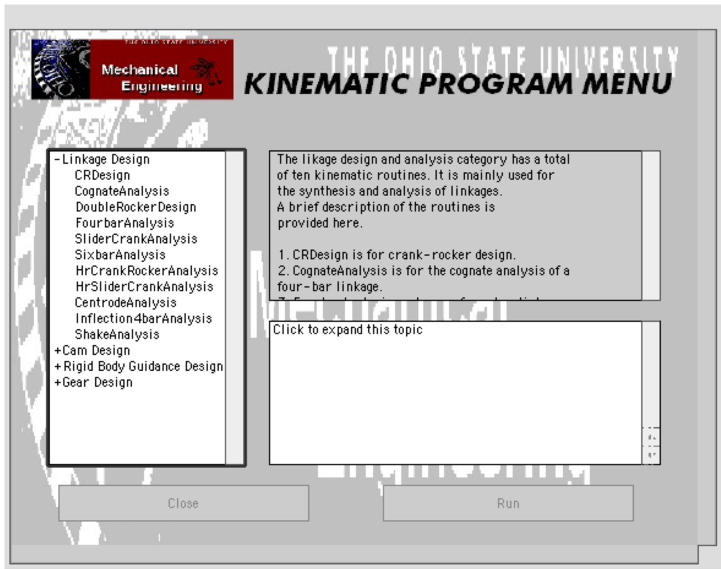


Fig. 1.3 Subtopics under *Linkage Design*

1.3.2 Programs Under *Cam Design*

Under *Cam Design*, there is one program, Cam2, as shown in Fig. 1.4. This program designs the follower displacement schedule and generates the cam profile for one of four types of followers: translating roller follower, translating flat-faced follower, oscillating roller follower, and oscillating flat-faced follower.

1.3.3 Programs Under *Rigid Body Guidance Design*

Under *Rigid Body Guidance Design*, there are three subtopics as shown in Fig. 1.5. These subtopics correspond to individual programs written for designing linkages for three-position rigid body guidance. The three programs are:

1) Rigid-body guidance or motion generation using a four-bar linkage (RBG4barDesign)

2) Rigid-body guidance or motion generation using a crank-slider mechanism (RBGCrankSliderDesign)

3) Rigid-body guidance or motion generation using a slider-crank mechanism (RBGSliderCrankDesign)

4) Rigid-body guidance or motion generation using a double slider or elliptic trammel mechanism (RBGElTrammelDesign)

1.3.4 Programs Under *Gear Design*

Under *Gear Design*, there are two programs as shown in Fig. 1.6. The first program (Arb2ThDesign) will compute and draw the tooth profile conjugate to an arbitrarily specified tooth form, and the second routine (GeardrDesign) will draw an involute profile given the parameters of the hob used to generate the gear form.

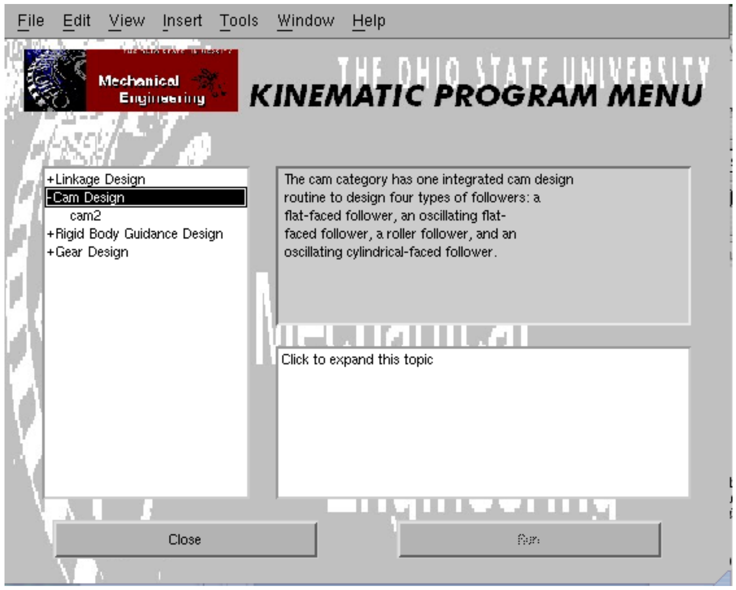


Fig. 1.4 Subtopic under *Cam Design*

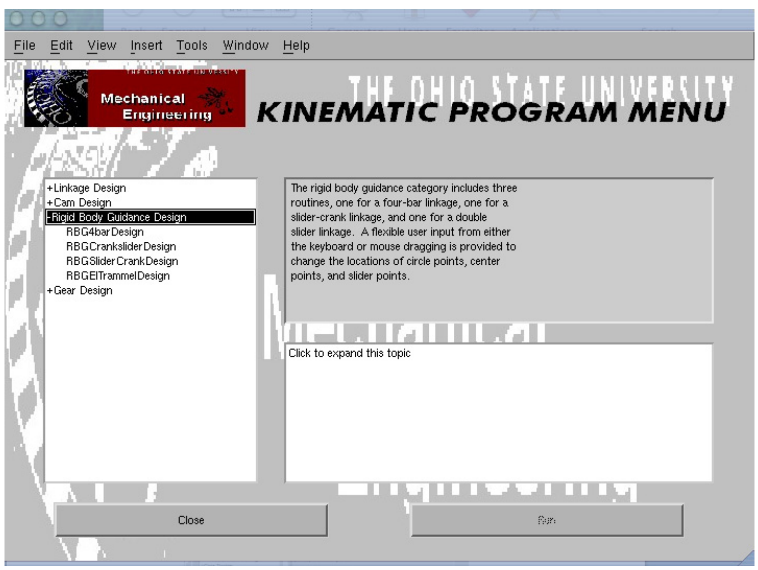


Fig. 1.5 Subtopics under *Rigid Body Guidance Design*

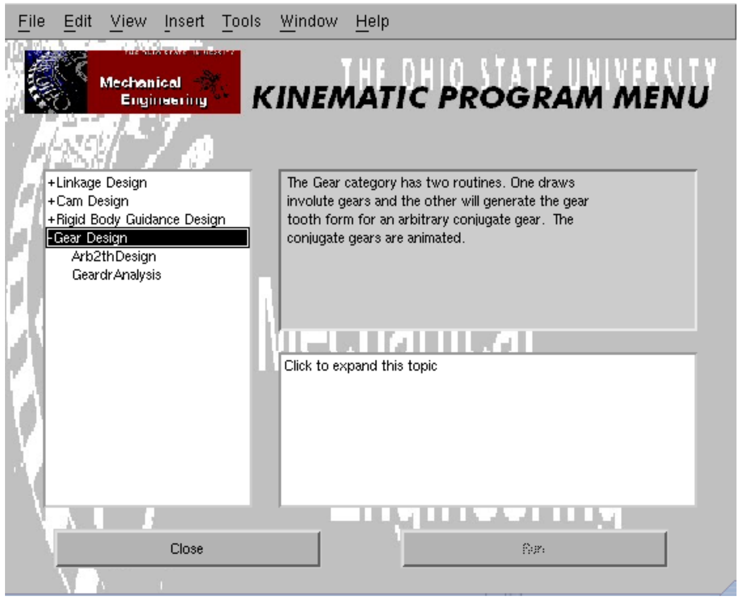


Fig. 1.6 Subtopics under *Gear Design*

1.4 MATLAB Graphics Window

As the programs are run, three windows will be of interest. The first is the MATLAB command window. This is the window that has the MATLAB command prompt and is where any errors are identified by the program. The second window is the graphics window identifying the program options. This is the window shown in Figs. 1.1-1.6. When you launch a program by selecting the program and clicking on *Run*, a second graphic will appear. This is the window where the input data are changed and where some of the design/analysis results are presentation. If the program includes an animation feature, the animation feature will appear in a third window. Note that if several windows are open while running the programs, some of the windows may become hidden under other windows.

1.5 Help in Using MATLAB

A brief overview of the use of the kinematic programs is given in the following sections. When describing how to use the programs, it is assumed that the user is familiar with the basics of MATLAB. For details, consult the MATLAB Users' Manual supplied by Mathworks. Alternatively, MATLAB has an excellent help facility. To obtain help on any topic in the library, simply type help and MATLAB will present a series of topics on which help may be obtained. By typing help and then the name of the topic, a description of that topic is displayed. Also, a list of subtopics on which help can be obtained is displayed. If the name of the subtopic is known, it is possible to type help followed by the subtopic name anytime that the MATLAB command prompt (>>) appears in the MATLAB window.

2.0 Programs under Linkage Design

2.1 Introduction

The descriptions given in the following will be limited to explaining how to run the programs available. It is assumed that the user will not be routinely modifying the code, and therefore, except in a few cases little theoretical information on the programs will be given. However, the programs based on the MATLAB GUI use essentially the same analytical routines used in the original set of MATLAB programs written for the first edition of the textbook *Kinematics, Dynamics, and Design of Machinery* by K. J. Waldron and G. L. Kinzel. The original set of routines along with a detailed user's manual are included elsewhere on this CD. Theoretical background information on some of the routines is included in that user's manual.

2.2 Crank-Rocker Design Program (CRDesign)

The objective in the design of a crank-rocker mechanism is to determine the lengths of the crank, rocker, and coupler for a given rocker angle and time ratio. The routine has two major windows, a design window and an analysis window. The design window accepts the user input data, including the rocker angle (theta), time ratio, and one link length. The second window is the analysis window that displays the final mechanism and shows the animated motion.

2.2.1 The Design Window for Crank-Rocker Program

The design window for the crank-rocker program is shown in Fig. 2.1. In the window, associated parameters are grouped together by a frame to visually indicate their relationship. Also, either-or options are given by providing a radio button set, which are also grouped together by a frame.

The design window uses the definitions given in Section 6.5 of the textbook. The program also gives the definitions of some of he variables if the *Definition* button is selected (clicked on). The definition window shown in Fig. 2.2 then appears. A brief description of the program is given if the *Info* button is pressed. The information file is displayed in Fig. 2.3.

The actual analysis uses the procedure explained in Section 6.5.4 of the textbook. The output oscillation angle (theta) must be input along with either alpha or Q. The radio button identifies the specific input variable. The user enters the value by moving the cursor over the value given and retyping a new value. To actually enter the value, the return key must be pressed on the computer keyboard. The locus for one extreme location for the output pivot (B_2) is the blue arc. To select new designs, either input the angle beta directly or click on and drag the green arrow to change the beta values. Here, beta is the counterclockwise angle between the positive X axis and the green vector (see Fig. 2.2). The design is automatically updated as beta is changed. In addition, the transmission angle range is shown and updated dynamically in the status bar at the bottom of the design window.

The program also has an optimization feature. If the *Optimization* button is selected, the program will determine the value for beta that optimizes the transmission angle. For the input values shown in Fig. 2.1, the optimized output values are shown in Fig. 2.4.

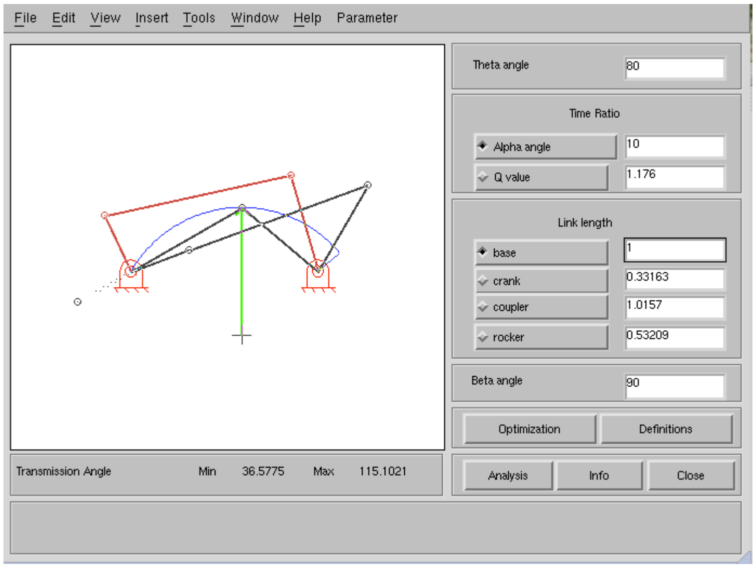


Fig. 2.1: The design window for the crank rocker design routine

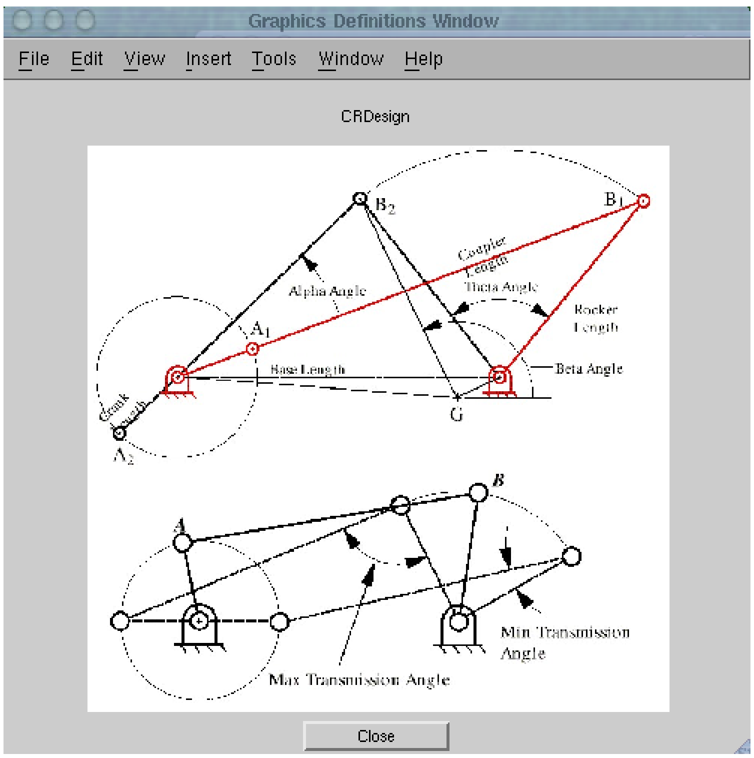


Fig. 2.2: The definitions window for the crank-rocker design routine

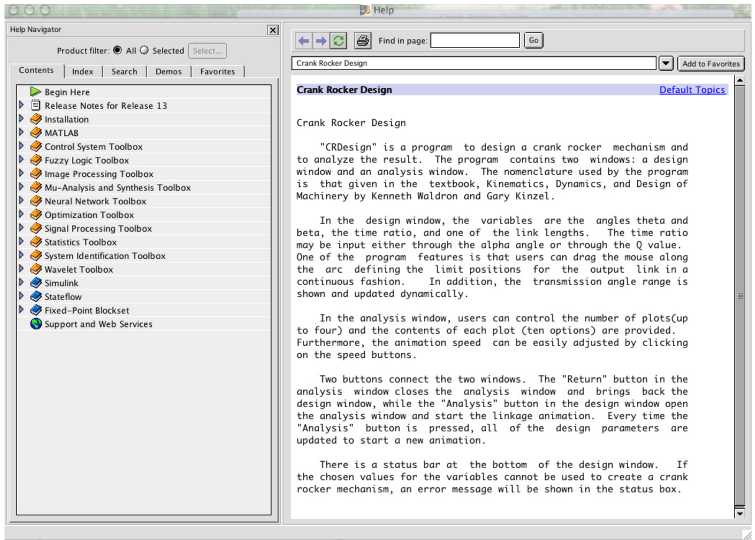


Fig. 2.3: The help window of the crank-rocker design routine

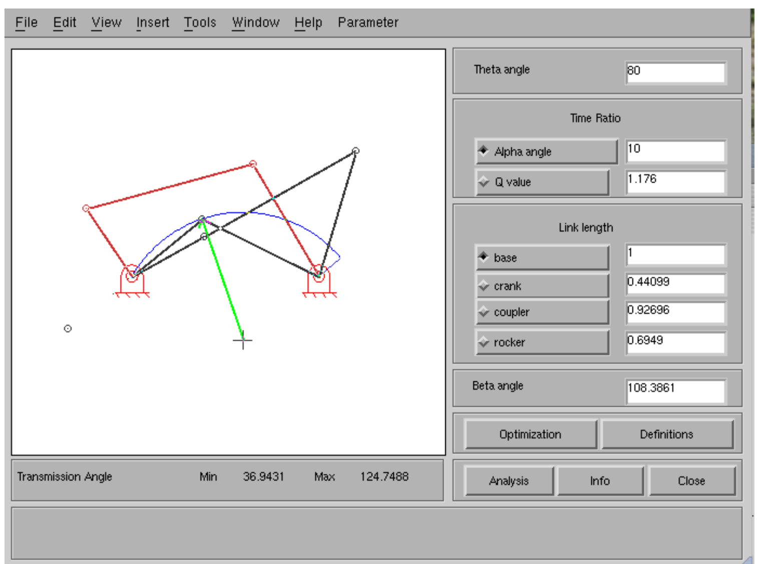


Fig. 2.4: Optimized linkage for input values in Fig. 2.1

2.2.2 The Analysis Window for Crank-Rocker Program

After the design is finalized, i.e., beta is selected, the *Analysis* button can be selected. In the analysis window shown in Fig. 2.5, users have control of the number of plots (up to four) shown and the contents of each plot (nine options). The nine options that are plotted as a function of the crank angle are:

1) Rocker (angular) position

2) Rocker (angular) velocity (for a constant crank angular velocity of 1 rad/sec)

3) Rocker (angular) acceleration (for a constant crank angular velocity of 1 rad/sec)

4) Copular (angular) position

5) Coupler (angular) velocity for a constant angular velocity of 1 rad/sec for the crank

6) Coupler (angular) acceleration for a constant angular velocity of 1 rad/sec for the crank

7) Input torque/output torque – This gives the mechanical advantage for the linkage

8) Transmission angle (degrees)

9) Mechanism plot

The nine options are shown in Figs. 2.5-2.8. The animation is continuous until the *Stop* button is selected. To change options, select the *Stop* button and make a change by changing either the number of plots or the items to be plotted. Then press *Start*. To change the item that is plotted, press on the title button, and select from the list presented. The animation can be speeded up or slowed down by pressing the plus (+) and minus (-) buttons, respectively. To return to the design window, select the *Return* button. Users can easily switch between the design and analysis windows at any time.

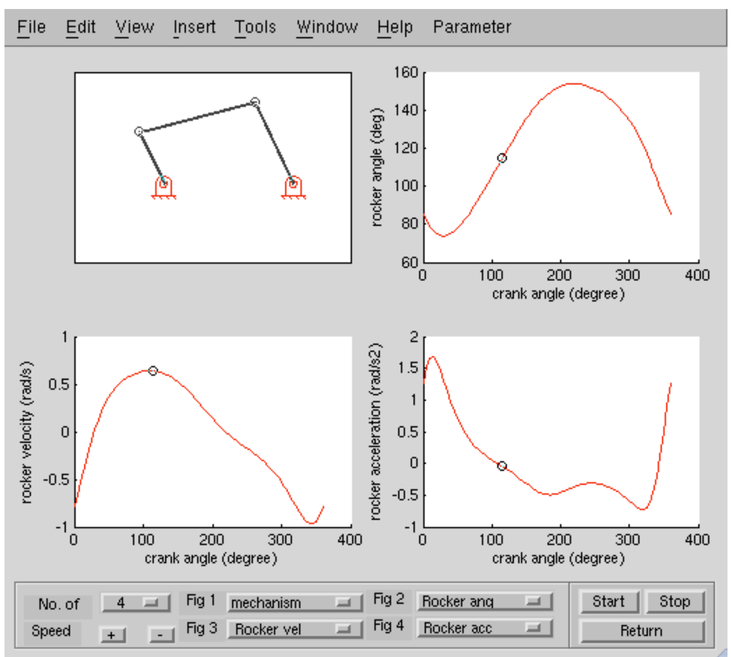


Fig. 2.5: Various output options for crank-rocker analysis (4 plots)

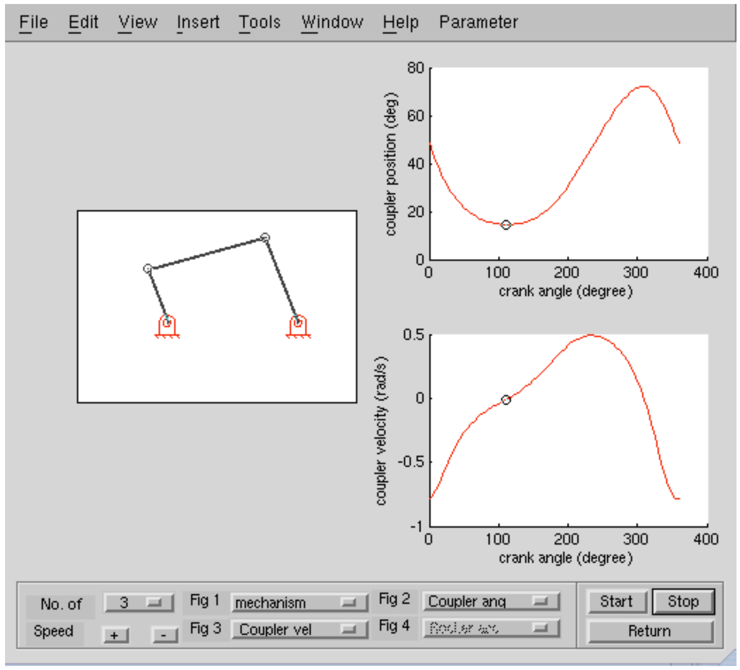


Fig. 2.6: Various output options for crank-rocker analysis (3 plots)

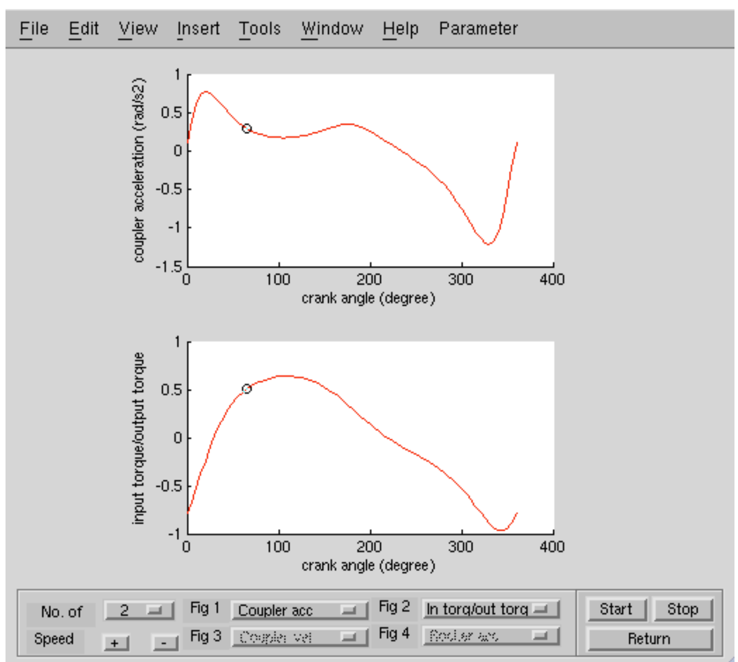


Fig. 2.7: Various output options for crank-rocker analysis (2 plots)

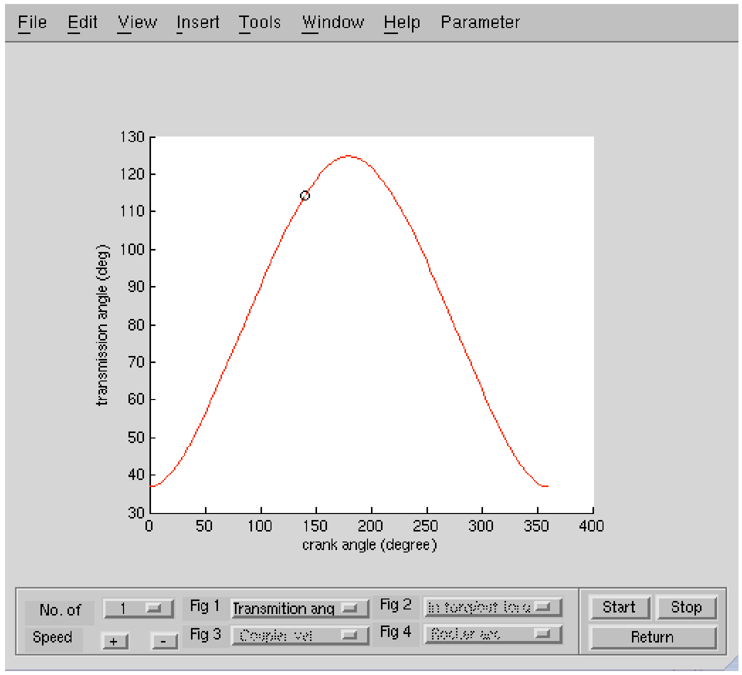


Fig. 2.8: Various output options for crank-rocker analysis (1 plot)

2.3 Program for Generating Cognate Linkages (CognateAnalysis)

This routine takes the basic four-bar linkage geometry and the location of the coupler point as input. It then determines Robert's linkage as well as the three individual cognate linkages. The equations are developed from Section 6.6.3 of the textbook.

The program is also structured in two windows, a design window (Fig. 2.9) and an analysis window (Fig. 2.10). In the analysis window, up to four plots can be displayed. Typically, these show each cognate separately along with Robert's linkage.

2.3.1 The Design Window for Cognate Program

The GUI displays the four-bar linkage and coupler curve on the left hand side of the design window. The editable link lengths (frame, crank, coupler, rocker, coupler point radius) are grouped together on the right hand side of the window. The angle between the coupler point radius and coupler and the frame angle are shown at the bottom of the window. The non-editable link lengths are grouped in another frame on the right-hand side of the design window. The radio button applies to the assembly mode desired. The first assembly mode is shown in Fig. 2.9 and the second in Fig. 2.10.

The user also can change the coupler point and curve by dragging the coupler curve around the screen. Two buttons *Zoom in* and *Zoom out* scale the plots because the parts of the mechanism and/or coupler curve might go outside of the plot window when the user drags the coupler point. If the *Info* button is chosen, the Robert's linkage information shown in Fig. 2.11 is displayed.

2.3.2 The Analysis Window for Cognate Program

The analysis window of the cognate design routine has the same layout as that of the crank-rocker routine. The only difference is the plot contents. For this routine, the analysis window has all of the mechanism plots. More than one instance of a graphic object can be generated if one cognate linkage is chosen for more than one plot. Several different plot options are shown in Figs. 2.12 – 2.14.

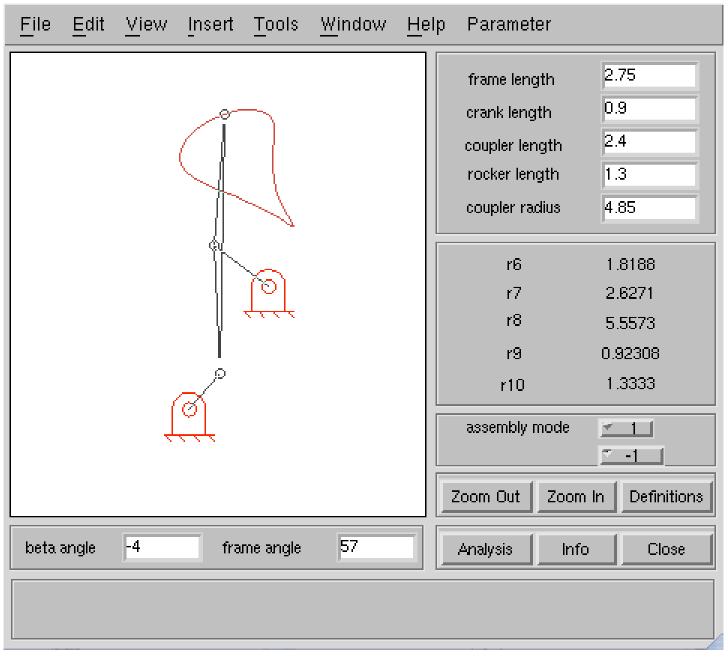


Fig. 2.9: The GUI design window for the cognates routine

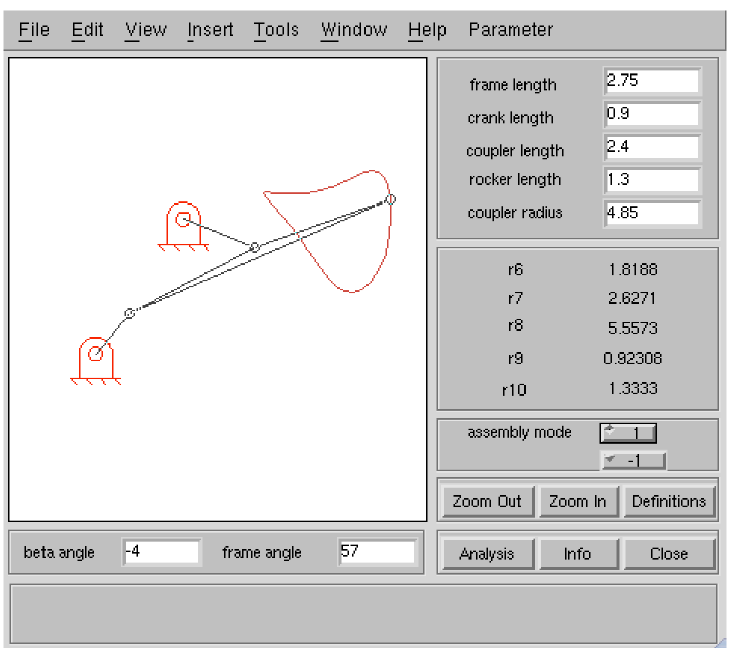


Fig. 2.10: Linkage for second mode

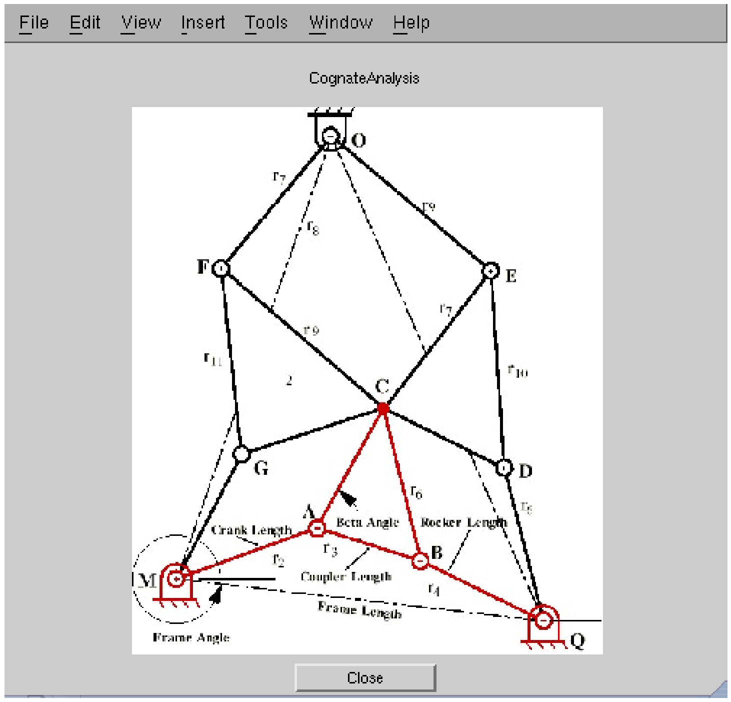


Fig. 2.11: Information page for cognates program

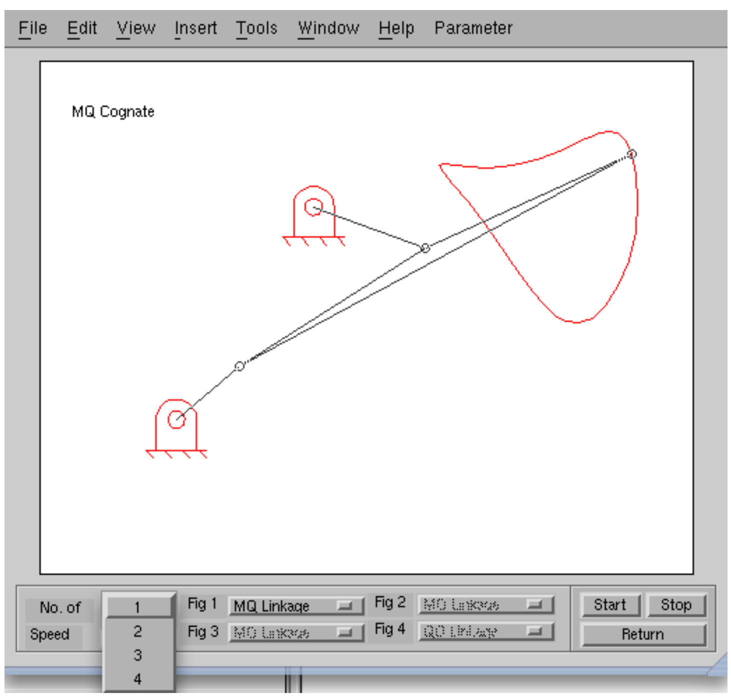


Fig. 2.12: Animation of single linkage

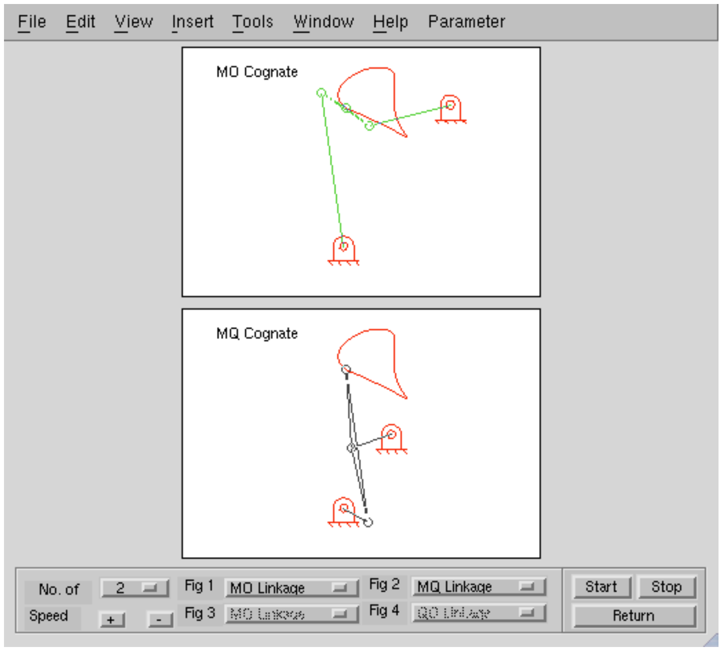


Fig. 2.13: Two cognate linkages.

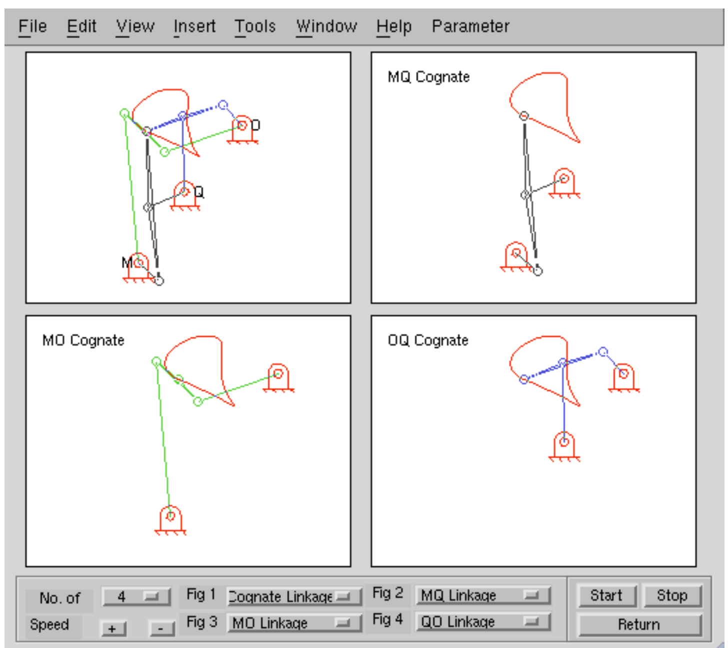


Fig. 2.14: The GUI analysis window for the cognates routine

2.4 Program for Designing a Double-Rocker Four-Bar Linkage (DoubleRockerDesign)

This routine facilitates the design of a four-bar linkage as a double rocker. The input information are the initial positions of the input and output links (rockers) and the input and output rocker angles. The equations are developed from Section 6.2.2 of the textbook.

The program is also structured in two windows, a design window (Fig. 2.15) and an analysis window (Fig. 2.16). In the analysis window, one or two plots can be displayed. These show the animated linkage and a plot of the output angle as a function of the input angle.

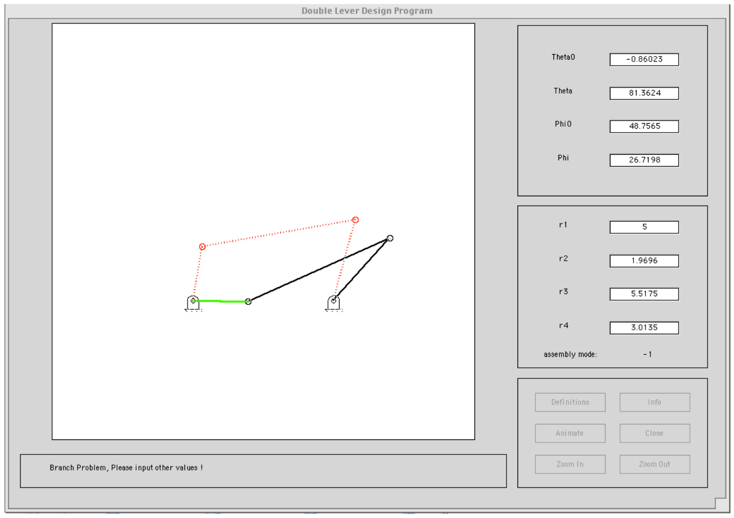


Fig. 2.15: The GUI design window for the double-rocker design routine

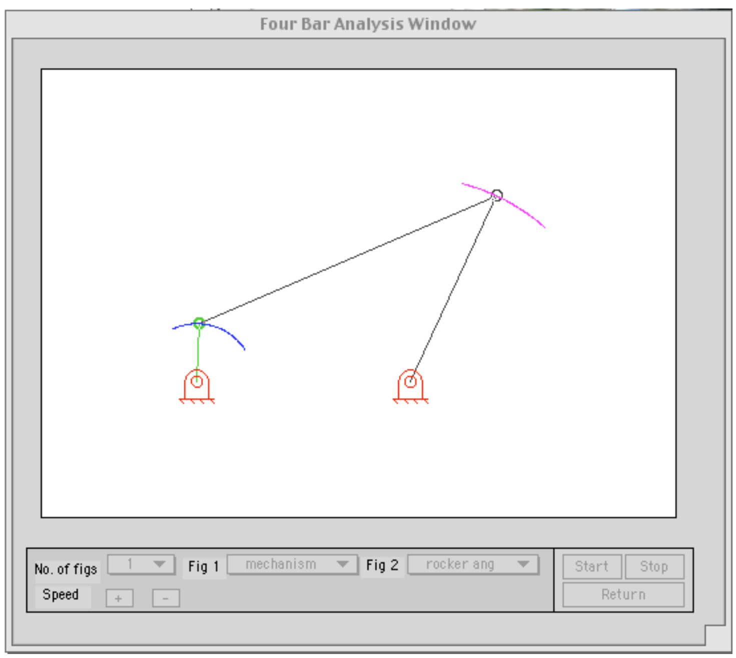


Fig. 2.16: The GUI analysis window for the double-rocker design routine

2.4.1 The Design Window for the Double-Rocker Design Program

The GUI displays the four-bar linkage on the left hand side of the design window. The editable angles (theta0, theta, phi0, phi) and link lengths (frame, crank, coupler, rocker) are grouped together on the right hand side of the window. If a linkage that will change branch is chosen, the message “Branch Problem, Please input other values” is displayed at the bottom of analysis window. Different values can be input either by typing in new values in the input boxes or by dragging the end points of the two rockers. In the drawing, the green link is taken as the driver corresponding to theta and theta0.

Two buttons *Zoom in* and *Zoom out* scale the plots because the parts of the mechanism might go outside of the plot window when the user drags the rocker points. If the *Info* button is chosen, general information about a double rocker is presented as shown in Fig. 2.17. If The *Definitions* button is chosen, a generic double-rocker linkage is displayed as shown in Fig. 2.18 is displayed.

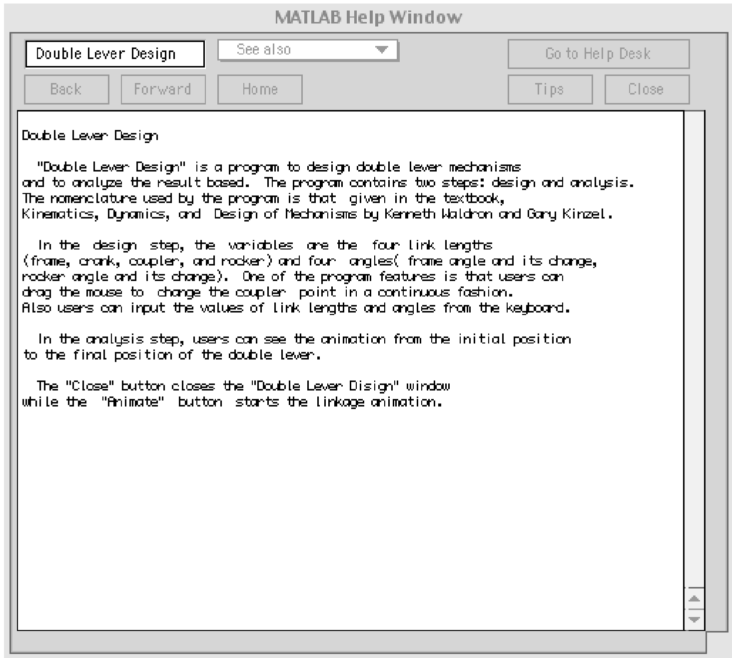


Fig. 2.17: The information window for the double-rocker design routine

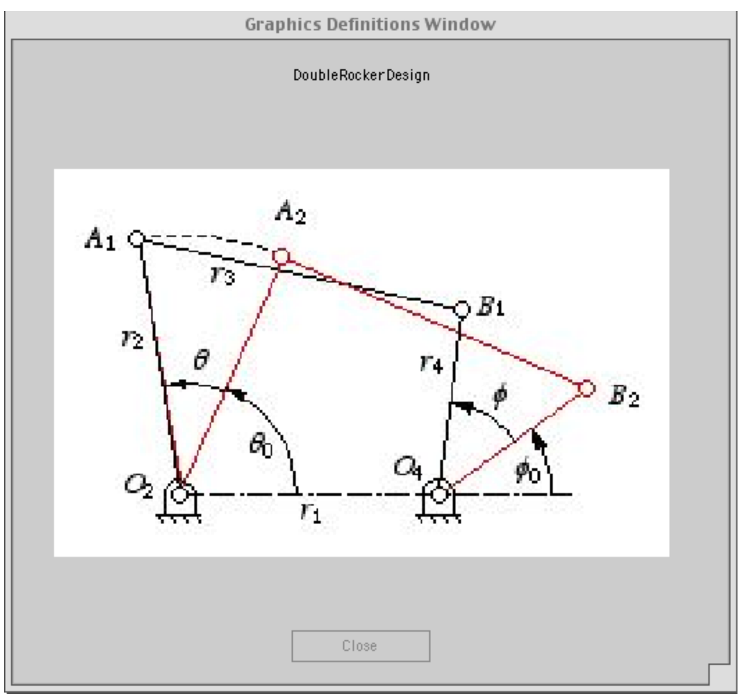


Fig. 2.18: The definition window for the double-rocker design routine

If the *Analysis* button is chosen, the linkage is animated. The animation will show that the design requirements are not met if a branch problem exists.

2.4.2 The Analysis Window for the Double-Rocker Design Program

The analysis window of the double-rocker design routine has the same layout as that of the crank-rocker routine except that only two plots are available. The results for one and two plots are shown in Figs. 2.16 and 2.19. To change from one to two plots or vice versa, it is necessary to press stop first if the animation is running. Then change the number of figures using the button indicated. The figures plotted are selected from the two titles for each of Fig 1 and Fig 2.

To change the speed of the animation, click on either the "+" or "-" button. Press multiple times to make a large change in the speed.

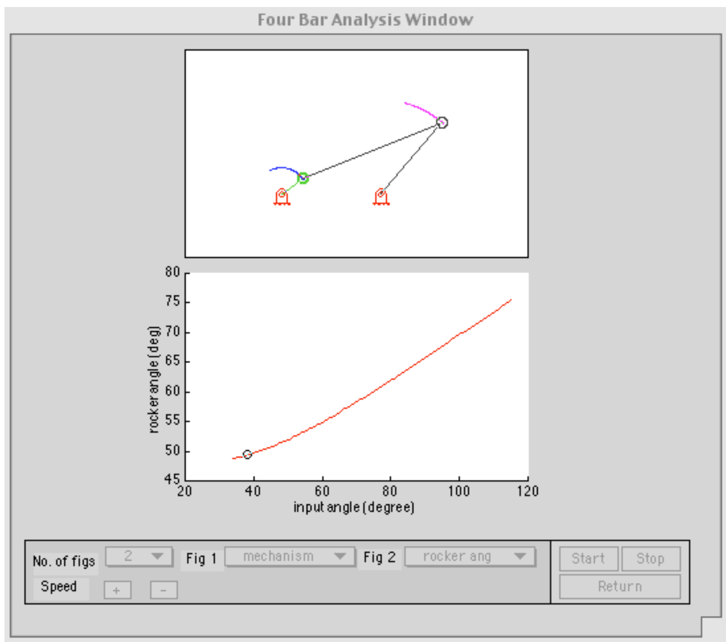


Fig. 2.19: The window for the double-rocker design routine when two plots are chosen

2.5 Program to Analyze a Four-bar Linkage (FourbarAnalysis)

This routine analyzes a four-bar linkage for which either a crank or the coupler can be specified as the driver. Associated analysis plots for the angular position of the rocker and coupler and the velocity of the rocker are shown in the animation. As in the previous two cases, the four-bar program is structured with a design window and an analysis window.

2.5.1 The Design Window for Four-bar Program

The design window is shown in Fig. 2.20. The design window has several radio button sets to set different features of the program. A frame groups each set. Otherwise, the design window is similar to that of the cognates GUI routine. The radio buttons are associated with the following options.

a) Line or triangle. The coupler can be drawn using either a line or a triangle. The coupler is represented by a line in Fig. 2.20 and by a triangle in Fig. 2.21.

b) Crank or coupler driven? Either the crank or the coupler can drive the linkage.

c) One mode or two? The linkage can be analyzed and the coupler curve displayed for either one mode or two. If only one mode is chosen, the coupler curve for that assembly mode only will be shown. If both modes are chosen, the coupler curve for both assembly modes will be displayed. This is shown in Fig. 2.22

d) Assembly mode? Either the +1 or –1 linkage assembly mode can be analyzed for positions and velocities.

The variables that must be input are shown in a figure if the *Definition* button is selected. The resulting figure is shown in Fig. 2.23.

All of the push buttons are the same as in the cognates GUI routine, which actually becomes the standard push-button sets for all the subsequent design windows. The capability of moving the coupler point continuously using mouse dragging is also included in this program.

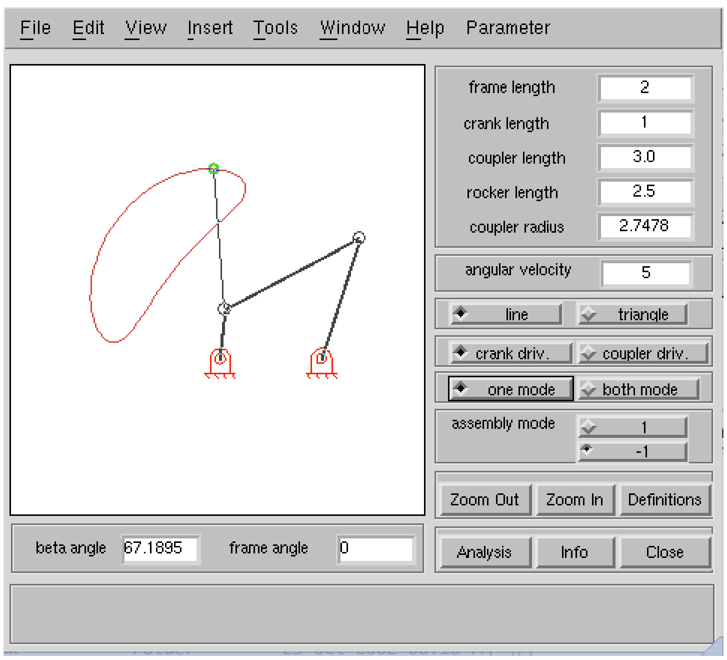


Fig. 2.20: Design window for the four-bar design, coupler represented by line

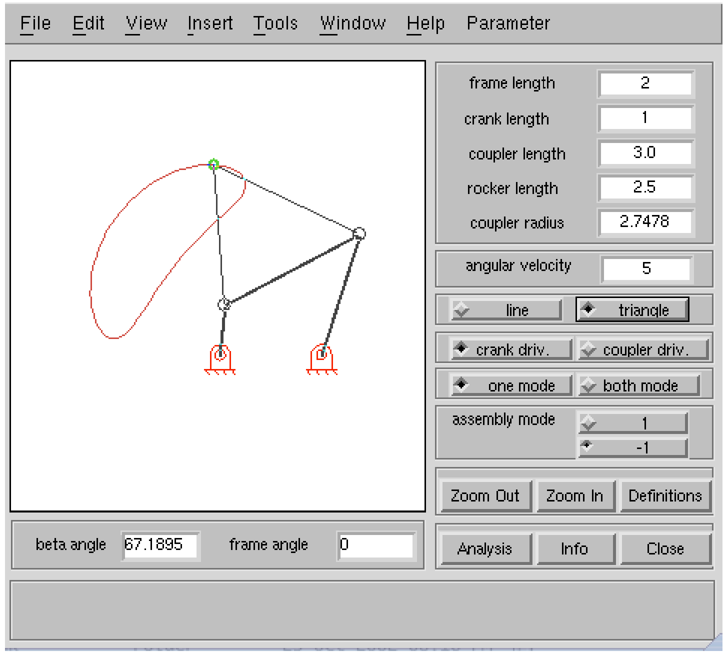


Fig. 2.21: Coupler represented by triangle

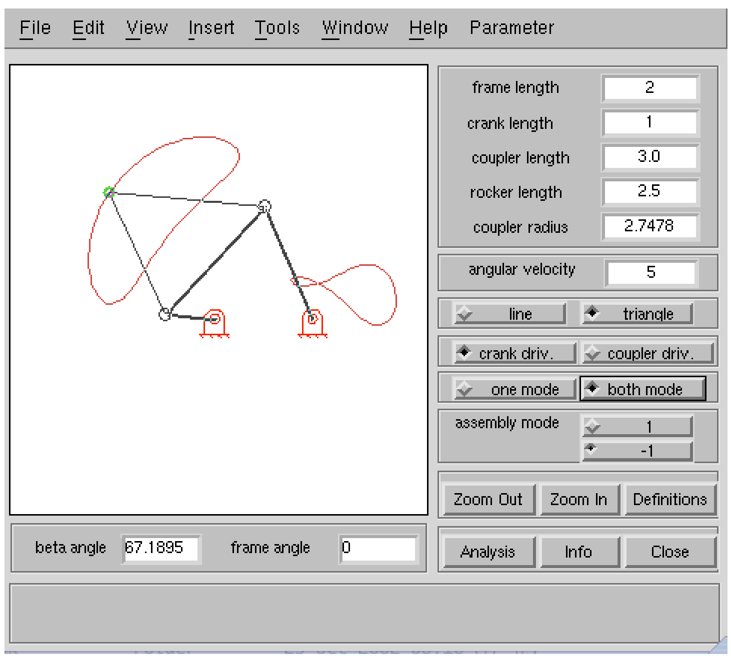


Fig. 2.22: Design window with coupler curve for both assembly modes

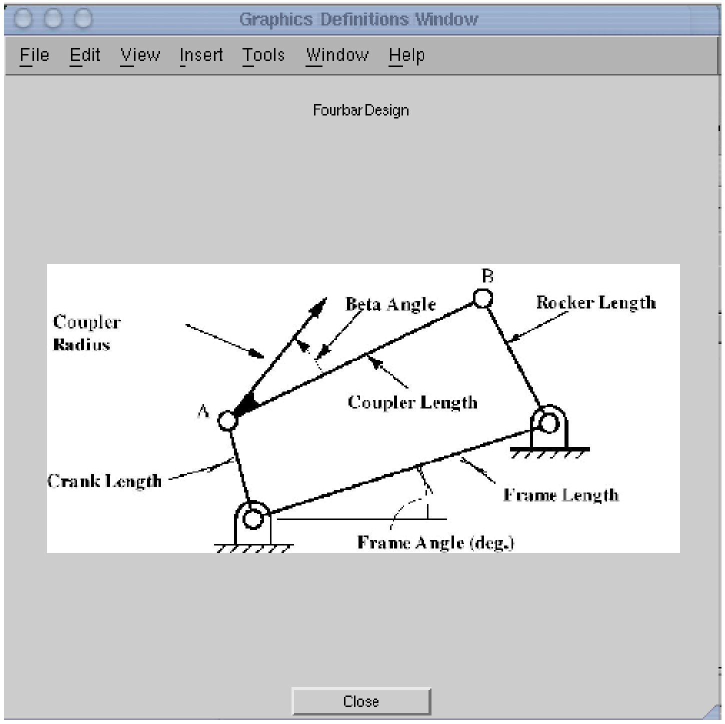


Fig. 2.23: Window showing variable definitions for four-bar analysis program

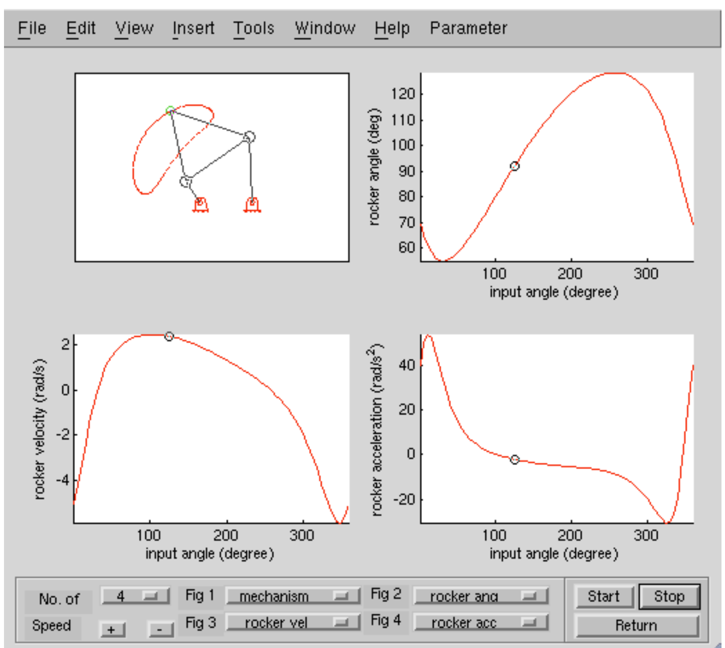


Fig. 2.24: Analysis window for the four-bar analysis program

2.5.2 The Analysis Window for Four-Bar Program

The analysis window is the same as those for the previous two analysis windows. An example is shown in Fig. 2.24. Either 1, 2, 3, or 4 figures can be plotted. The animation must be stopped before the figures can be changed. To resume the animation, select the *Start* button. To change the linkage design, select the *Return* button and return to the design window. Any of the linkage parameters can be changed before returning the analysis window. Note that the coupler curve in Fig. 2.24 is dashed. One dash corresponds to 5 degrees of crank rotation. By observing the lengths of the dashes, it is possible to estimate the relative speed of the coupler point as the linkage moves.

2.6 Program to Analyze a Slider-Crank linkage (SliderCrankAnalysis)

This routine analyzes a slider-crank mechanism, in which the link driver (slider, coupler, or crank) can be specified. Associated analysis plots for the position of the output link and coupler and the velocity of the output link are shown in the animation. As in the previous cases, the slider-crank program is structured with a design window and an analysis window.

2.6.1 The Design Window for Slider-Crank Program

The design window is shown in Fig. 2.25. Again, the design window has several radio button sets to set different features of the program. A frame groups each set. Otherwise, the design window is similar to that of the four-bar program. In the slider-crank program, the coupler point must be identified by a triangle. The radio buttons are associated with the following options.

a) Crank, coupler, or slider driven? The crank, coupler, or slider can drive the linkage.

b) Assembly mode? Either the +1 or –1 linkage assembly mode can be analyzed for positions and velocities. The assembly mode will have different meanings depending on which link is the driver.

c) One mode or two? The linkage can be analyzed and the coupler curve displayed for either one mode or two. If only one mode is chosen, the coupler curve for that assembly mode only will be shown. If both modes are chosen, the coupler curve for both assembly modes will be displayed. An example is shown in Fig. 2.26

The variables that must be input are shown in a figure if the *Definition* button is selected. The resulting figure is shown in Fig. 2.27.

All of the push buttons are the same as in the four-bar routine. The capability of moving the coupler point continuously using mouse dragging is also included in this program.

2.6.2 The Analysis Window for Slider-Crank Program

The analysis window is the same as the previous cases. An example is shown in Fig. 2.28. Either 1, 2, 3, or 4 figures can be plotted. The animation must be stopped before the figures can be changed. To resume the animation, select the *Start* button. To change the linkage design, select the return button and return to the design window. Any of the linkage parameters can be changed before returning the analysis window.

As in the case of the four-bar linkage program, the coupler curve for the slider-crank program is shown dashed. This feature allows the user to visualize the relative speed of the coupler point as it moves along the coupler curve.

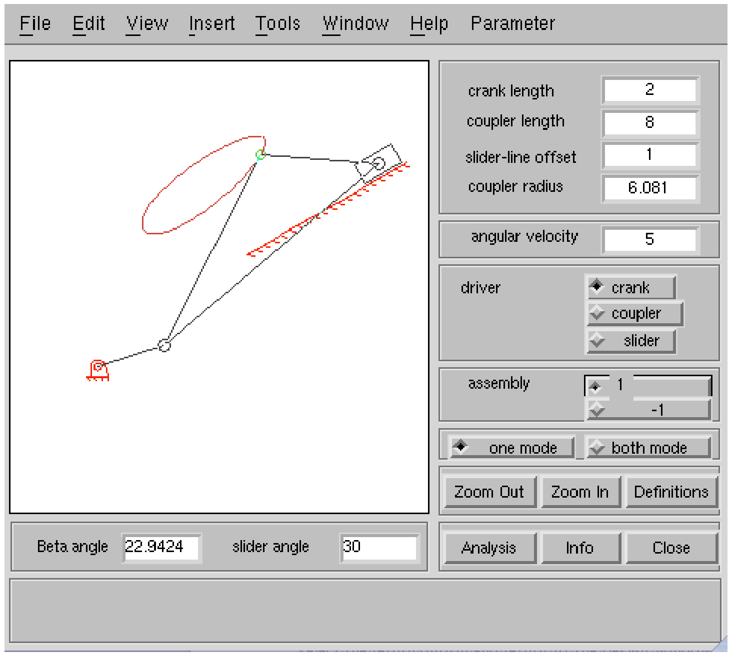


Fig. 2.25: Design window for the slider-crank program (slider driving)

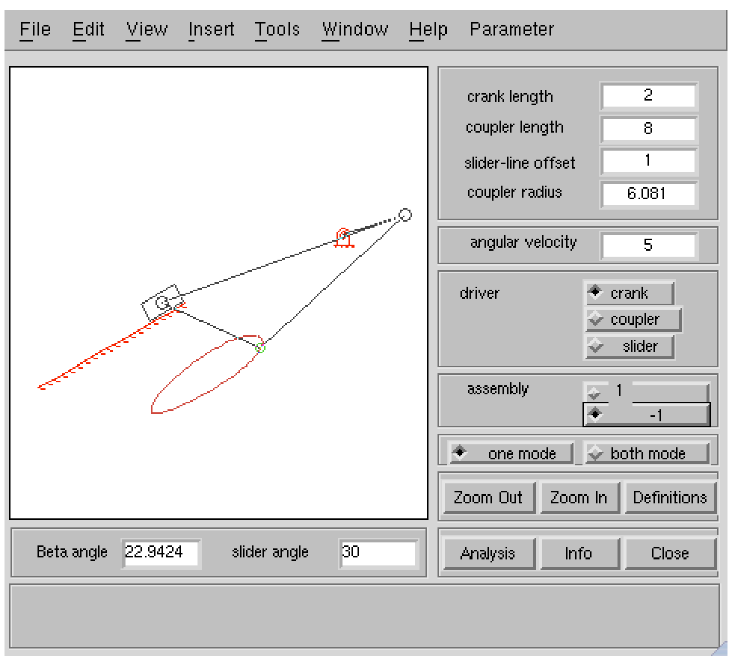


Fig. 2.26: Design window with second assembly mode

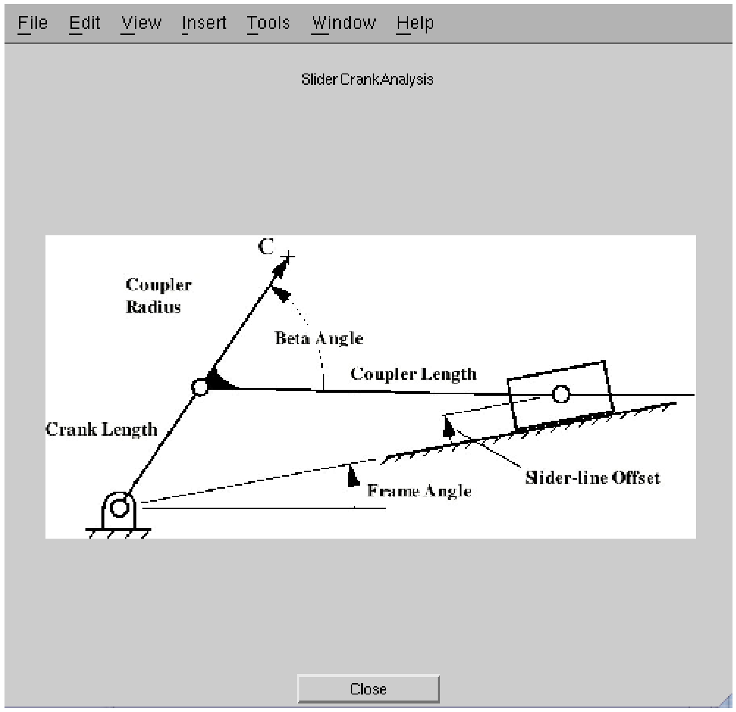


Fig. 2.27: Window showing variable definitions for slider-crank analysis program

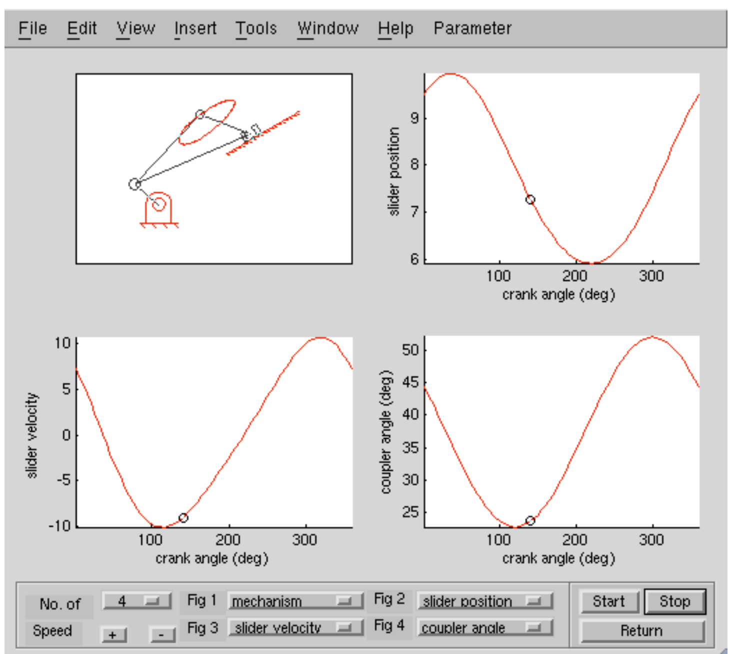


Fig. 2.28: Analysis window for the slider-crank analysis program

2.7 Program for Analyzing a Stevenson's Six-Bar Linkage (SixBarAnalysis)

This routine analyzes a Stevenson type six-bar. The analysis is conducted by treating the six-bar as an assembly of a four-bar, a rigid body and a dyad. As in the previous cases, the program is structured into a design window where the linkage information is input and an analysis window where the output is displayed graphically. This program is intended as an analysis program for the linkages designed using the procedure given in Section 6.6 of the textbook.

2.7.1 The Design Window for Six-Bar Program

The design window is shown in Fig. 2.29. Again the GUI layout is similar to the others except for the additional required input data. The location of the third bushing is an input from the user, and this program makes it mouse-movable. The coupler point is also mouse moveable.

The assembly mode refers to the output dyad of the six-bar linkage. The opposite assembly mode corresponding to Fig. 2.29 is shown in Fig. 2.30.

The variables that must be input are shown in a figure if the *Definition* button is selected. The resulting figure for the six-bar linkage is shown in Fig. 2.31. The remaining push buttons are the same as in the previous programs.

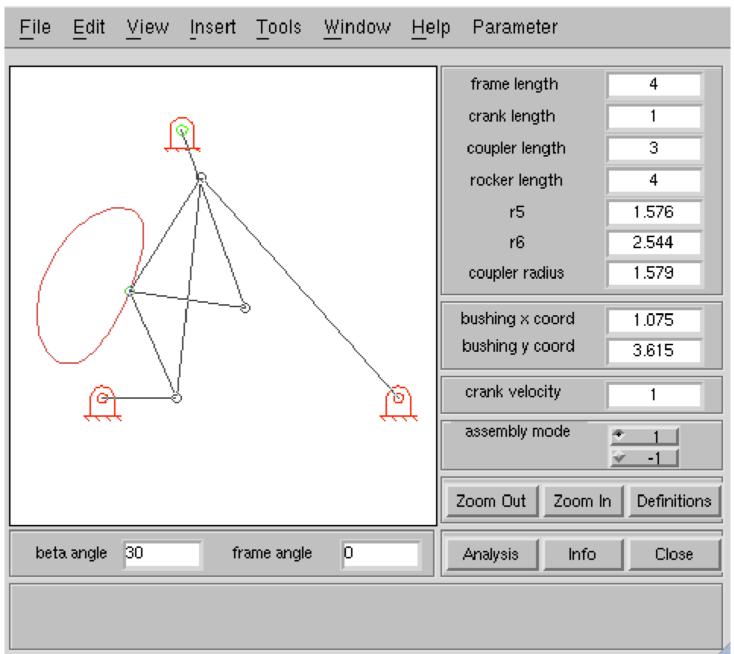


Fig. 2.29: The Design window for the six-bar design

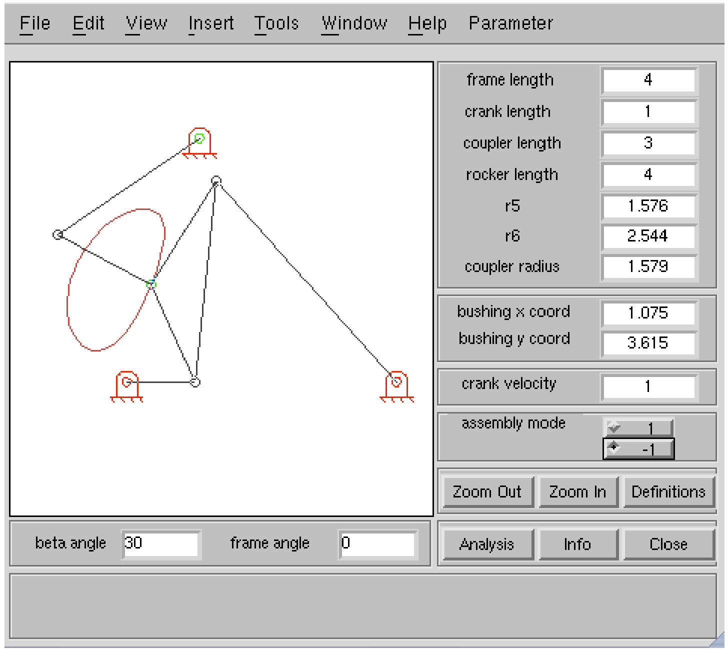


Fig. 2.30: The second assembly mode for six-bar output dyad

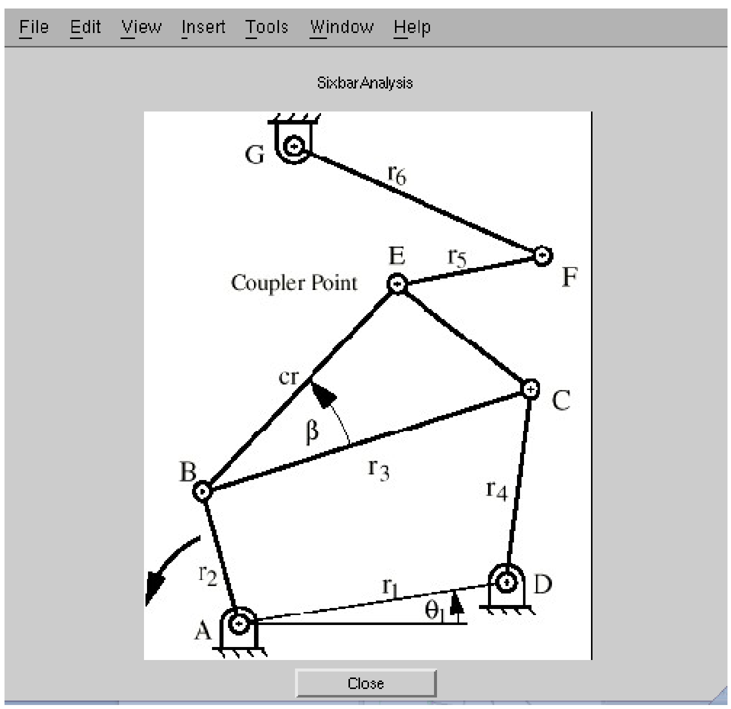


Fig. 2.31: Window showing variable definitions for six-bar program

2.7.2 The Analysis Window for Six-Bar Program

The analysis window can display 1, 2, 3, or 4 plots. The plot options are the mechanism, the rocker angle for the basic four-bar linkage, the dwell angle and the angular velocity for the rocker of the output dyad. An example is shown in Fig. 2.32. The animation must be stopped before the figures can be changed. To resume the animation, select the *Start* button. To change the linkage design, select the return button and return to the design window. Any of the linkage parameters can be changed before returning the analysis window.

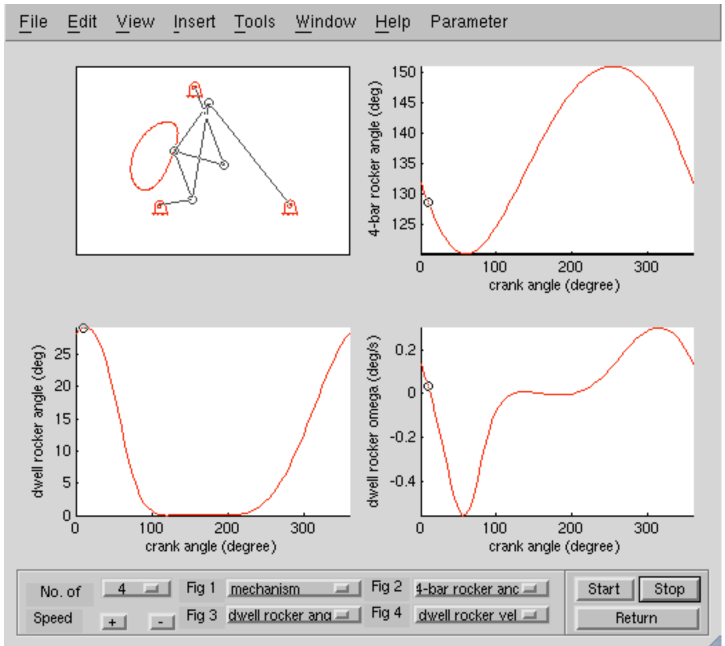


Fig. 2.32: Analysis window for the six-bar analysis program

2.8 Program for Generating Atlas of Coupler Curves for Four-Bar Linkage (HRCrankRockerAnalysis)

This routine is to generate the coupler curves for a four-bar mechanism. The program is called HRCrankRockerAnalysis after Hrones and Nelson who developed an atlas of coupler curves. A uniform grid of coupler points is assumed for the coupler, and the user can choose one point for analysis by following a sequence of selecting grid dimension, grid density, and row and column numbers.

The program uses three windows. The first is a design window where the linkage and coupler-point grid is defined. The next is an animation window that displays the coupler curves for the points identified in the analysis window. One of the coupler points can be selected for further analysis. The third window is the analysis window for the mechanism with the single coupler point that is selected.

2.8.1 The Design Window for Four-Bar Coupler-Point Analysis Program

The design window is shown in Fig. 2.33. In the design window, a uniform grid is created. When a point is selected, its grid marker changes to a hollow circle. The user can specify both the length and height of the coupler rectangular grid and the number of rows and columns of points. In addition, the user can select the specific grid points that will be analyze further. These are identified in the *Animation range*. The push-button set of *Definitions*, *Zoom Out* and *Zoom In* buttons is moved to the space below the plot because all the grid creation options are arranged to be close together. The *Definitions* page gives a description of most of the input variables. This page is shown in Fig. 2.34

The GUI program checks the valid range for the grid to avoid interrupting execution. If invalid data are inputted, the previous data are retrieved and an error message is shown in the status bar below the plot.

2.8.2 The Animation Window for Four-Bar Coupler-Point Analysis Program

The animation window (Fig. 2.35) displays the coupler curves for the coupler points identified in the design window. One of these coupler points can be selected via the mouse for further analysis. The color of the coupler point is changed when it is selected. In the animation window, the *Analysis* button is not available until the user selects the exact coupler point using the mouse.

2.8.3 The Analysis Window for Four-Bar Coupler-Point Analysis Program

The analysis window (Fig. 2.36) is the same as the other analysis windows. Up to four plots can be displayed. The plot options, shown in Fig. 2.36, are the mechanism, the rocker angle, the rocker velocity, and the rocker acceleration. The coupler curve shown in the analysis window is dished where each dash corresponds to 5 degrees of crank rotation. This gives a means of coordinating the travel of the coupler point to the position of the crank. Also, the length of the dashes gives a visual comparison of the relative speed of the coupler point as it moves along the coupler curve.

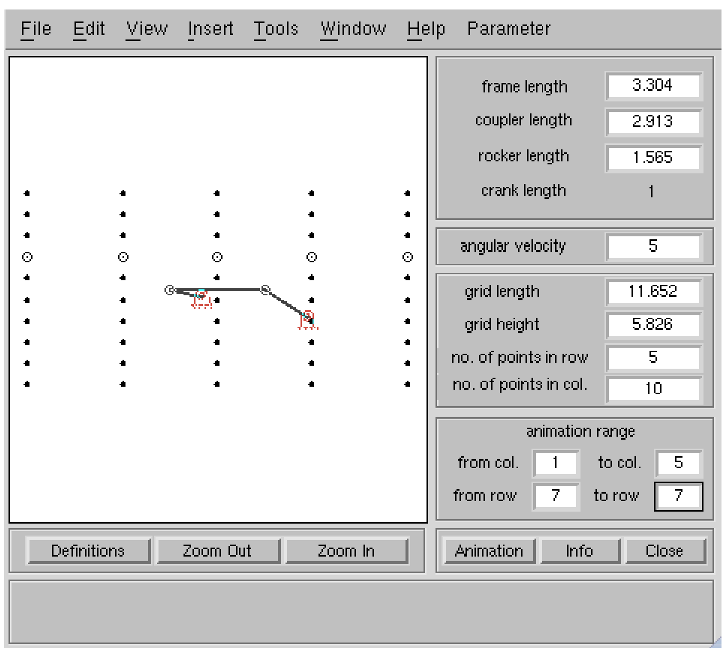


Fig. 2.33: Design window for four-bar linkage coupler-point atlas program

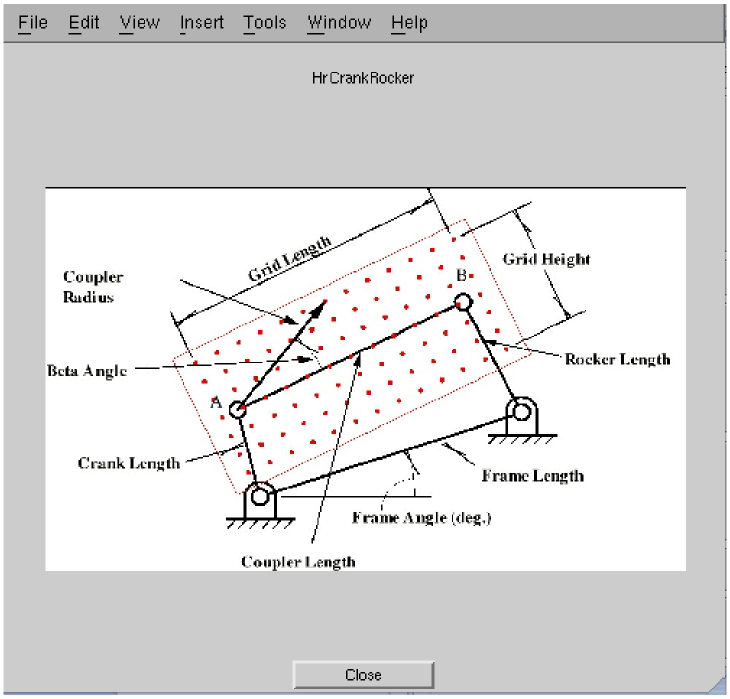


Fig. 2.34: Window showing variable definitions for four-bar linkage coupler-point atlas program

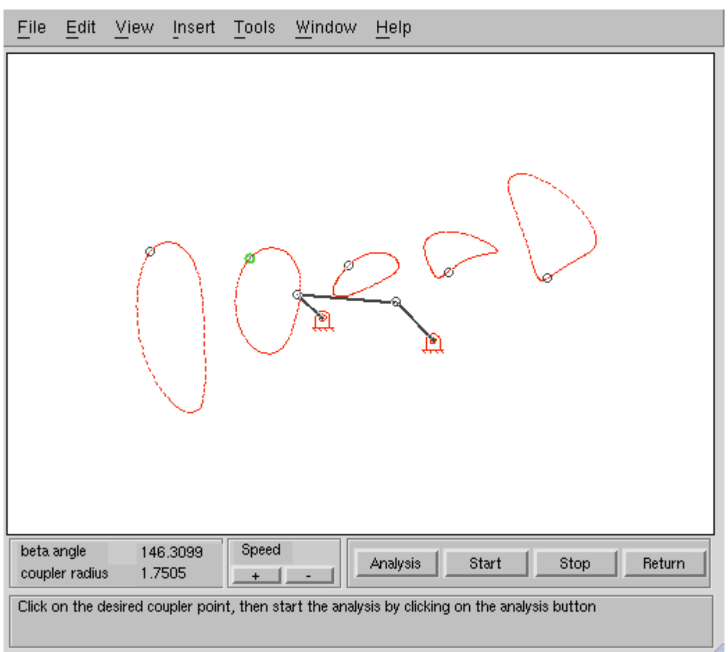


Fig. 2.35: Animation window for four-bar linkage coupler-point atlas program

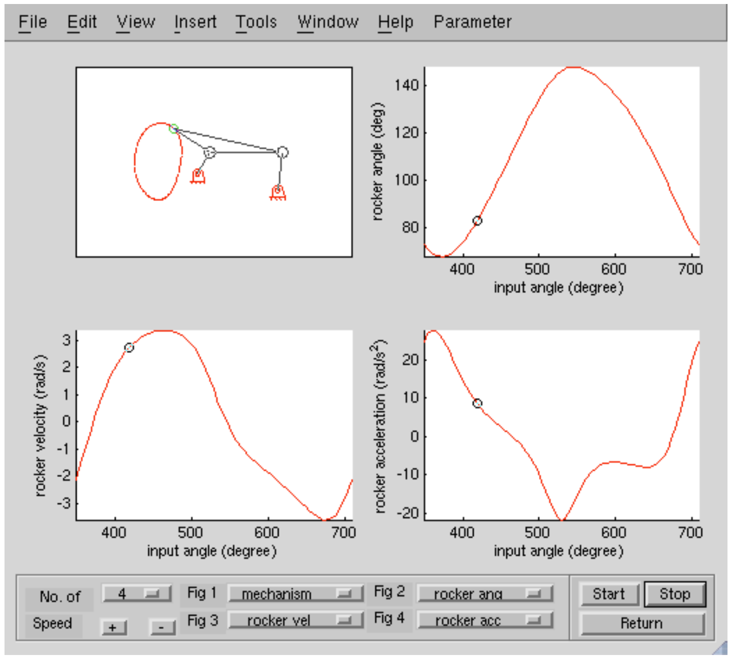


Fig. 2.36: Analysis window for four-bar linkage coupler-point atlas program

2.9 Program for Generating Atlas of Coupler Curves for Slider-Crank Mechanism (HRSliderCrankDesign)

This routine is to generate the coupler curves for a slider-crank mechanism. The program is called HRSliderCrankDesign after Hrones and Nelson who developed an atlas of coupler curves. A uniform grid of coupler points is assumed for the coupler, and the user can choose one point for analysis by following a sequence of selecting grid dimension, grid density and row and column numbers.

The program uses three windows. The first is a design window where the linkage and coupler-point grid is defined. The next is an animation window that displays the coupler curves for the points identified in the analysis window. One of the coupler points can be selected for further analysis. The third window is the analysis window for the mechanism with the single coupler point that is selected.

2.9.1 The Design Window for Slider-Crank Coupler-Point Analysis Program

The design window is shown in Fig. 2.37. In the design window, a uniform grid is created. When a point is selected, its grid marker changes to a hollow circle, adding visual assistance. The user can specify both the length and height of the coupler rectangular grid and the number of rows and columns of points. In addition, the user can select the specific grid points that will be analyze further. These are identified in the "animation range". The push-button set of 'Definitions', 'Zoom Out' and 'Zoom In' buttons is moved to the space below the plot because all the grid creation options are arranged to be close together. The definitions page gives a description of most of the input variables. This page is shown in Fig. 2.38

The GUI program checks the valid range for the grid to avoid interrupting execution. If invalid data are inputted, the previous data are retrieved and an error message is shown in the status bar below the plot.

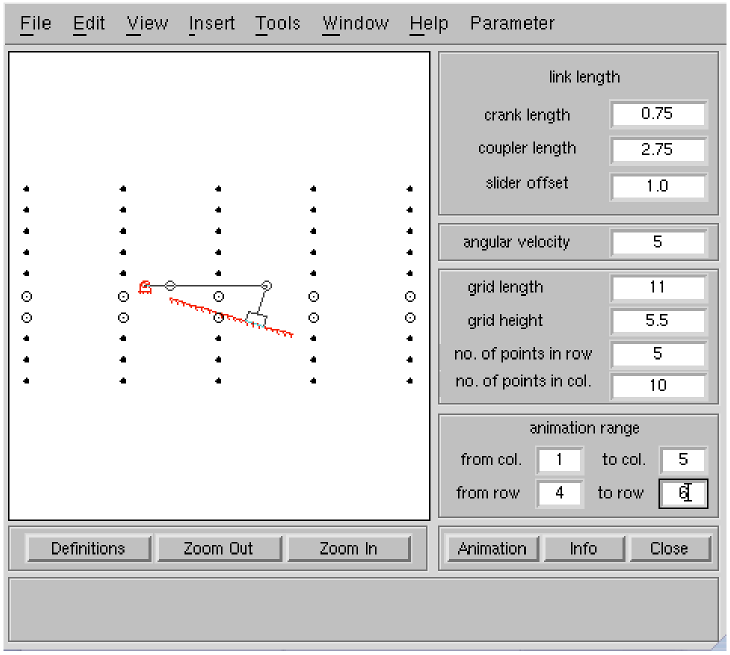


Fig. 2.37: Design window for slider-crank mechanism coupler-point atlas program

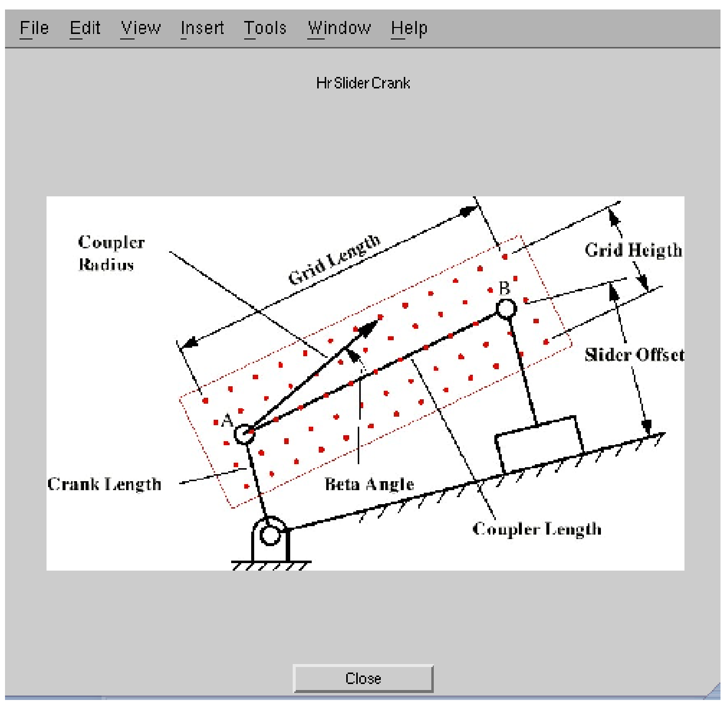


Fig. 2.38: Window showing variable definitions for slider-crank mechanism coupler-point atlas program

2.9.2 The Animation Window for Slider-Crank Coupler-Point Analysis Program

The animation window (Fig. 2.39) displays the coupler curves for the coupler points identified in the design window. One of these coupler points can be selected for further analysis. The color of the coupler point is changed when it is selected. In the animation window, the *Analysis* button is not available until the user selects the exact coupler point using the mouse.

2.9.3 The Analysis Window for Slider-Crank Coupler-Point Analysis Program

The analysis window (Fig. 2.40) is the same as the other analysis windows. Up to four plots can be displayed. The plot options, shown in Fig. 2.40, are the mechanism, the slider distance, the magnitude of the coupler velocity, and the magnitude of the coupler acceleration. The coupler curve shown in the analysis window is dished where each dash corresponds to 5 degrees of crank rotation. This gives a means of coordinating the travel of the coupler point to the position of the crank. Also, the length of the dashes gives a visual comparison of the relative speed of the coupler point as it moves along the coupler curve.

2.10 Program for Analyzing Four-Bar Linkage Centrodes

This routine generates the fixed and moving centrodes for the coupler of a four-bar linkage given the linkage geometry. The program consists of a design window where the linkage geometry is defined, and an animation window where the motion is animated.

2.10.1 Design Window for Centrode Program

The GUI layout for the design window (Fig. 2.41) is quite simple compared to other examples. Different centrodes can be generated by changing the link lengths for the four-bar linkage. A simple definition of terms is displayed if the *Definitions* button is selected. The window is shown in Fig. 2.42.

2.10.2 Analysis Window for Centrode Program

The analysis window (Fig. 2.43) animates the motion of the linkage. This shows that apparent rolling of the moving centrode on the fixed centrode as the linkage moves.

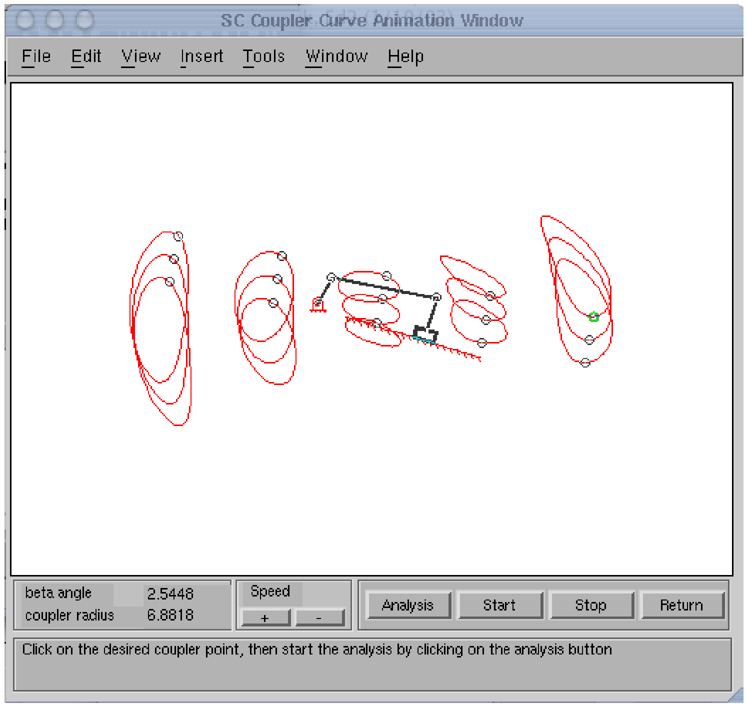


Fig. 2.39: Animation window for slider-crank mechanism coupler-point atlas program

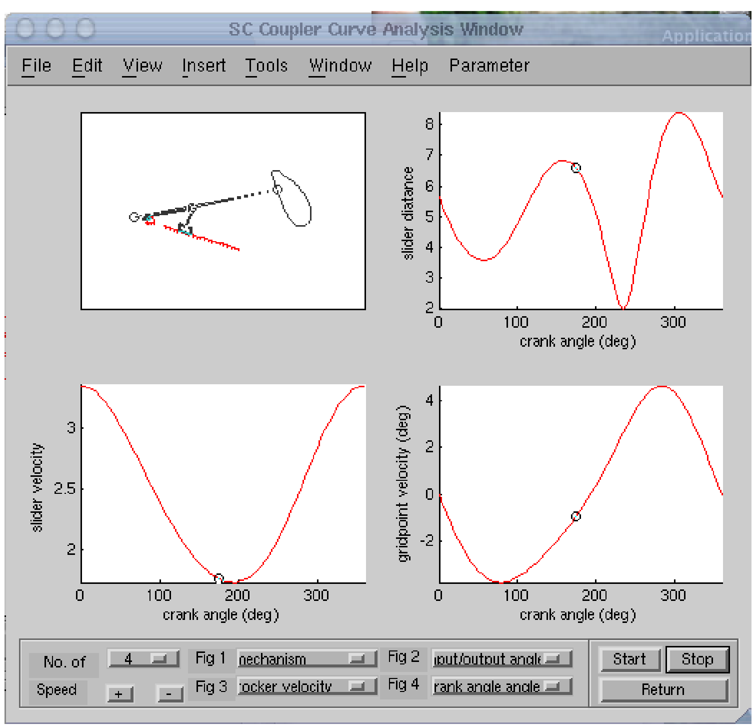


Fig. 2.40: Analysis window for slider-crank mechanism coupler-point atlas program

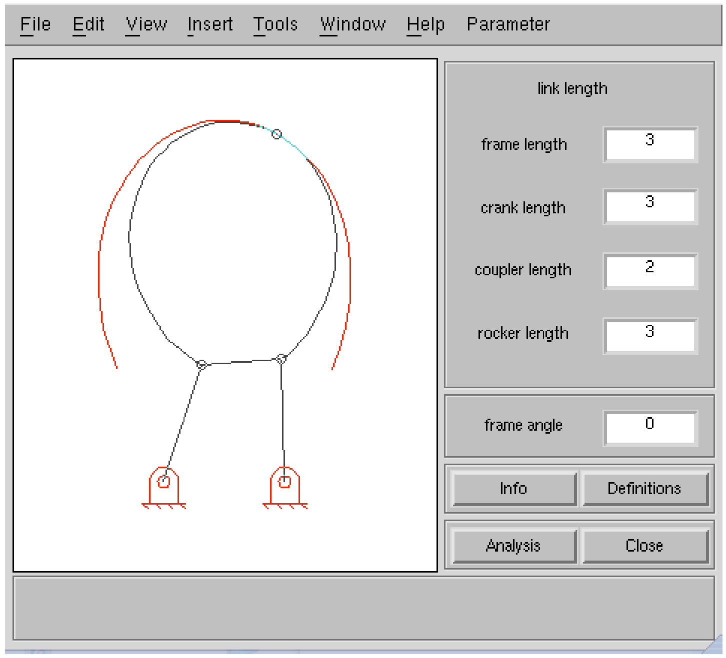


Fig. 2.41: Design window for the centrodes routine

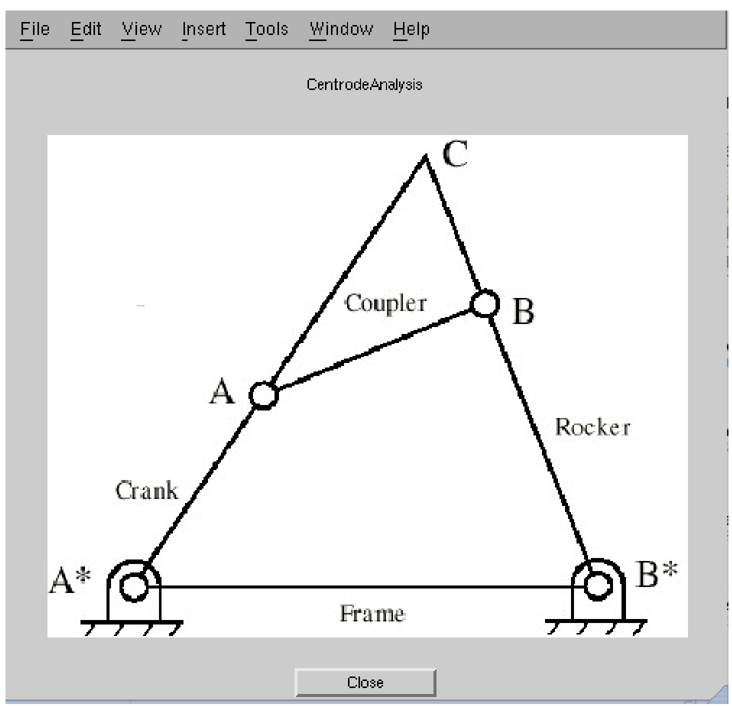


Fig. 2.42: Window showing variable definitions for centrodes program

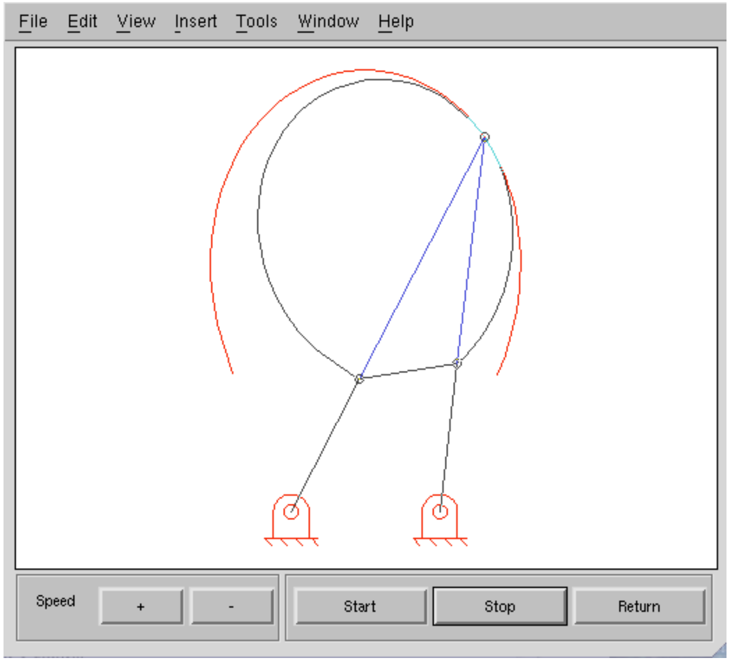


Fig. 2.43: Analysis window for the centrodes routine

2.11 Program for Analyzing Path Curvature (Inflection4barAnalysis)

This routine graphically displays the solution of the Euler-Savary equation for a four-bar linkage. The Euler-Savary equation gives a relationship between points in the coupler of a four-bar linkage and their centers of path curvature. Because the theory for path curvature is not covered in the textbook, a brief description of the procedure is given in Appendix A.

The inflection circle routine is developed in a single window of the GUI program as shown in Fig. 2.44. In the graphics window, the four-bar linkage and the inflection circle is displayed. The coupler point is designated by a green circle, and the center of the coupler point's path by a red cross. The coupler point is mouse moveable, and the center of the path is dynamically updated as the coupler point is moved.

The inflection circle changes with the position of the linkage. Moving the green crank with the mouse will change the position of the linkage. Simply click the mouse near the joint between the crank and the coupler and drag the link. A second position is shown in Fig. 2.45. In the frame corresponding to the point coordinates, "A" designates the coupler point and "Astar" denotes the center of path. The x and y numerical values for "A" may be input.

A graphical description of the input variables is displayed when the *Definitions* button is selected. This is shown in Fig. 2.46.

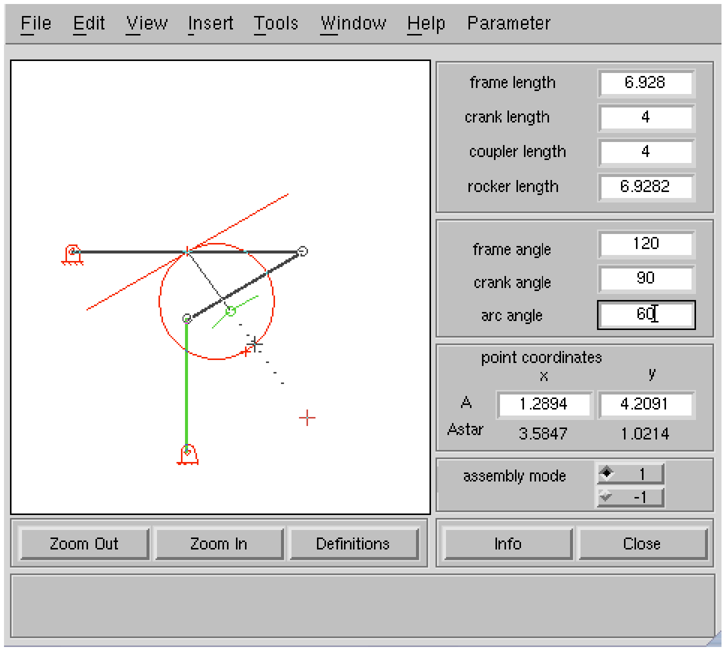


Fig. 2.44: Design window for the inflection-circle routine

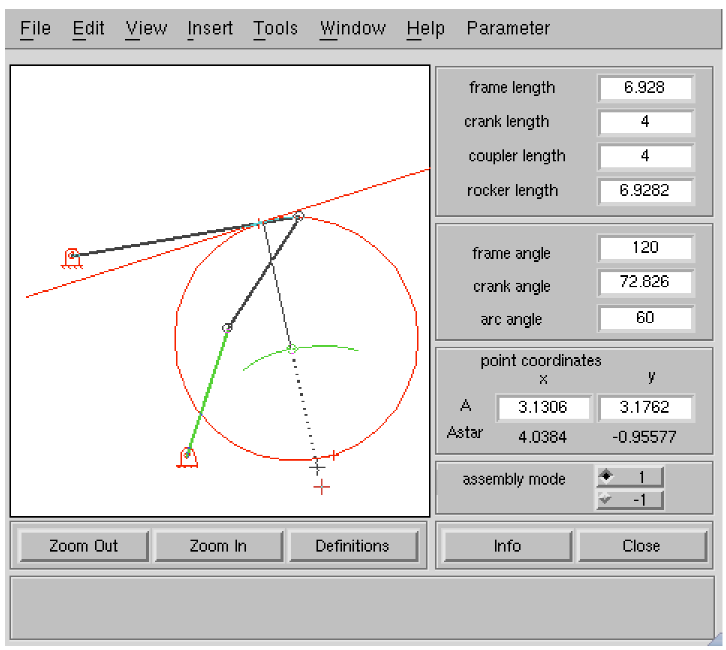


Fig. 2.45: Inflection circle when crank and coupler positions changes

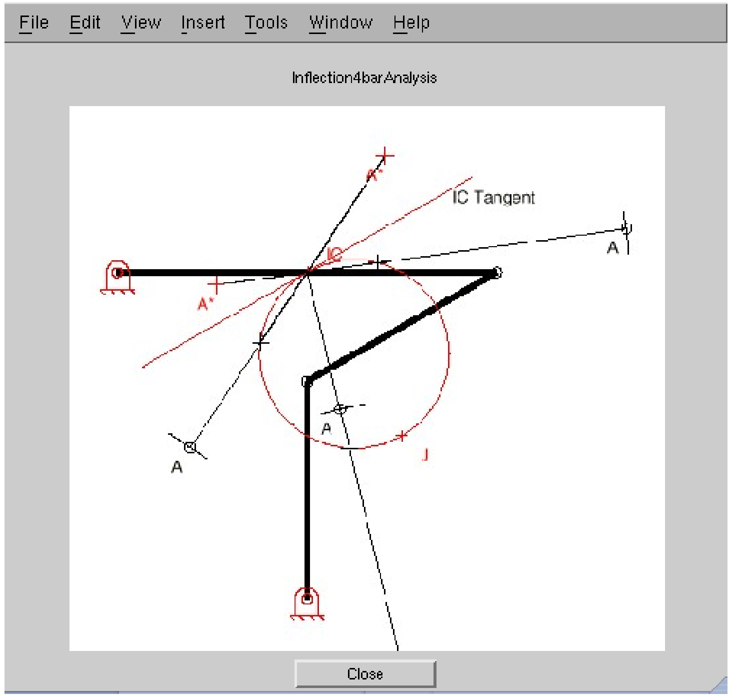


Fig. 2.46: Window showing variable definitions for inflection-circle program

2.12 Program for Analyzing the Shaking Force in a Slider-Crank Program (ShakeAnalysis)

This routine analyzes the slider-crank mechanism for position, velocity, and acceleration for one–degree increments of the crank. In addition, the shaking force is computed at each angle increment for both the given value of the counter-balance weight and for zero counter-balance weight. The optimum value of the counter-balance weight is also determined. As in the cases of the majority of the programs, the shaking force program is divided into a design window and an analysis window.

2.12.1 The Design Window for Slider-Crank Shaking-Force Program

The design window is shown in Fig. 2.47. The basic mechanism is a slider crank, and the input motion is similar to that for the slider-crank program in Section 2.6. This routine is focused on the calculation of the shaking force, the counter-balance weight, and its optimization. A large space below the plot is utilized to output the numerical results associated with the shaking force. The output data are in blue for emphasis. In addition to the link lengths, the acceleration of gravity and the weights of the crank, coupler, piston, and counter-balance weight must be input. It is assumed that the weights and the acceleration of gravity are in consistent units.

As in the case of the other programs, a graphics window that is displayed when the Definitions button is selected defines most of the variables. This window is shown in Fig. 2.48.

2.12.2 The Analysis Window for Slider-Crank Shaking-Force Program

The analysis window is shown in Fig. 2.49. Again, up to four plots can be displayed at one time. The display options are the mechanism, the polar shaking force diagram for no counterbalance weight, the shaking force diagram for the given counterbalance weight, and the shaking force for the optimum counterbalance weight. The optimum counterbalance is the one that produces the minimum value for the maximum shaking force magnitude.

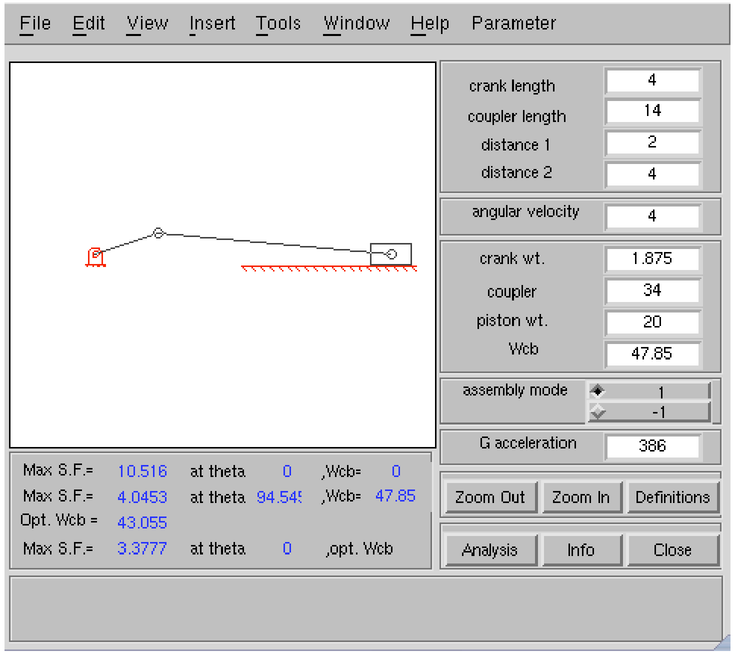


Fig. 2.47: Design window for the shaking force routine

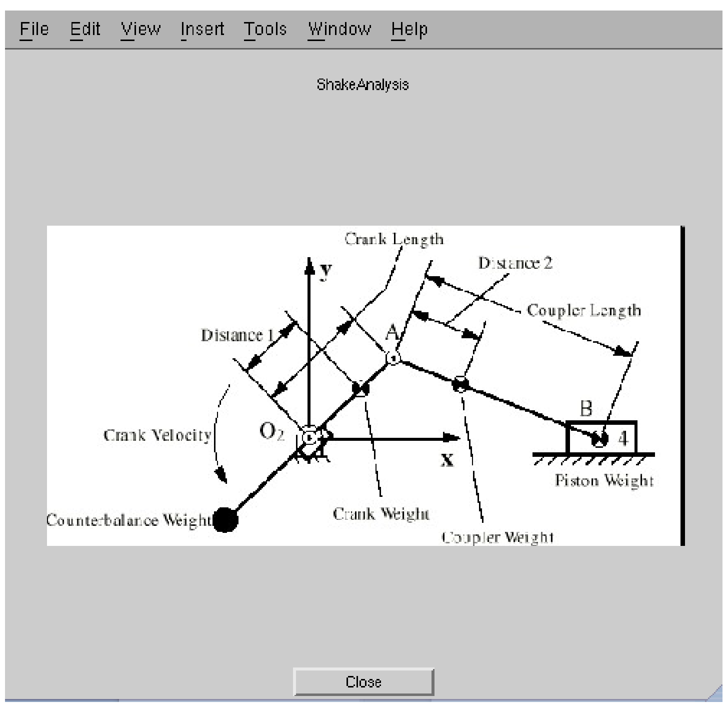


Fig. 2.48: Window showing variable definitions for shaking force program

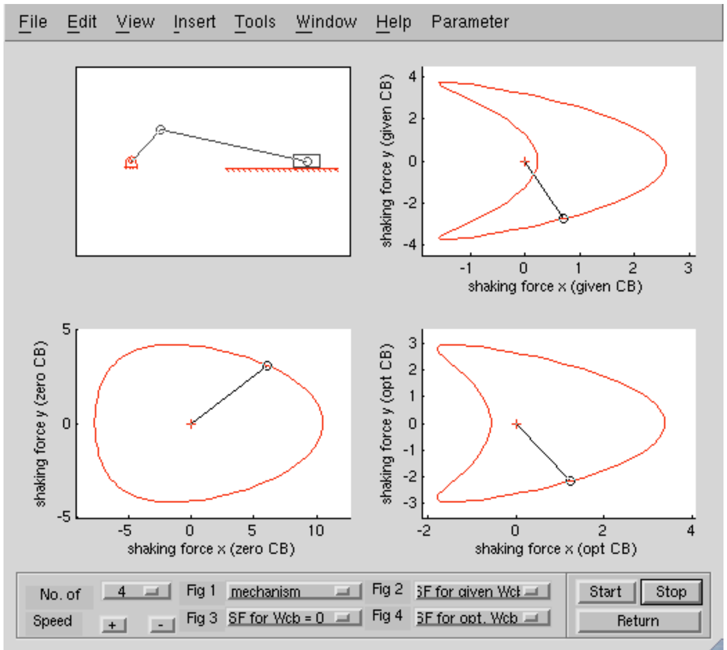


Fig. 2.49: Analysis window for the shaking force routine

3.0 Program for Cam Design

3.1 Introduction

The cam design program is called Cam2. This program allows the user to select the cam follower to be either translating or oscillating, and also allows flat-faced or roller-faced followers. The follower motion types included in the program are uniform, harmonic, cycloidal, and polynomial. The program also includes two procedures for optimizing the follower motion. Finally, the program generates the cam profile using the procedures described in the textbook.

This manual will include only a description of how to use the program. It is assumed that the reader is familiar with the material in Chapter 8 of the textbook. The program itself was written by Michael Stevens as part of the research associated with his MS thesis. His thesis is entitled, *Interactive Design of Plate Cams with Optimal Acceleration Characteristics* and was completed in 2002. It is available through The Ohio State University.

3.2 Cam-Follower and Motion-Specification Window

The program employs a graphical user interface and has three separate windows. The first of these windows (Fig. 3.1) is where the user specifies the cam follower type and motion specification conditions.

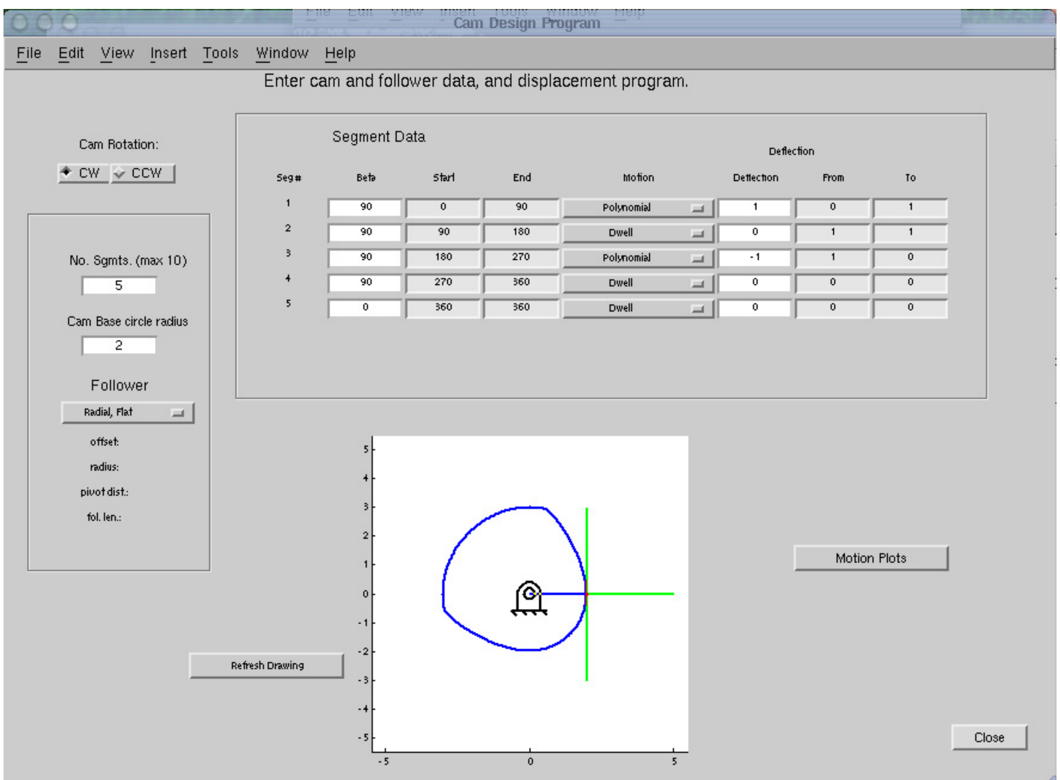


Fig. 3.1: Window for specifying the follower type and motion specification

This first screen allows the program's user to design the cam-follower system and to specify the follower's motion program. The direction of the cam's rotation, cam base circle radius, follower type, and follower parameters may all be input here. From a pull-down menu, shown in Fig. 3.2, the follower type can be chosen from the four available. The follower motion can be specified segment-by-segment in the box in the center of the screen, shown in Fig. 3.1. A follower motion type can be specified for each segment from the pull-down menu shown in Fig. 3.3.

In the specification window in Fig. 3.1, “beta” gives the duration of the cam rotation corresponding to the type of displacement chosen. A continuous rotation cam is assumed. Therefore, “theta” is assumed to start at 0 and to end at 360. The input angles are in degrees. Note that any of the values “beta”, “Start”, and “End” may be input where “Start” is the initial angle for the cam rotation and “End” is the end value for the range being considered. Once a value is input, the values that can be computed are computed.

Similarly, the follower deflection is assumed to start and stop at zero. For any given segment, either the ending value can be given or the total deflection. Again, after each value is input, the program will compute any value that it can based on the inputted value.

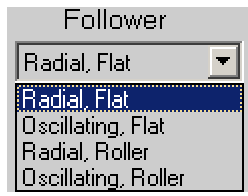


Fig. 3.2: Follower Type Pull-Down Menu

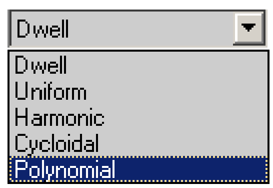


Fig. 3.3: Follower Motion Type Pull-Down Menu

After the cam and follower parameters are specified and the follower motion is defined, the cam and follower can be displayed. By clicking the *Refresh Drawing* button shown in Fig. 3.1, the user can update the drawing to show the cam follower system defined by the current inputs. When the cam-follower system and motion program is satisfactory, select the *Motion Plots* button to advance the program to its next window.

3.3 Motion Plots Window

The motion plots window is shown in Fig. 3.4. This second screen shows the follower’s displacement, velocity, acceleration, and jerk plots. In addition, this screen allows the option to apply an optimization procedure to the follower’s motion. The optimization method can be chosen from the pull-down menu displayed in Fig. 3.5

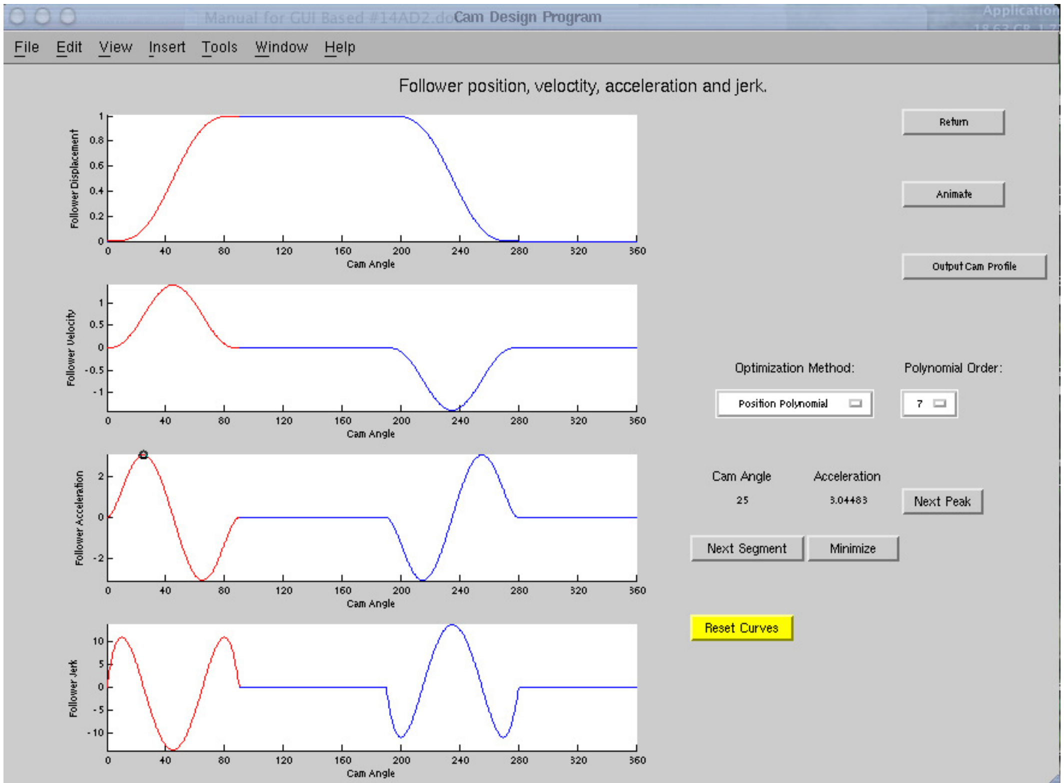


Fig. 3.4: Cam motion plots window

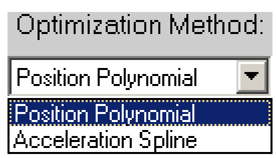


Fig. 3.5: Optimization Method Pull-Down Menu

The optimization procedure minimizes the maximum acceleration in any given segment of the curve. Two methods can be used. The first method (Position Polynomial) approximates the segment selected by a polynomial of *n* order where *n* is specified by the user and is larger than 6. Values allowed are 7, 8, 9, and 10. The optimization procedure uses six of the polynomial coefficients to match the position, slope, and curvature conditions at both ends of the segment. The remaining coefficients are then selected to minimize the maximum acceleration in the range of the segment.

The second optimization procedure involves beginning with an initial profile and then fitting a set of splines to the acceleration curve. The control points of the splines are used as the design variables in optimizing the acceleration curve for minimum acceleration. The control points are selected such that continuity is maintained in position, velocity, and acceleration at both ends of the segment. Both procedures work well, and it normally does not matter which procedure is chosen for the optimization. Typically, the improvement in the acceleration curve is modest, but it is worth the effort because it is fast and all functional requirements are satisfied.

The segment of the curve to optimize can be selected by clicking the *Next Segment* button until the desired segment is selected. As segments are selected, they change from blue to red. By clicking the *Minimize* button, the selected segment will be minimized using the specified optimization procedure. Also, the numerical value of the acceleration peaks can be displayed by clicking the *Next Peak* button until the desired peak acceleration is indicated by the circle, as in Fig. 3.6.

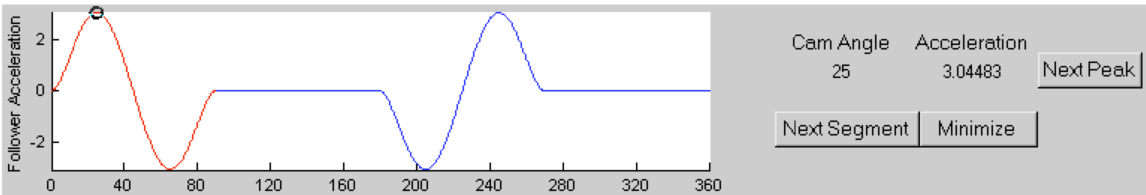


Fig. 3.6: Peak Acceleration Selection

If the result of the optimization is undesirable the *Reset* button can be used to reset the follower motion to that specified in the first screen. When the follower motion is acceptable, the *Animation* button advances the program to the animation window.

3.4 Cam-Follower Animation Window

This third, and final, screen is displayed in Fig. 3.7. On this third screen, the cam-follower system can be animated using the *animate* button, and the speed of the cam can be adjusted using the speed controls. These speed controls include a + button for increasing the speed, and a - button for decreasing the speed. One click of either button changes the speed by 10% of its current value. The *Reset Speed* button resets the speed to its initial value. From the animation screen, the *Motion Plots* button will return the program to the previous screen, shown in Fig. 3.4. Finally, the *Output Cam Profile* button outputs the cam profile to a text file titled "cam_profile" and located in the same directory as the program. The coordinates of the points are given by ordered triples of numbers (x, y, z) where z is always 0.0

3.5 Radial Roller Follower Example

In this example, the follower's motion is defined by a dwell from 0° to 90°. Then, the follower rises with cycloidal motion during the rotation of the cam from 90° to 180°. The follower then dwells for 60° of cam rotation, and then returns with simple harmonic motion for the cam rotation from 270° to 360°. The amplitude of the follower translation is 2 cm, and the follower radius is 1 cm. The base circle radius of the cam is 4 cm, and the offset is 0.5 cm. Finally, the cam's direction of rotation is clockwise.

Once the follower motion is entered into the design program, the Segment Data looks like Fig. 3.8. The cam and follower data appears as it does in Fig. 3.9. Finally, the cam-follower system is shown in Fig. 3.10.

The motion plots for the follower motion specified in Fig. 3.8 look like those in Fig. 3.11.

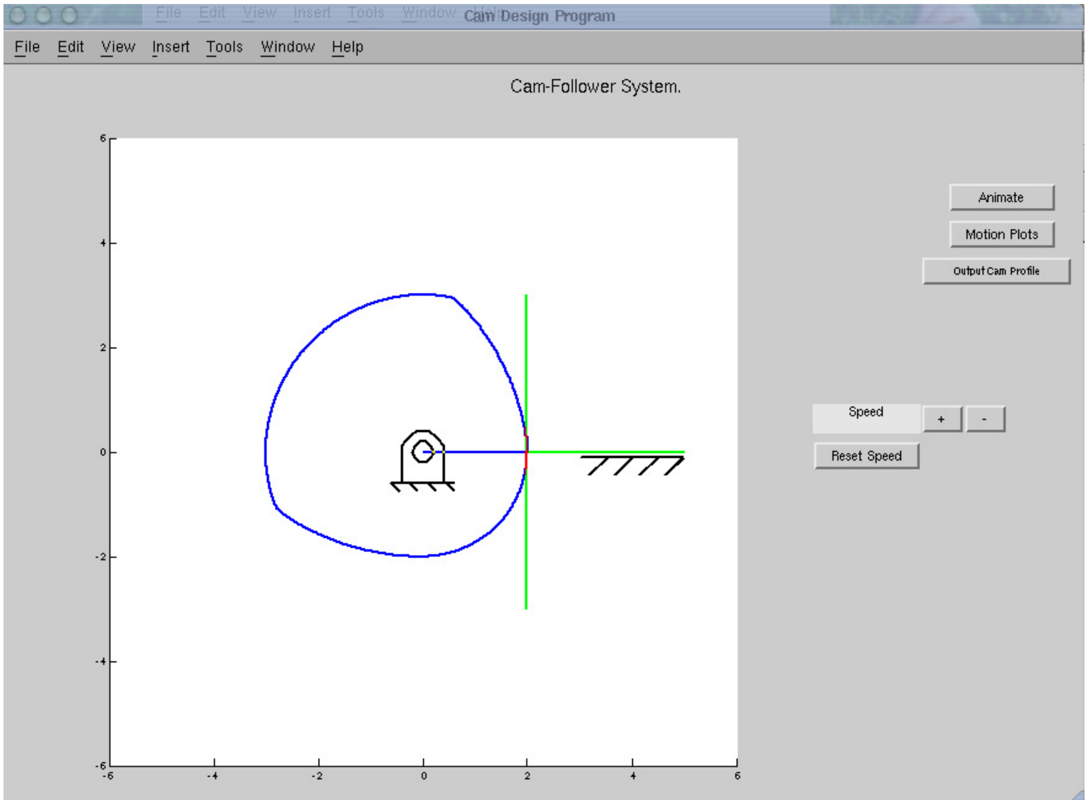


Fig. 3.7: Cam – follower animation window

Segment Data

Seg #	Beta	Start	End	Motion	Deflection	Deflection From	Deflection To
1	90	0	90	Dwell	0	0	0
2	90	90	180	Cycloidal	2	0	2
3	60	180	240	Dwell	0	2	2
4	120	240	360	Harmonic	-2	2	0

Fig. 3.8: Segment Data for Radial Roller Example

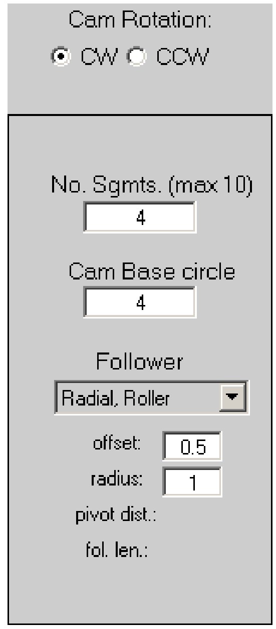


Fig. 3.9: Cam and Follower Data for Radial Roller Example

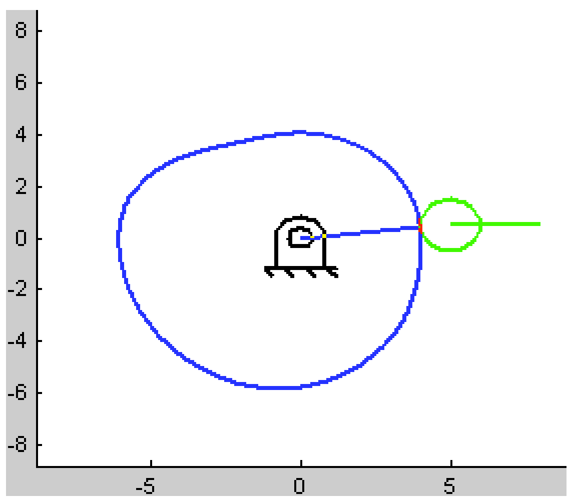


Fig. 3.10: Cam-Follower System for Radial Roller Example

The motion program can be optimized. In this example, the acceleration splines method of optimization is applied to the rise segment. The optimization parameters are shown in Fig. 3.12. After the optimization the peak acceleration is reduced from 5.09 to 4.70. The optimized motion plots are displayed in Fig. 3.13.

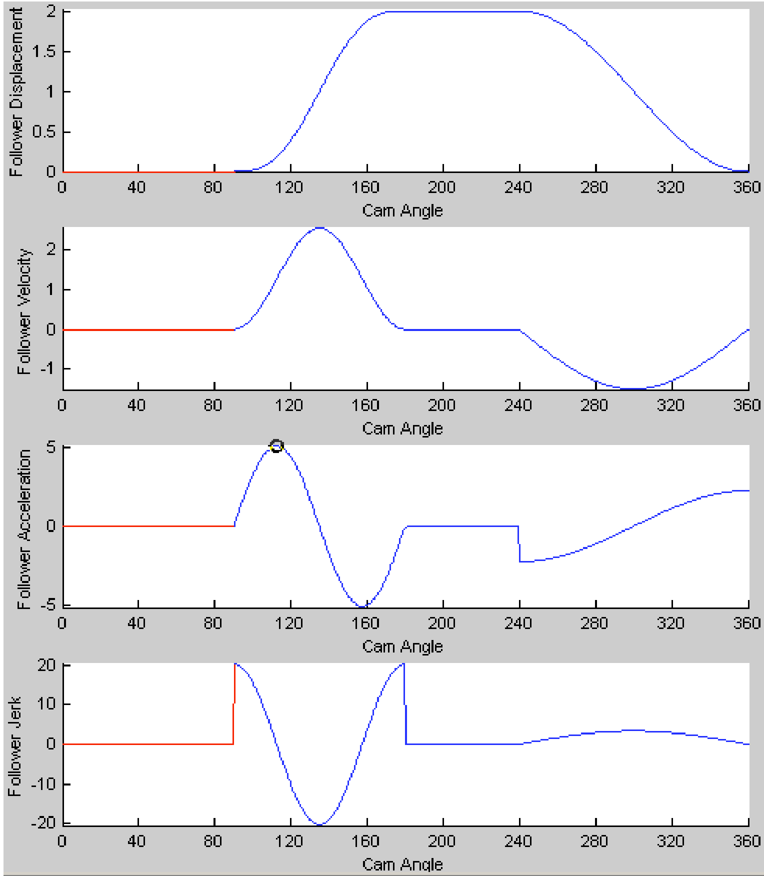


Fig. 3.11: Motion Plots for Radial Roller Example

Fig. 3.12: Optimization Inputs for Radial Roller Example

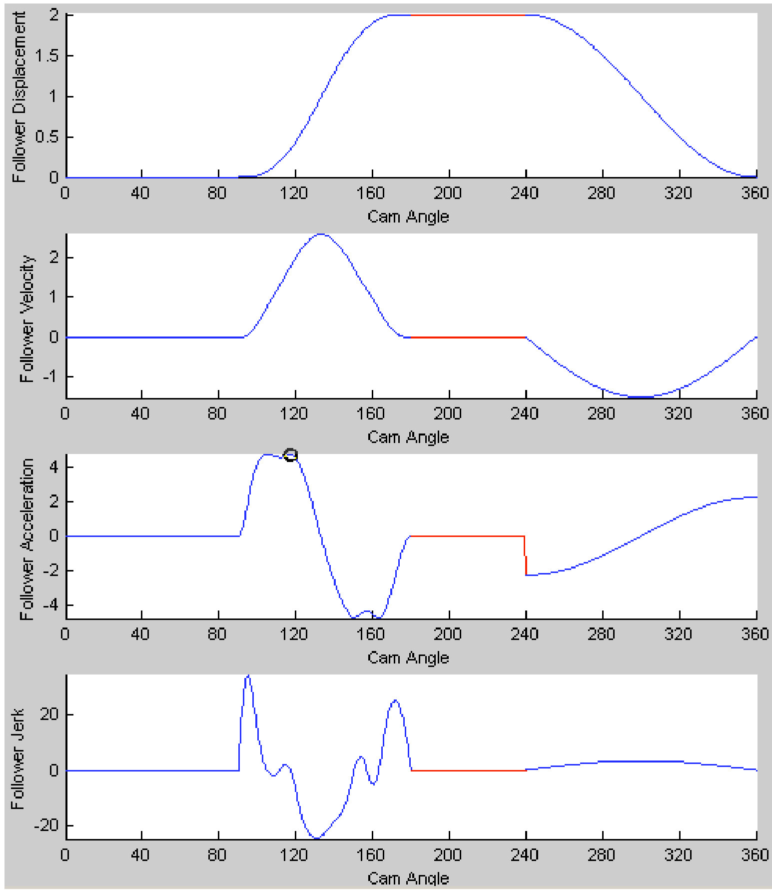


Fig. 3.13: Optimized Motion Plots for Radial Roller Example

4.0 Program for Rigid Body Guidance

4.1 Introduction

In rigid-body guidance or motion generation, the coupler of a linkage is guided through a series of positions. The programs in this set address three-positions, and three different programs are available as shown in Fig. 4.1

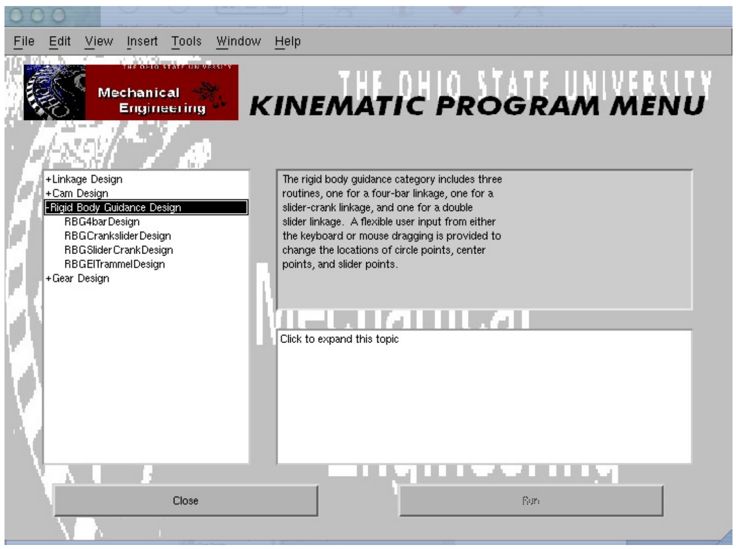


Fig. 4.1: Programs available for rigid-body guidance

The first program (RBG4barDesign) is for the design of four-bar linkages. The second (RBGCrankSliderDesign) is for crank-rockers and rocker-crank linkages, and the third (RBGElTrammelDesign) is for double-slider or elliptic-trammel linkages. Each of these will be discussed separately in the following. The basic programs follow the procedures and use the nomenclature in Section 6.3 of the textbook. Therefore, the basic design procedure will not be discussed here. However, the programs RBG4barDesign and RBGCrankSliderDesign both include a rectification feature that is not covered in the textbook. This will be discussed briefly when the topic is covered to describe the use of the programs.

4.2 Rigid-Body Guidance Using a Four-Bar Linkage (RBG4barDesign)

This routine is used for the design of four-bar linkages with either center points or circle points as input. The user can specify three coupler positions and angles. The pole locations between each two coupler positions are calculated and shown as a '+' marker in the graph. The circle of sliders corresponding to the three image poles for position 1 is shown in black. After all of the input data are identified, the linkage can be animated to determine if the linkage moves through all of the positions identified.

The program is structured in two windows. The first window is the design window where all of the input data are identified. The second window is the animation window where the linkage can be verified.

4.2.1 Design Window for Four-Bar Linkage for Rigid-Body Guidance

The design window is shown in Fig. 4.2. In the design window, frames are utilized to group three types of geometry, the center points, circle points, and coupler positions. Editable boxes for user input of the three coupler positions are provided. The user can either input the positions numerically, or move the locations and angles of the three coupler positions by mouse dragging. The GUI implementation also allows users to drag any circle or center point continuously with its coordinates updated dynamically. To be able to recognize corresponding points on the plot and data in the editable boxes, three different colors (red, blue, green) are used for the coupler positions.

As the circle and center points are moved using the mouse, the linkage will change shape. The current link lengths along with the Grashof type are continuously updated. When a Grashof type 2 linkage is indicated, the number "2" is printed in red.

4.2.1.1 Visual Aid To Identify Limits for Center Points

In Fig. 4.2, a red polygon is shown made up of dashed lines. This is provided as a visual aid to the user if there are locations where center points are or are not permitted. The coordinates of the corners of this polygon are provided in the editable boxes below the picture. The user may change any of the points. This polygon is for visual purposes only. It has no direct affect on the equations used in the design procedure.

4.2.1.2 Rectification

As indicated in Section 6.3.6, when linkages are designed using the basic procedure outlined in Section 6.3 of the textbook, it is common to find that they do not guide the rigid body through all three positions unless the assembly mode is changed. In such cases, when the linkage is animated, the rigid body will pass through 1 or 2 positions in one assembly mode and 2 or 1 positions in the other assembly mode. When this happens, the linkage design is unacceptable. This problem was referred to in the textbook as a change of branch. Considerable research has been devoted to identifying at the beginning of the design process linkages that do not have the change in branch. Waldron and his student have developed a relatively simple procedure that has been implemented in the four-bar linkage and slider crank-mechanism routines [1-4][*]. The procedure identifies acceptable regions for locating the two circle points under most circumstances, and that procedure has been implemented in the programs in this section. .

Avoiding the branch problem is a two-step process, and the regions in the two steps are different. In the first step, the circle point for the <u>driven</u> crank is chosen. In the design window, the driving crank is shown in green, and the driven crank is shown in black in order to distinguish between the two. The three image pole circles define acceptable locations for the circle point for the driven crank. The distances between successive image poles define the diameters of these three circles. There are three image poles (P_{12}, P_{13}, and P'_{23}), and these are the same points used to draw the circle of sliders in position 1. The unacceptable positions for the driven circle point are shown shaded in yellow. If the circle point for the driven crank is chosen in the yellow shaded area, the linkage will have a branch problem and be unacceptable. The user can

[*] References are given in Section 4.6 at the end of the chapter.

experiment with this by choosing linkages with the driven circle point in the yellow region and then animating the result. This can be done easily by dragging the driven circle point with the mouse. Regardless of where the driver circle point is chosen, the linkage will have a branch problem.

After the driven circle point is chosen, the driver circle point can be identified. However, even if the driven circle point is chosen outside of the yellow shaded region, it is possible to choose the location of the driver circle point such that branching will occur. In the design window, colored linear regions are shown radiating from the driven circle point. The driver circle point must be chosen to lie outside of the colored regions. Sometimes the acceptable region is very small or even nonexistent. If there is no linear region that is free of color, then there is no solution that will be free of the branch problem. In the example shown in Fig. 4.2, the linkage chosen has the driver circle point in the forbidden region. Therefore, that linkage will have a branch problem as will be apparent when the linkage is animated.

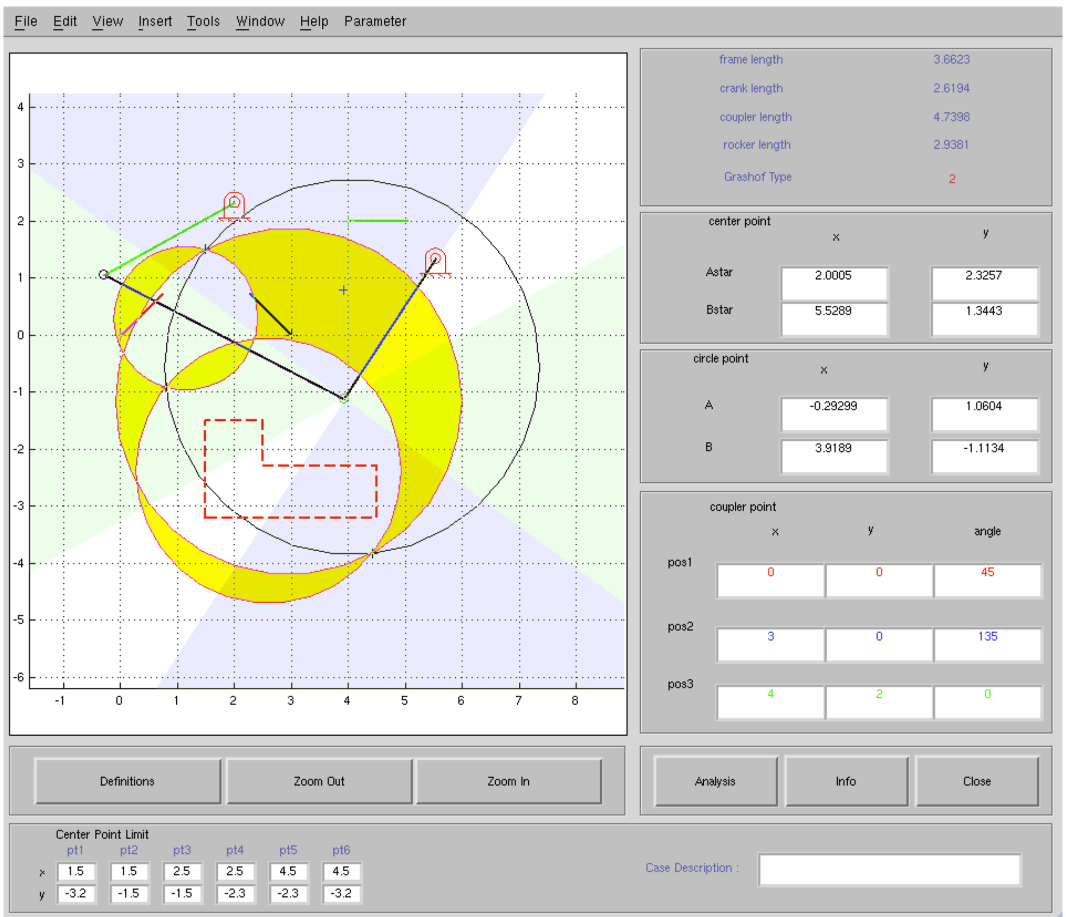


Fig. 4.2: The design window for the four bar linkage design for the rigid body guidance

4.2.2 Analysis Window for Four-Bar Linkage for Rigid-Body Guidance

Those three coupler positions in the analysis window (Fig. 4.3) use the same colors as in the design window. It is therefore possible to identify which position is missed if the rectification procedure is not applied. In the analysis window, 1 or 2 plots can be shown and animated. The plotting options are the animations for the two assembly modes. By animating both assembly modes, the effect of branching can be illustrated. For example, in assembly mode –1 in Fig. 4.3, the coupler moves through position 2, and in assembly mode 1, the coupler moves through positions 1 and 3.

Figures 4.4 and 4.5 are the design and analysis windows for a linkage chosen to avoid the branching problem. When this linkage is animated, assembly mode 1 goes through all of the positions.

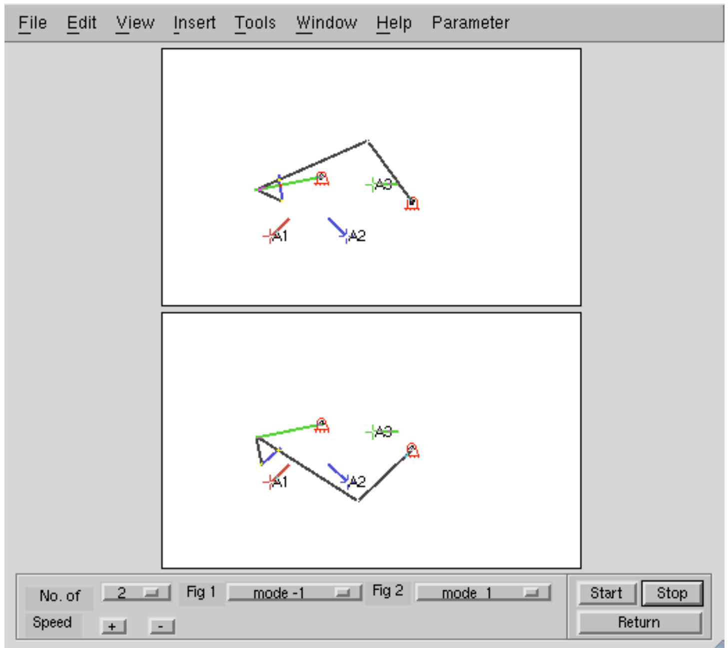


Fig. 4.3: The analysis window for the four bar linkage program for rigid body guidance

4.2.3 Zooming Feature in Analysis Window

To zoom in or out in the analysis window, the mouse is used. To zoom in, draw a marquee around the area that is to fill the window and click the mouse. This is shown in Fig. 4.6. The local window will redraw in a zoomed view as shown in Fig. 4.7. To zoom out and return to the original view, click the right mouse button.

An alternative way to activate the zoom feature is to place the mouse cursor on the figure and click the mouse. The program will zoom relative to the point where the cursor is located. A left mouse click will zoom in and a right mouse click will zoom out.

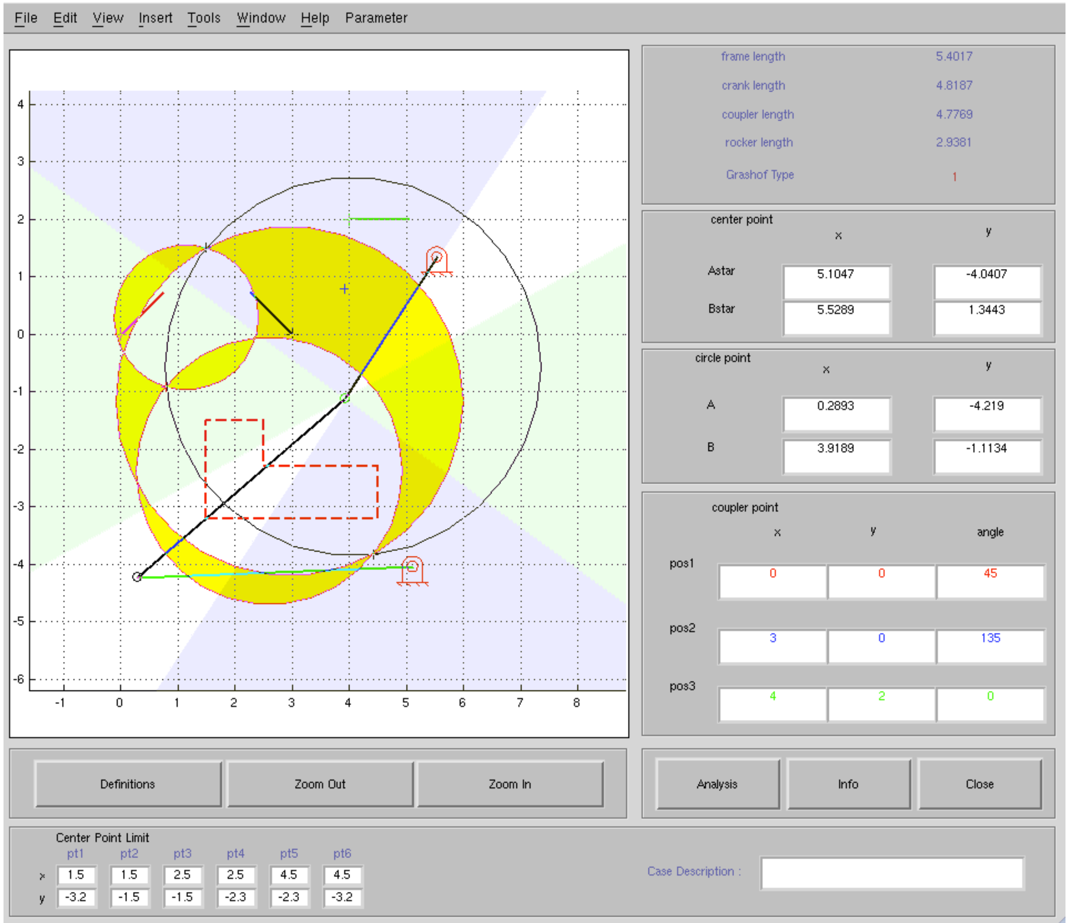


Fig. 4.4: Selection of driver circle point in acceptable region

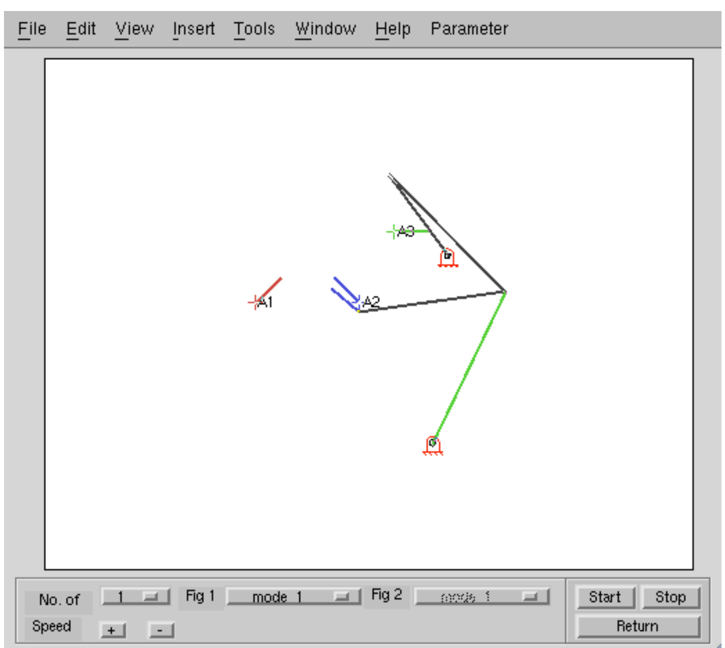


Fig. 4.5: Analysis window showing that the linkage goes through all positions

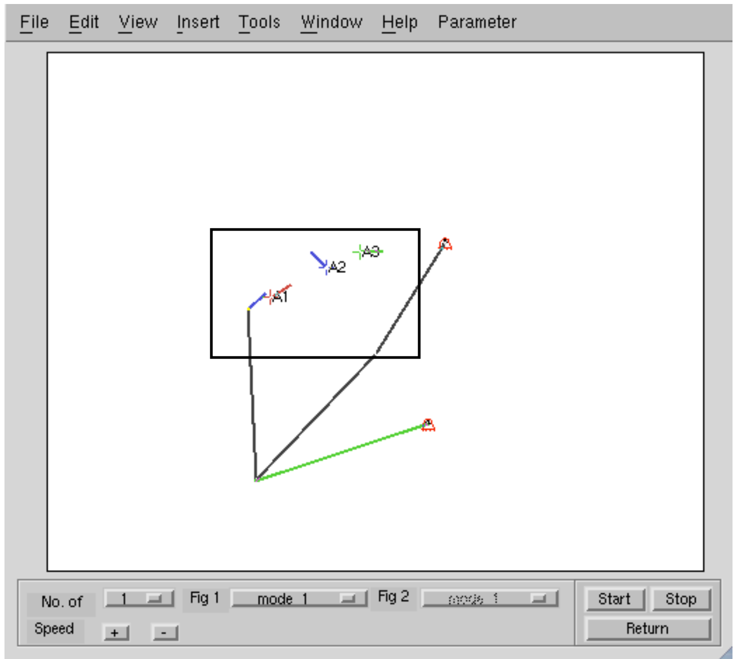


Fig. 4.6: Selecting the zoom region

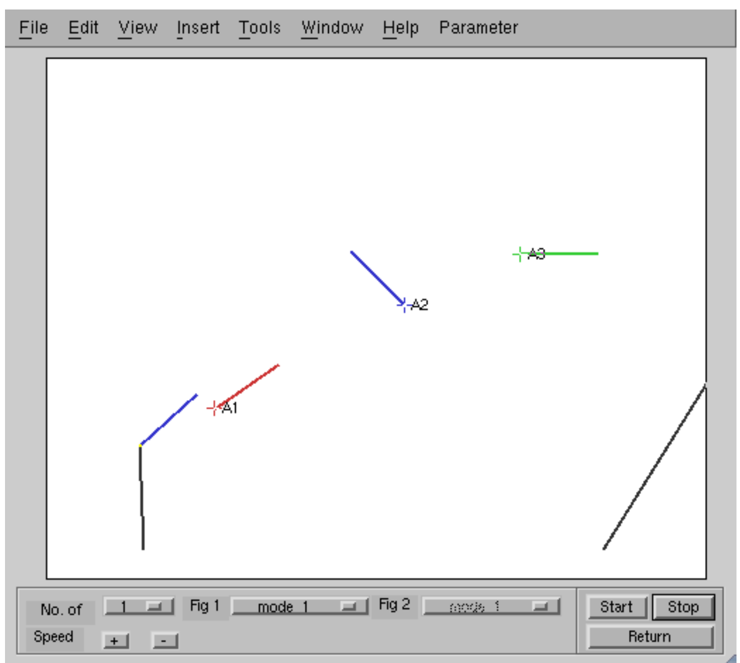


Fig. 4.7: Figure after zoom option

4.3 Rigid-Body Guidance Using a Crank-Slider Linkage (RBGCrankSliderDesign)

This routine is used for the design of slider-crank linkages with the crank as the driver. The center point, circle point, and slider point are the inputs. The user can specify three coupler positions and angles. The pole locations between each two coupler positions are calculated and shown as a '+' marker in the graph. The circle of sliders corresponding to the three image poles for position 1 is shown in black. After all of the input data are provided, the linkage can be animated to determine if it moves through all of the positions identified.

The program is structured in two windows. The first window is the design window where all of the input data are identified. The second window is the animation window where the linkage can be verified.

The need to distinguish between a crank slider and a slider crank is due to the rectification process. The forbidden regions for the slider point and the circle point for the crank are different for the two cases, and the two cases are treated in separate subprograms.

The design window for the case when the crank is the driver is shown in Fig. 4.8. In the design window, frames are utilized to group four types of geometry, the center points, circle points, slider point, and coupler positions. Editable boxes for the user input of three coupler positions are provided. The user can either input the positions numerically, or move the locations and angles of the three coupler positions by mouse dragging. The GUI implementation also allows users to drag any circle, center, or slider point continuously with its coordinates updated dynamically. To be able to recognize corresponding points on the plot and data in the editable boxes, three different colors (red, blue, green) are used for the coupler positions.

The slider point can be input either through the mouse by dragging the mouse cursor around the circle of sliders of though the input boxes. The slider points must lie exactly on the circle of sliders. Therefore, the program will correct any user numerical input to force the point to the nearest point on the circle. To do this, the program identifies a straight line from the point to the center of the circle and finds the nearest intersection of that straight line with the circle. The value input by the user is designated by "entered coord." and the value of the corresponding slider point is designated by "slider coord."

As in the case of a four-bar linkage, it is common to find that a slider-crank linkage designed using the basic procedure outlined in Section 6.3.6 of the textbook does not function as assumed. It is common to find that they do not guide the rigid body through all three positions unless the assembly mode is changed. In such cases, when the linkage is animated, the rigid body will pass through 1 or 2 positions in one assembly mode and 2 or 1 positions in the other assembly mode. When this happens, the linkage design is unacceptable. This problem was referred to in the textbook as a change of branch. The slider-crank program uses a procedure developed by Waldron to identify linkages that are unacceptable in the initial stages of the design. The procedure is similar to that used for four-bar linkages.

4.3.1 Rectification When the Crank Is the Driver

Avoiding the branch problem is a two-step process, and the regions in the two steps are different. The slider point is considered first because it is the <u>driven</u> link. Since the slider point is really just a circle point with the corresponding center point at infinity, the slider point can be chosen in the same way that the driven circle point was chosen in the case of the four-bar linkage. The main restriction is that only points on the circle of sliders can be chosen. The three image pole circles identify acceptable locations for the slider point. The distances between successive image poles define the diameters of these three circles. There are three image poles (P_{12}, P_{13}, and P'_{23}), and these are the same points used to draw the circle of sliders in position 1. The unacceptable positions for the driven slider point are shown shaded in yellow in the program. If the slider point is chosen in the yellow shaded area, the linkage will have a branch problem and be unacceptable.

After the driven slider point is chosen, the driver circle point can be identified. However, even if the driven slider point is chosen outside of the yellow shaded region, it is possible to choose the location of the driver circle point such that branching will still occur. In the design window, colored linear regions are shown radiating from the driven slider point. The driver circle point must be chosen to lie outside of the colored regions. Sometimes the acceptable region is very small or even nonexistent. If there is no linear region that is free of color, then there is no solution that will be free of the branch problem. In the example shown in Fig. 4.8, the linkage chosen has the driver circle point outside the forbidden region. Therefore, that linkage will not have a branch problem.

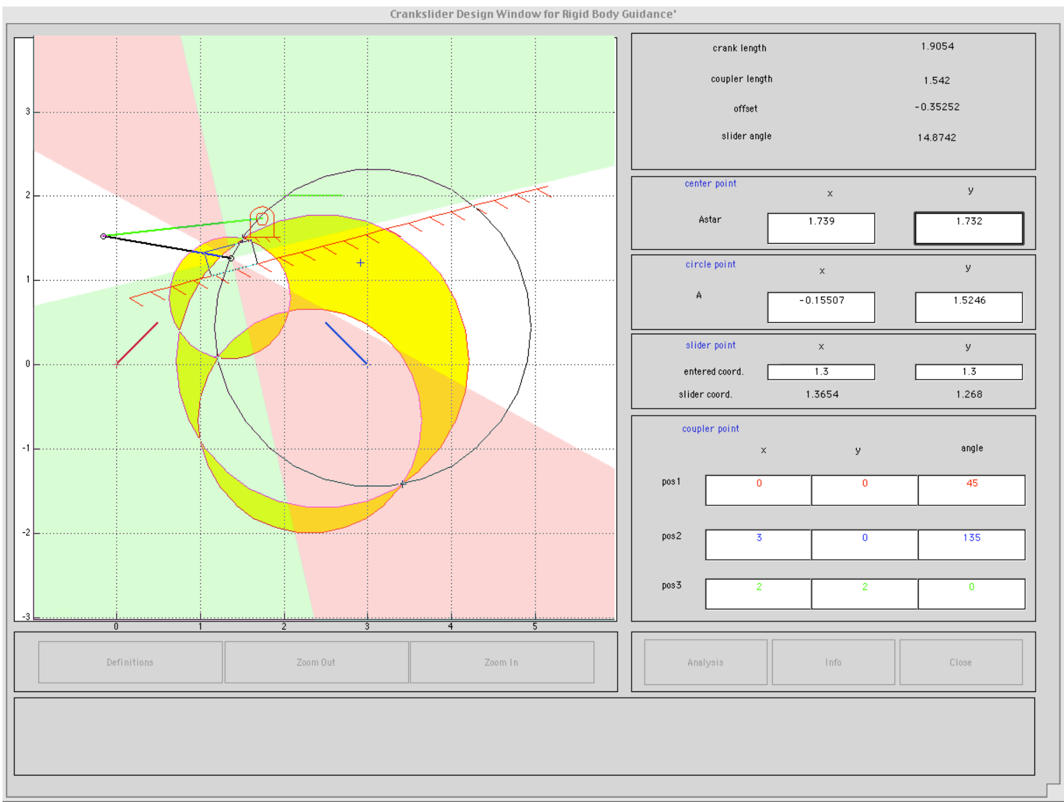


Fig. 4.8: The design window for the crank-slider design program (crank driving)

4.3.2 Analysis Window When the Crank Is the Driver

The three coupler positions in the analysis window (Fig. 4.9) use the same colors as in the design window. It is therefore possible to identify which position is missed if the rectification procedure is not applied. In the analysis window, 1 or 2 plots can be shown and animated. The plotting options are the animations for the two assembly modes. By animating both assembly modes, the effect of branching can be illustrated. In this example, the design was based on the rectification procedure, and the coupler passes through all of the positions.

The zooming feature is also available in the RBGCrankSliderDesign program. To zoom in, locate a rectangular marquee around the figure, and click the left mouse button. To zoom out, click the right mouse button. Alternatively, simply locate the cursor about the new center of the figure and click the left mouse button to zoom out and the right mouse button to zoom in.

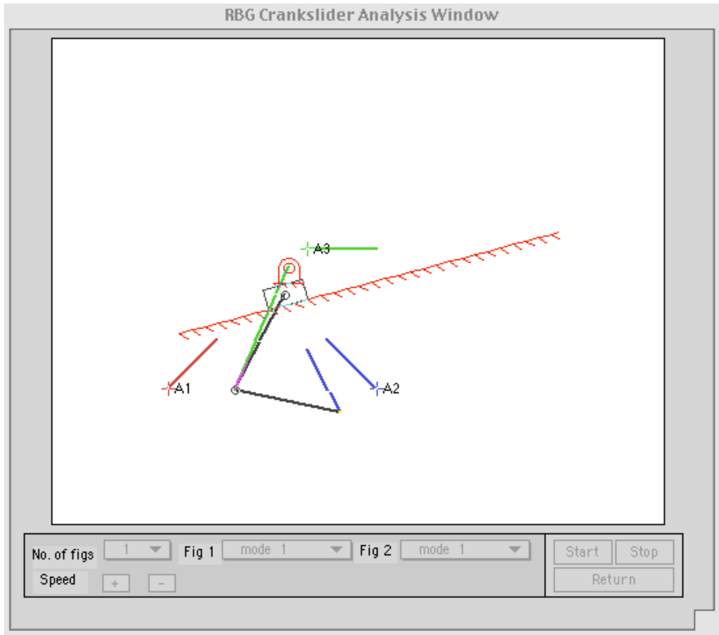


Fig. 4.9: The analysis window for the example in Fig. 4.8 (crank driving)

4.4 Rigid-Body Guidance Using a Slider-Crank Linkage (RBGSliderCrankDesign)

This routine is used for the design of slider-crank linkages with the slider as the driver. Again, the center point, circle point, and slider point are the inputs, and the user can specify three coupler positions and angles. The pole locations between each two coupler positions are calculated and shown as a '+' marker in the graph. The circle of sliders corresponding to the three image poles for position 1 is shown in black. After all of the input data are provided, the linkage can be animated to determine if it moves through all of the positions identified.

Figure 4.10 contains the same input information as Fig. 4.8, but now the slider is the driver and the crank is the driven link. Note that some of the rectification regions have changed. The only difference between the design procedures for a slider crank and crank slider is in the rectification process.

4.4.1 Rectification When the Slider Is the Driver

If the slider is the driver, the crank is the driven link so that the crank circle point is considered first. The three image pole circles define acceptable locations for the circle point. Note that the locations of these image pole triangles are the same regardless of which link is chosen as the driver. The unacceptable positions for the driven circle point are shown shaded in yellow. If the circle point is chosen in the yellow shaded area, the linkage will have a branch problem and be unacceptable regardless of where the slider point is chosen.

After the driven circle point is chosen, the driver slider point can be identified. In the design window, colored linear regions are shown radiating from the driven circle point. The slider point must be chosen to lie on the parts of the circle of sliders that are outside of the colored regions. Again the acceptable regions may be very small or even nonexistent. If there is no linear region that is free of color, then there is no solution that will be free of the branch problem. In the example shown in Fig. 4.10, the linkage chosen has the driver slider point inside the forbidden (colored) region. Therefore, that linkage will have a branch problem. This illustrates the importance of identifying the actual driver since if the crank is the driver, there will be no branch problem as illustrated in the example from Section 4.3. As indicated in Fig. 4.10, only a small part of the circle of sliders is in the acceptable region. We have chosen a different set of circle and slider points in Fig. 4.11. The resulting linkage does not have a branch problem.

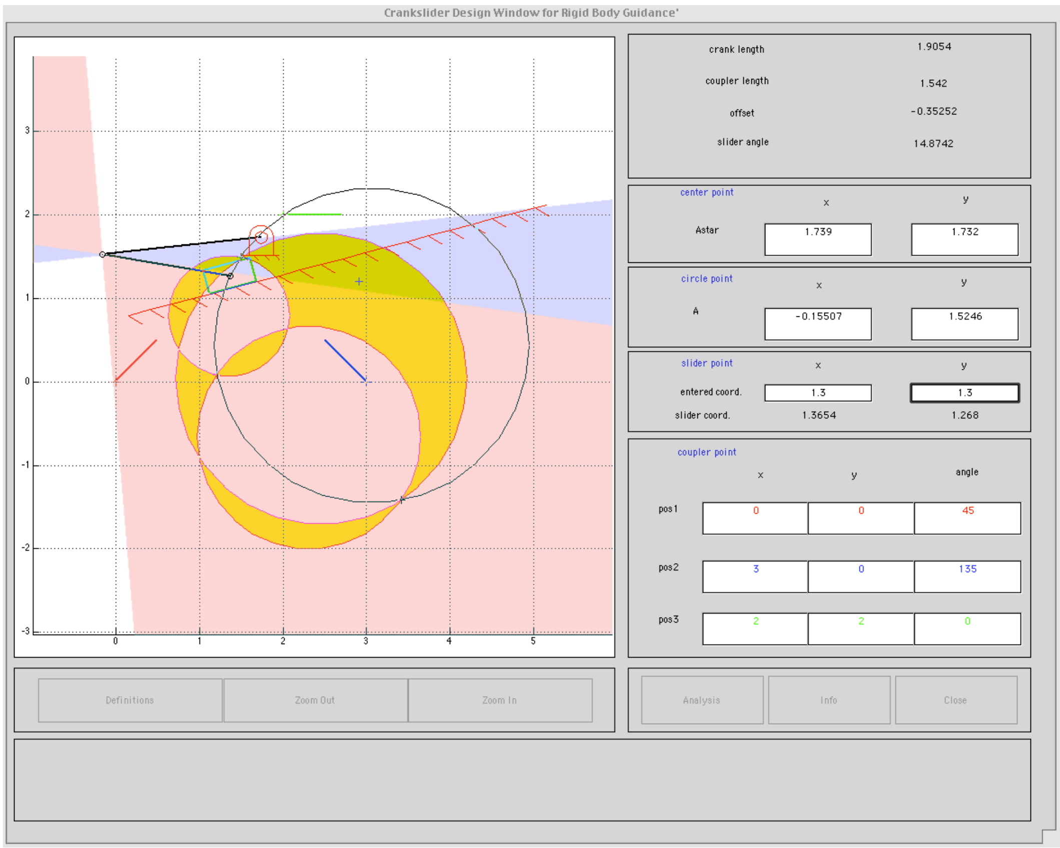


Fig. 4.10: The design window for the slider-crank design program (slider driving)

4.4.2 Analysis Window When the Slider is the Driver

The three coupler positions in the analysis window (Fig. 4.12) use the same colors as in the design window. It is therefore possible to identify which position is missed if the rectification procedure is not applied. In the analysis window, 1 or 2 plots can be shown and animated. The plotting options are the animations for the two assembly modes. By animating both assembly modes, the effect of branching can be illustrated. In this example, the design was based on the rectification procedure, and the coupler passes through all of the positions.

The zooming feature is also available in the RBGSliderCrankDesign program. To zoom in, locate a rectangular marquee around the figure, and click the left mouse button. To zoom out, click the left mouse button.

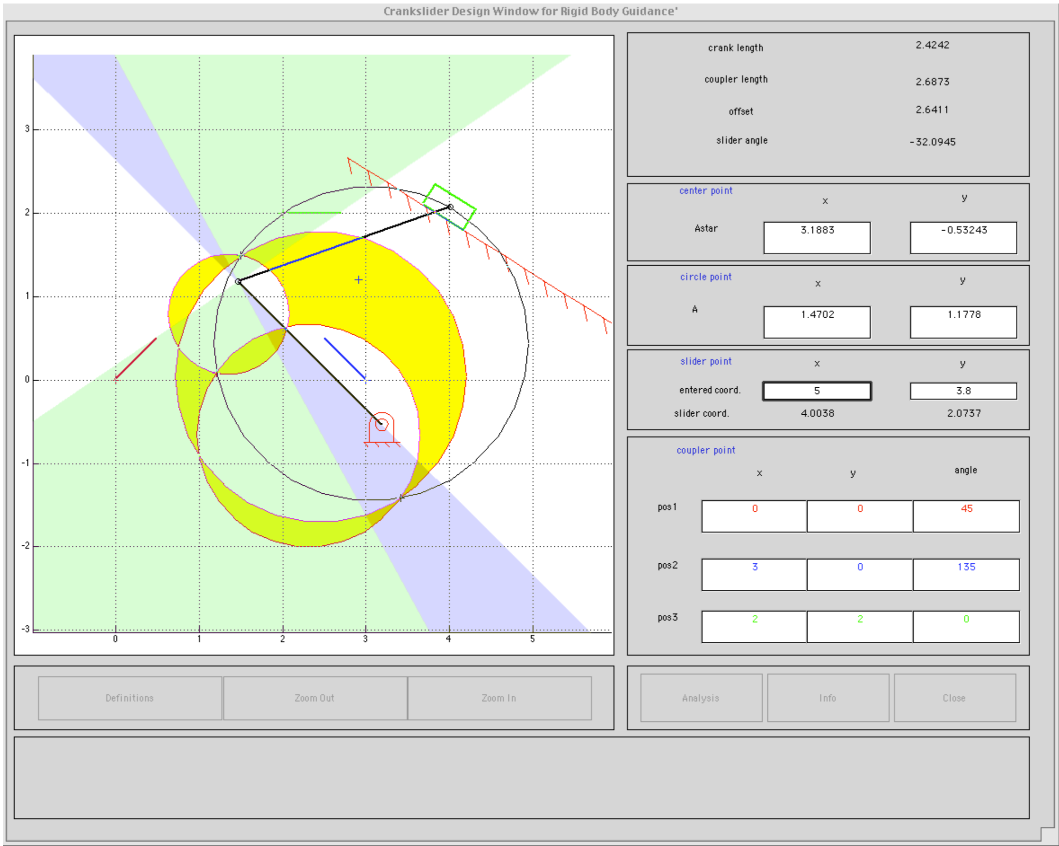


Fig. 4.11: Selection of the slider and circle points that will give no branch problem

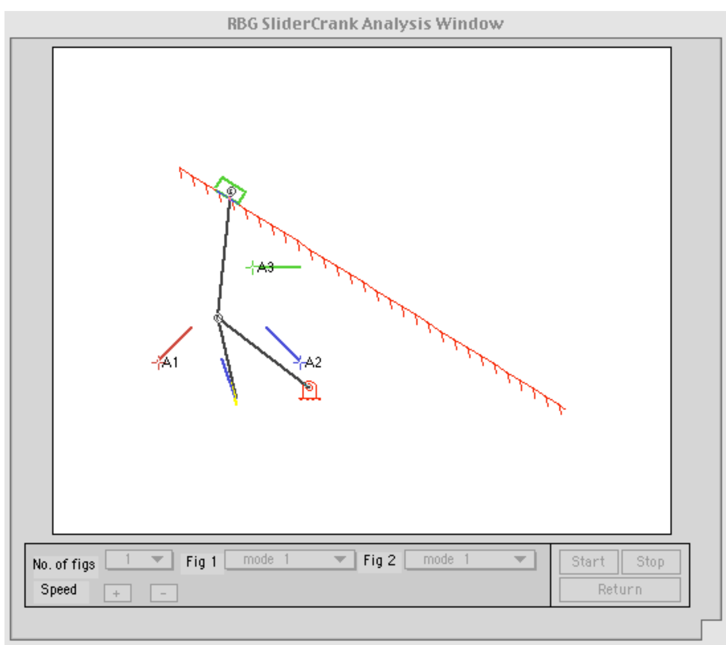


Fig. 4.12: The analysis window when the when the slider is the driver

4.5 Rigid-Body Guidance Using an Elliptic Trammel Linkage (RBGElTrammelDesign)

This routine is used for the design of a double slider mechanism or elliptic trammel for rigid body guidance. For this design, two slider points must be chosen on the circle of sliders.

The program is structured in two windows. The first window is the design window where all of the input data are identified. The second window is the animation window where the linkage can be verified.

The design process does not depend on which slider is chosen as the input. Once the slider points are chosen, only one assembly mode is possible. Therefore, rectification is not an issue.

4.5.1 Design Window for Elliptic Trammel Linkage

The design window for the elliptic trammel is shown in Fig. 4.13. In the design window, frames are utilized to group three types of geometry, the slider points and lines and coupler positions. Editable boxes for user input of three coupler positions are provided. The user can either input the positions numerically, or move the locations and angles of the three coupler positions by mouse dragging. The GUI implementation also allows users to drag any slider point continuously with its coordinates updated dynamically. To be able to recognize corresponding points on the plot and data in the editable boxes, three different colors (red, blue, green) are used for the coupler positions.

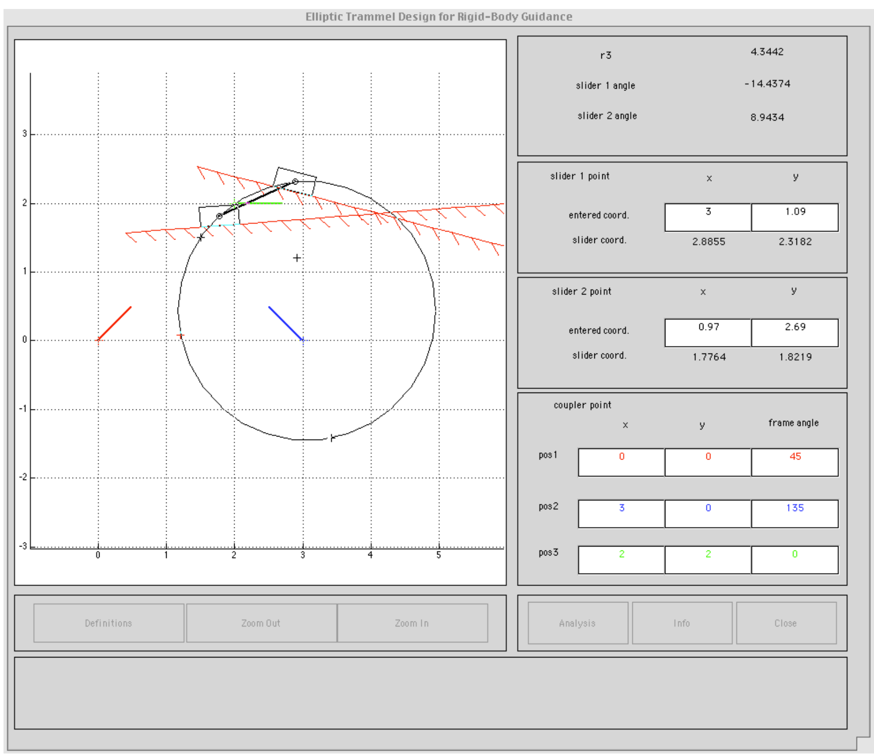


Fig. 4.13: The design window for the elliptic trammel routine

4.5.2 Analysis Window for Elliptic Trammel Mechanism

The three coupler positions in the analysis window (Fig. 4.14) use the same colors as in the design window. In the analysis window, only one plot is shown and animated because only one assembly position is possible.

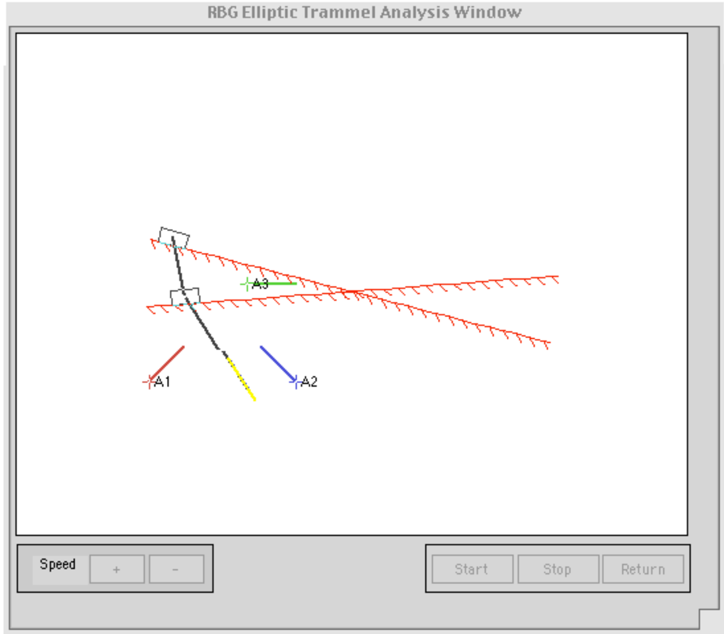


Fig. 4.14: The analysis window for the elliptic trammel routine

4.6 Situations When Rectification Procedure Fails

As indicated earlier, in most circumstances, the rectification procedure implemented in the programs identifies the regions that will give unacceptable linkages. However, after using the programs, the observant student will notice that the four-bar and slider-crank programs will sometimes identify linkages that will not move through all of the positions. The problem that is not addressed is the "circuit defect" [5]. This occurs, when the two assembly modes of linkage are separated for all positions of the driver link. When this happens, it will be obvious from the animation that the linkage cannot move through the range of motion identified without disassembly. To resolve the problem, choose a different set of circle, center, of slider points.

4.7 References

1. Waldron, K. J., "Range of Joint Rotation in Planar Four-Bar Synthesis for Finitely Separated Positions: Part I – The Multiple Branch Problem," *ASME Paper* No. 74–DET-108, Mechanisms Conference, New York, 1974.

2. Waldron, K. J., "Solution Rectification in Three Position Motion Generation Synthesis," Department of Mechanical Engineering, The Ohio State University, pp. 301-306.

3. Chuang, J. C., Strong, R. T. and Waldron, K. J., "Implementation of Solution Rectification Techniques in an Interactive Linkage Synthesis Program," *Journal of Mechanical Design, Trans. ASME*, Vol. 103, 1981, pp. 657-664.

4. Waldron, K. J., "Graphical Solution of the Branch and Order Problems of Linkage Synthesis for Multiply Separated Positions," *Journal of Engineering for Industry, Trans. ASME*, Series B, Vol. 99, 1977, pp. 591 – 597.

5. Mirth, J.A. and Chase, T., "Circuits and Branches of Single-Degree-of-Freedom Planar Linkages", *ASME Journal of Mechanical Design,* vol. 115, no. 2, pp. 223-230, 1993.

5.0 Program for Displaying Gears

5.1 Introduction

The gear group includes two programs as shown in Fig. 5.1. The first program (Arb2thDesign) determines the tooth form that is conjugate to a straight-sided tooth. The second program (GeardrAnalysis) draws a gear tooth given the geometry of the generating rack. Each of the programs will be discussed separately.

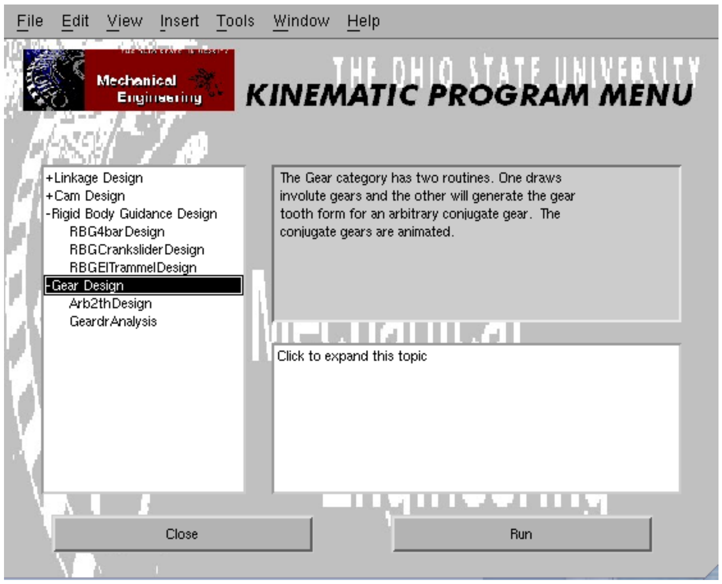


Fig. 5.1: Programs available under gear design

5.2 Arb2thDesign Program

This program displays two windows, a design window where the input data are identified, and an analysis window. The analysis window is described first.

5.2.1 Design Window for Arb2thDesign Program

The procedure used to generate the tooth profile is given in Appendix B of this manual. The design window is shown in Fig. 5.2. The generated tooth form is displayed in the graphics window, and the generating tooth information is shown in the frame to the right of the figure. Geometric information on the generating tooth form is shown if the *Definition* button is selected. This is shown in Fig. 5.3. For the generating gear, the user may input the following:

1) Number of teeth
2) Number of points describing the generating tooth. The more points, the higher the accuracy of the generated tooth.
3) Addendum constant, a_3
4) Flank angle for generating tooth (see Fig. 5.3)

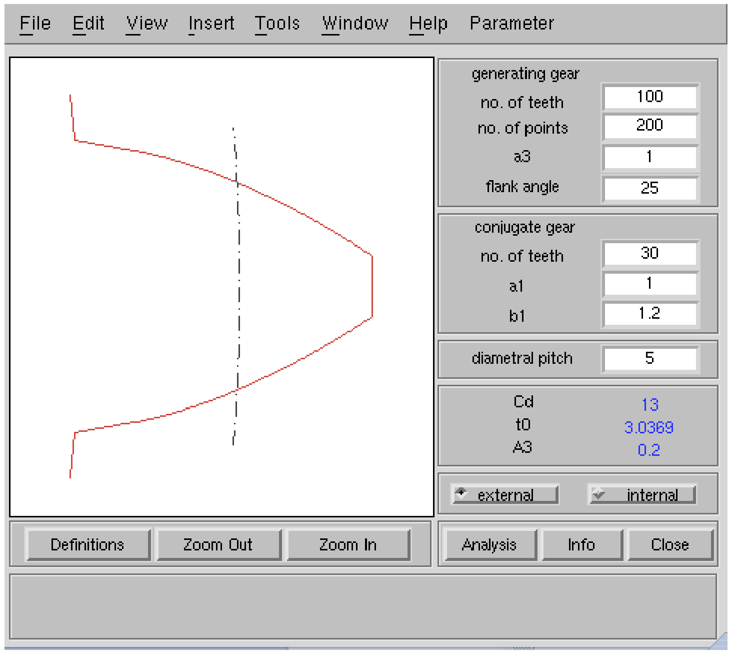


Fig. 5.2: Design window for Arb2thDesign program

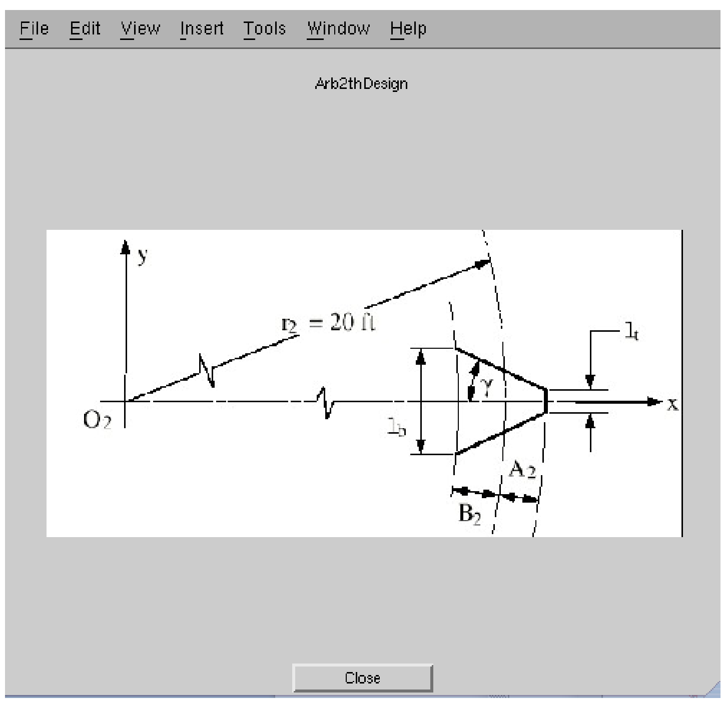


Fig. 5.3: The definitions window for the Arb2thDesign program

The generated gear information is summarized in the frame below that for the generating gear. For the generated gear, the user may input the following

1) Number of teeth
2) Addendum constant, a_1
3) Dedendum constant, b_1

The final piece of information that the user may input is the diametral pitch. Once the input data are established, the following information is displayed:

1) Center distance, C_d
2) Tooth thickness at the pitch circle, t_0
3) Addendum length on generating gear, A_3

Buttons are provided for an internal or external gear; however, currently the program can generate only external gears. To see the entire gears, select the analysis button.

5.2.2 Analysis Window for Arb2thDesign Program

The analysis window is shown in Fig. 5.4. In the analysis window, 1, 2, 3, or 4 plots can be displayed. Options for the plots are an animated view of the gear teeth, the tooth form for the generating tooth form, and the entire gear for both the generating gear and generated gear. In the individual windows, the gears are plotted as large as possible, so they are not normally plotted to the same scale.

The buttons for starting, stopping, and changing the speed of the animation is the same as in the previous programs. By changing the display to a single figure as is shown in Fig. 5.5, it is possible to observer the details of tooth meshing as the teeth come into and leave contact.

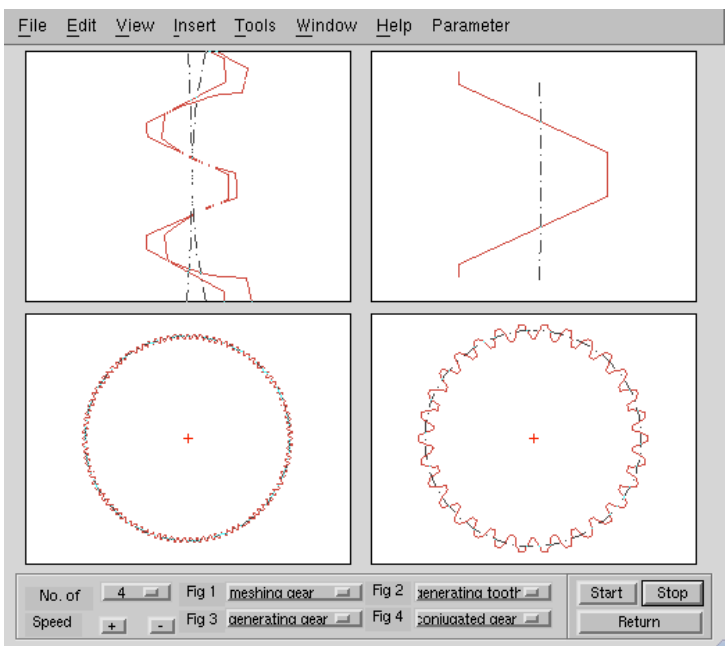


Fig. 5.4: Analysis window for the Arb2thDesign program

5.3 GeardrAnalysis Program

This program will generate an involute tooth form given the geometry of the generating rack, and the equations given in Section 10.12 are programmed. The program uses two windows, a design window where the input data are identified, and an analysis window. The analysis window is described first.

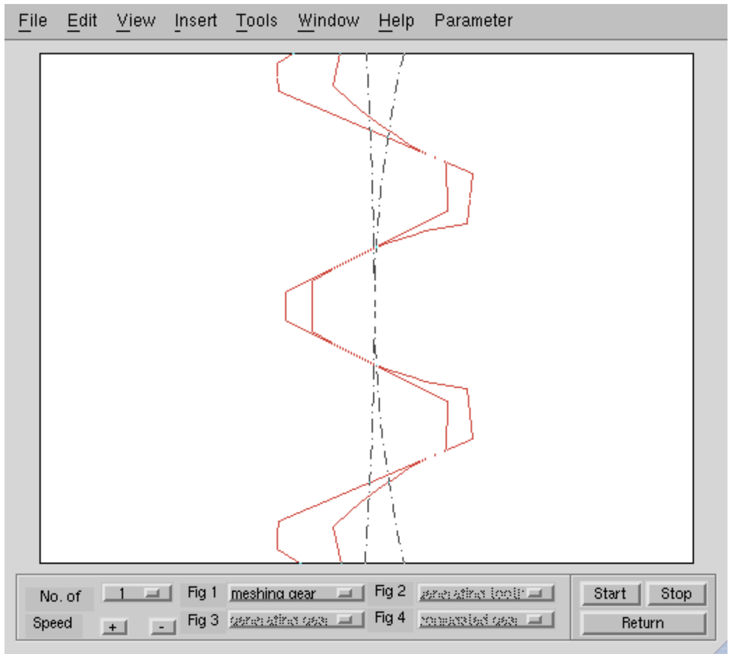


Fig. 5.5: Closeup of gear mesh when one figure only is displayed

5.3.1 Design Window for GeardrAnalysis Program

The design window is shown in Fig. 5.6. Half of the generating hob or rack tooth form is displayed in the graphics window, and the generating tooth information is shown in the frame to the right of the figure.

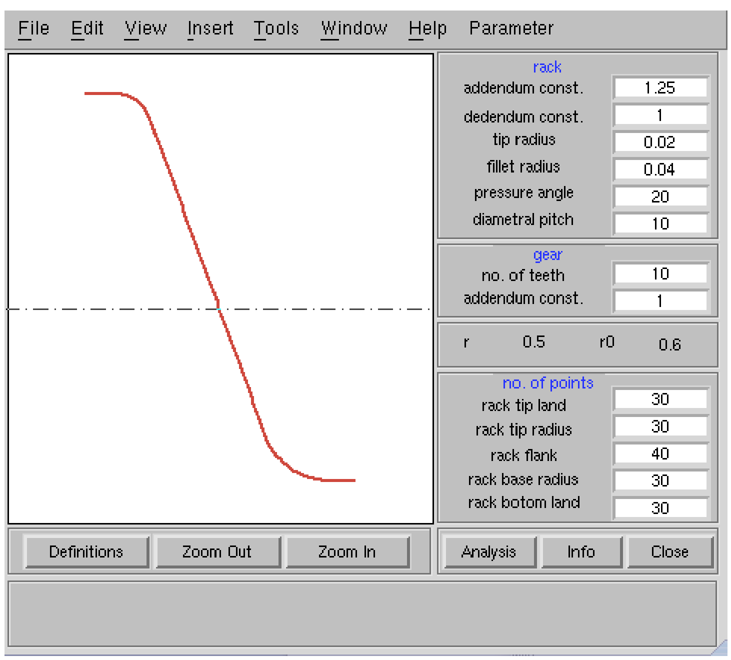


Fig. 5.6: Design window for GeardrAnalysis program

Geometric information on the generated tooth form is shown if the *Definition* button is selected.

This is shown in Fig. 5.7.

For the generating hob or rack, the user may input the following:

1) Addendum constant for rack
2) Dedendum constant for rack
3) Radius of tip of rack tooth
4) Radius of fillet of rack tooth
5) Pressure angle in degrees

6) Diametral pitch for the generated gear, the user may input the following:

1) Number of teeth
2) Addendum constant

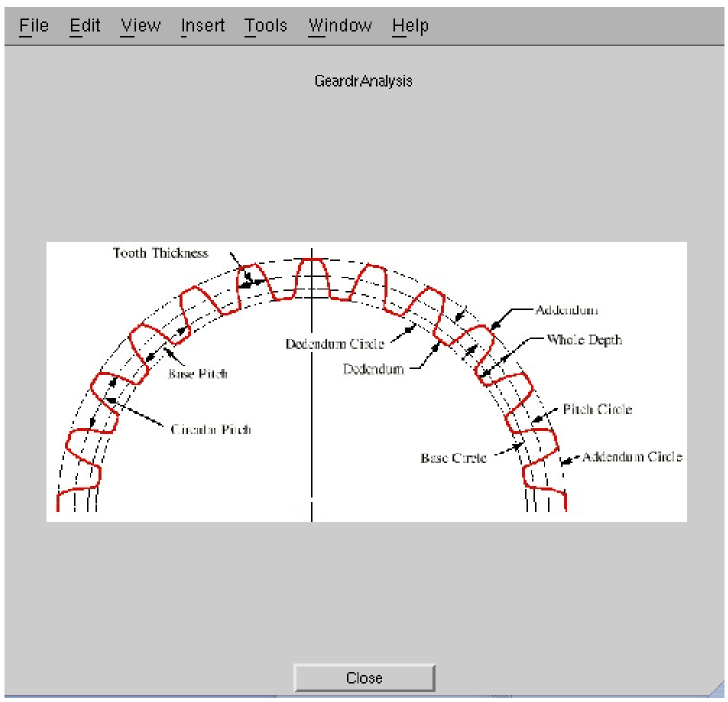


Fig. 5.7: The definitions window for the GeardrAnalysis program

The gear tooth coordinates are generated numerically. Therefore, it is necessary to identify the number of points in the different regions of the hob. Five regions are identified, and the user may input the number of points in each region. The accuracy generally increases with the number of points.

1) Number of points in region of rack tip land
2) Number of points in region of rack tip radius
3) Number of points in region of rack flank
4) Number of points in region of rack base radius
5) Number of points in region of rack bottom land

After the input data are identified, the generated tooth form and gear is shown in the analysis window.

5.3.2 Analysis Window for GeardrAnalysis Program

The analysis window is shown in Fig. 5.8. In the analysis window, 1 or 2 plots can be displayed. Options for the plots are the generated gear tooth form and the entire gear. By displaying one figure with the gear only, it is possible to show visually how undercutting appears on the gear.

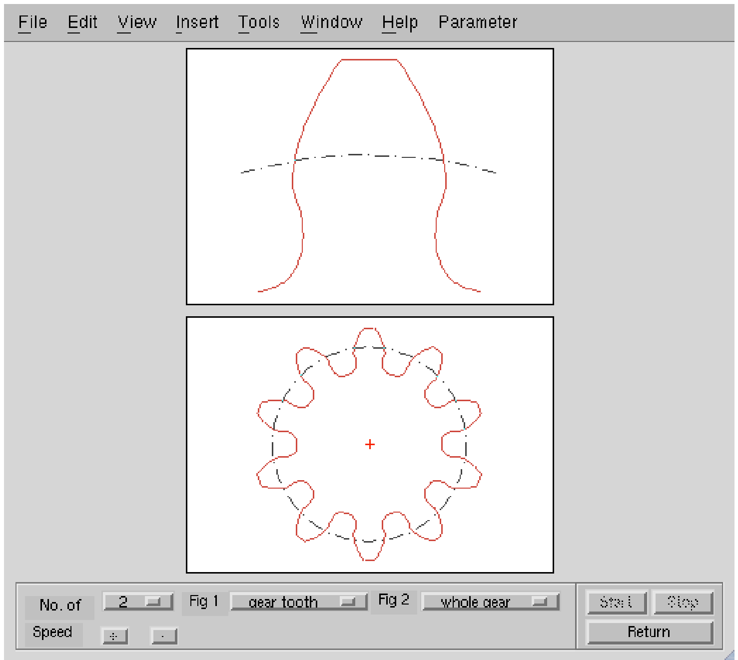


Fig. 5.8: Analysis window for GeardrAnalysis program

Appendix A

Procedure for Euler-Savary Equation

A.1 Introduction

The information in this Appendix was originally contained in the main textbook; however, it was removed because of limited space. The entire development is given here although the MATLAB programs apply to only part of what is presented.

Another way of generating a point path with desired properties is to use curvature theory. This provides a way of precisely controlling the trajectory in one position of a lamina. For example, the direction and curvature of the path of a given point can be controlled in a given position. The expectation is that the path will retain a similar curvature at all positions near to the designated point.

Curvature theory is actually closely related to the theory of motion generation through a series of finitely separated positions. It can be thought of as the limiting case in which the design positions become infinitesimally separated. There are many similarities. For example, as was shown in Section 4.2.5, the points in a lamina that lie on a straight line in three specified positions of that lamina lie on a circle. The corresponding result when the positions become infinitesimally separated is that at any instant in the motion of a lamina, the points whose paths have inflections, that is the points whose paths are locally straight, lie on a circle, called the inflection circle. The inflection circle passes through the instantaneous center and is tangent to the same line as the fixed and moving centrodes, which are the loci of the successive positions of the instantaneous center relative to the fixed and moving reference frames. The pole triangle collapses into the instantaneous center.

A.2 Two Infinitesimally Separated Positions

Specifying two design positions infinitesimally separated from one another is equivalent to specifying a position of a lamina and the velocity state of the lamina as it moves through that position. The velocity state can be specified by specifying the velocity $\boldsymbol{v}_O$ of the point O in the moving lamina that is instantaneously coincident with the origin, together with the angular velocity $\boldsymbol{\omega}$ of the lamina. The velocity of any other point A is then given by

$$\boldsymbol{v}_A = \boldsymbol{v}_O + \boldsymbol{v}_{A/O} = \boldsymbol{v}_O + \omega \times \boldsymbol{r}_{A/O} \tag{A.1}$$

where $\mathbf{r}_{A/O}$ is the directed line $\overline{OA}$.

Let us choose any point, C, in the moving lamina as a circle point. We seek a crank, with circle point at C such that the path of C produced by that crank is tangent to the path of C required by the velocity state. That is, the circular path of C produced by the crank should have the velocity vector $\boldsymbol{v}_C$ tangent to it. Clearly, any point on the normal to $\boldsymbol{v}_C$ through C can serve as the center point C*.

Example A.1 (Synthesis of Linkage for Specified Velocity of Point in Coupler)

Problem:

Synthesize a four-bar linkage to give the coupler point at the origin a velocity of one unit per second in the *X* direction when the angular velocity is 4 rad/sec counter-clockwise.

Solution:

Let the four-bar linkage be defined in the usual manner with link 2 as the driver and link 3 as the coupler. From the problem statement, point O_3 is the coupler point at the origin (coordinates relative to the frame are 0,0). In the following, the subscript 3 will be dropped because it is understood that all points being considered are in the coupler.

$$\boldsymbol{v}_O = 1\mathbf{i},\ \boldsymbol{\omega} = 4\boldsymbol{k} \text{ rad/sec}$$

For the four-bar linkage, we need to select two circle points, and for this we will choose points C(1, 1), and D(2, 0). Then,

$$\boldsymbol{v}_C = \boldsymbol{v}_O + \boldsymbol{v}_{C/O} = \boldsymbol{v}_O + \boldsymbol{\omega}\times\boldsymbol{r}_{C/O} = 1\boldsymbol{i} + 4\boldsymbol{k}\times(\boldsymbol{i}+\boldsymbol{j}) = -3\boldsymbol{i} + 4\boldsymbol{j}$$

where $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are orthogonal unit vectors in the x, y and normal directions, respectively. Also,

$$\boldsymbol{v}_D = 1\,\boldsymbol{i} + 4\,\boldsymbol{k}\times(2\,\boldsymbol{i}) = \boldsymbol{i} + 8\,\boldsymbol{j}$$

Points C and D and velocities $\boldsymbol{v}_C$ and $\boldsymbol{v}_D$ are plotted on Fig. A.1. The normals to $\boldsymbol{v}_C$ and $\boldsymbol{v}_D$ at those points were drawn and C* and D* were selected on those normals. The resulting linkage is C*CD D*. Compare this procedure to that used for two finitely separated positions.

Note that this linkage will give a different velocity state for each value of angular velocity for the coupler. Therefore, an infinite number of velocity states are possible.

The instant center for the coupler is shown in Fig. A.2. Notice that C*CI and D D*I are collinear. This corresponds to the result that a crank subtends angle $\theta_{12}/2$ at the pole P_{12}. As θ_{12} approaches zero, the pole becomes co-linear with the circle and center points, and becomes the instantaneous center of rotation, I.

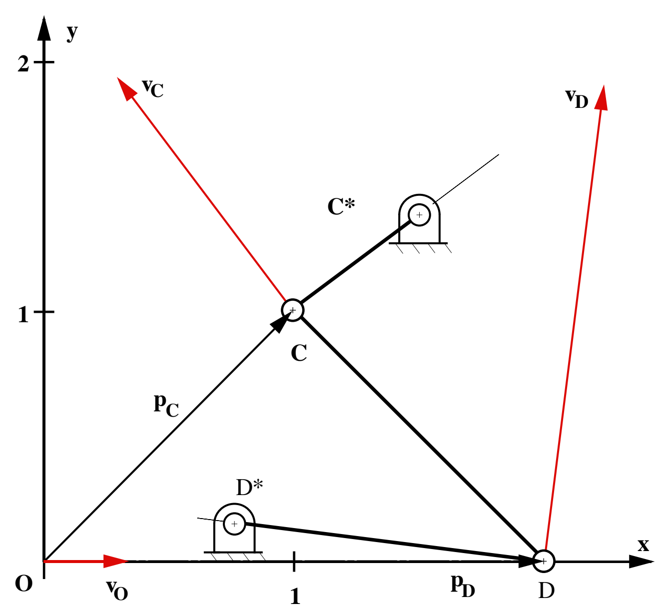


Fig. A.1: The solution of Example 4.6.

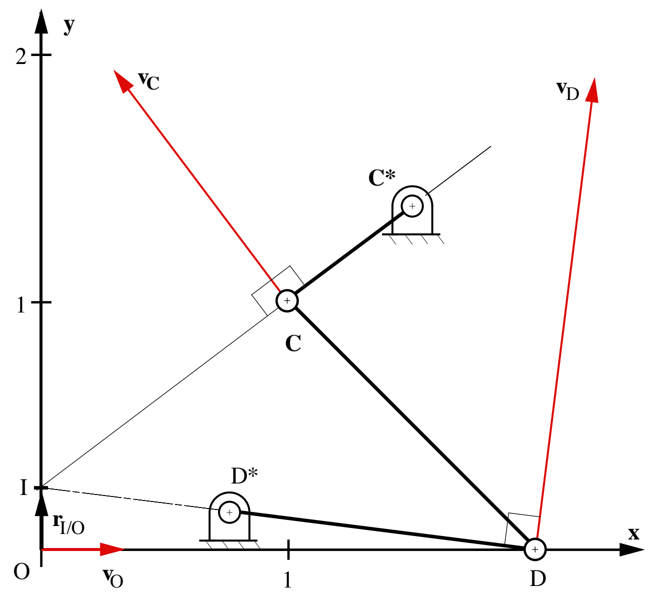


Fig. A.2: The location of the instantaneous center for the velocity field of Example 4.6.

In the case of two finitely separated positions, we found that it was also possible to move the moving lamina through the two design positions using only a single pivot between the fixed and moving planes as shown in Fig. 4.14. This point was the pole for those two positions. Correspondingly, the required velocity state can be generated by means of a single pivot at the instantaneous center of the motion. The location, $\boldsymbol{r}_I$, of the instantaneous center relative to the origin is obtained from Eq. (A.1) by letting $\boldsymbol{v}_C = 0$ to get

$$0 = \boldsymbol{v}_O + \boldsymbol{\omega} \times \boldsymbol{r}_{I/O}$$

or

$$0 = \boldsymbol{\omega} \times \boldsymbol{v}_O + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r}_{I/O}) = \boldsymbol{\omega} \times \boldsymbol{v}_O - \omega^2\, \boldsymbol{r}_{I/O}$$

or

$$\boldsymbol{r}_{I/O} = \frac{\omega \times \boldsymbol{v}_O}{\omega^2} = \frac{\boldsymbol{k} \times \boldsymbol{v}_O}{\omega} \tag{A.2}$$

where $\boldsymbol{\omega} = \omega\boldsymbol{k}$, requiring that ω be positive counter-clockwise.

In Example A.1 above,

$$\boldsymbol{r}_{I/O} = \frac{\boldsymbol{k} \times (1\boldsymbol{i})}{4} = \frac{\boldsymbol{j}}{4}.$$

This is shown in Fig. A.2.

Notice that from Eq. (H.2), whatever solution is used

$$\frac{v_O}{\omega} = |\boldsymbol{r}_{I/O}|$$

is constant. That is, regardless of the angular velocity, the ratio of the velocity of the point at the origin to that angular velocity in the design position is constant. Put another way

$$\frac{d\boldsymbol{p}_O}{d\theta}$$

is constant, where $\boldsymbol{p}_O$ is the position vector from the origin of the coordinate system to the coupler point O which has coordinates (momentarily) of (0, 0). It is convenient to say that we are specifying the velocity state of the moving body, but it is more precise to say that we are specifying the derivative of the position of a point on the coupler with respect to the coupler angle.

A.3 Three Infinitesimally Separated Positions

A.3.1 Center of Curvature of Path of Moving Point Relative to Frame

Specifying three infinitesimally separated design positions is equivalent to specifying a position of the moving lamina and its velocity and acceleration states in that position. In addition to the velocity of the point in the moving lamina coincident with the origin and the angular velocity, we must specify the acceleration $\boldsymbol{a}_O$ of the point at the origin and the angular acceleration, $\boldsymbol{\alpha}$ of the moving lamina. The acceleration of any point, A, in the moving lamina can then be found

$$\boldsymbol{a}_A = \boldsymbol{a}_O + \boldsymbol{a}_{A/O} = \boldsymbol{a}_O + \alpha \times \boldsymbol{r}_{A/O} + \omega \times (\omega \times \boldsymbol{r}_{A/O}) = \boldsymbol{a}_O + \alpha \times \boldsymbol{r}_{A/O} - \omega^2 \boldsymbol{r}_{A/O} \tag{A.3}$$

Given the velocity and acceleration states of the moving lamina, we can find the radius of curvature of the path of any point in the moving lamina as that lamina passes through the design position. This is done by resolving the acceleration of that point into components tangent to, and normal to its path. Let $\boldsymbol{n}$ be a unit vector normal to the path that A traces on the frame and let $\mathbf{t}$ be a unit vector tangent to that path. We know that the velocity of the point will be tangent to the path that the point traces on the frame. Therefore, $\boldsymbol{t}$ is in the $\boldsymbol{v}_A$ direction, and $\boldsymbol{n}$ is directed so the $\boldsymbol{k} \times \boldsymbol{t} = \boldsymbol{n}$. Then the acceleration of point A can be written as:

$$\boldsymbol{a}_A = \boldsymbol{a}_A^t + \boldsymbol{a}_A^n = a_A^t \boldsymbol{t} + a_A^n \boldsymbol{n}$$

When the acceleration is expressed in terms of the normal and tangential components, it is the normal component which is a function of velocity and geometry. An expression for this component was derived in Section 3.3.2 when coincident points were considered. In particular, the acceleration of A can be rewritten as

$$\boldsymbol{a}_A = a_A^t \boldsymbol{t} + \frac{v_A^2}{\rho} \boldsymbol{n} \tag{A.4}$$

where ρ is the radius of curvature of the path that the point A traces on the frame. Equation (A.4) is derived in most undergraduate engineering mechanics texts, and a detailed derivation is given by Hall[1].

If we take the dot product of $\boldsymbol{n}$ with each side of Eq. (A.4), we get

$$\boldsymbol{n} \bullet \boldsymbol{a}_A = a_A^n = \frac{v_A^2}{\rho}$$

or

$$\rho = \frac{v_A^2}{\boldsymbol{n} \bullet \boldsymbol{a}_A} = \frac{v_A^2}{a_A^n} \tag{A.5}$$

---

[1]Hall, A.S., *Kinematics and Linkage Design*. Balt Publishers, West Lafayette, IN (1961).

Equation (A.5) allows us to locate the center of curvature of the path of any point in a linkage once the basic velocity and acceleration analyses have been completed. The center of curvature of the path is in the direction of the normal component of acceleration. In Eq. (A.5), the normal component of acceleration can be plus or minus. If it is plus, it is in the ***+n*** direction, and if it is minus, it is in the -***n*** direction.

Example A.2 (Center of Curvature of the Path that a Point on the Coupler of a Slider-Crank Mechanism Traces on the Frame)

Problem:

Identify a procedure whereby we can locate the center of curvature of the path traced on the frame by points on the coupler of a slider-crank mechanism.

Solution:

Consider the slider-crank mechanism in Fig. A.3, and assume that the path of C_3 is of interest. The center of curvature of the path is a purely geometric quantity, and therefore, the actual values used for the velocity and acceleration analysis are arbitrary. Also, the choice of the driver is arbitrary.

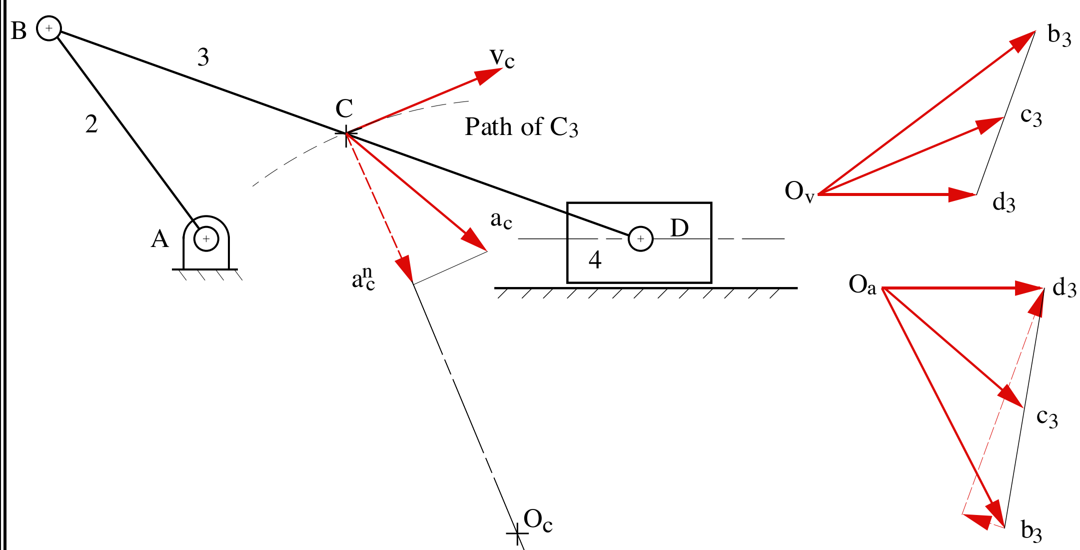


Fig. A.3: Center of curvature of path of C_3 on the frame.

Because the velocity of C_3 is tangent to the path that C_3 traces on link 1, the velocity vector for C_3 indicates the direction of the tangent to the path. The center of curvature for the path will be on a line through C and normal to the velocity of C_3. From the acceleration analysis, we can determine ${}^1\boldsymbol{a}_{C_3}$ and resolve the acceleration into two components which are in the direction of ${}^1\boldsymbol{v}_{C_3}$ (tangent) and normal to ${}^1\boldsymbol{v}_{C_3}$. Then,

$$\boldsymbol{a}_{C3} = \boldsymbol{a}_{C3}^{n} + \boldsymbol{a}_{C3}^{t}$$

The radius of curvature of the path is calculated by using the magnitudes of the velocity and normal acceleration in the following:

$$r_{C/O_C} = \frac{v_{C3}^2}{a_{C3}^n}$$

The location of O_C is along the normal vector in the direction of $\boldsymbol{a}_{C3}^n$. This is shown in Fig. A.3.

A.3.2 Synthesis Using the Center of Curvature at a Point and Along a Path

To synthesize a linkage to move a lamina through three infinitesimally separated positions, we can take any point in that lamina, find the direction of its path and the radius of curvature of that path, and hence the center of curvature of the path. By locating the center point C* at the center of curvature, we get a crank which gives the required path direction and path curvature in the design position. Repeating this procedure for a second crank we generate a four-bar linkage which gives the required velocity and acceleration states while passing through the design position.

Example A.3 (Synthesis of a four-bar linkage for three infinitesimally separated positions of a point in the coupler)

Problem:

The velocity state of a lamina is to be as in Example A.2. That is, $\boldsymbol{v}_O = 1$ in/s in the x direction, $\boldsymbol{\omega} =$ 4 rad/s counter-clockwise. In addition $\boldsymbol{a}_O$ is to be 20 in/s^2 in the y direction, and $\boldsymbol{\alpha}$ is to be 10 rad/s^2 clockwise.

Solution:

Choose C at position (1, 1) and D at (2, 0) as before, then $\boldsymbol{v}_C = -3\boldsymbol{i} + 4\boldsymbol{j}$ in/s and $\boldsymbol{v}_D = \boldsymbol{i} + 8\boldsymbol{j}$ in/s. From the problem statement, $\boldsymbol{\alpha} = -10\boldsymbol{k}$ rad/s^2 and $\boldsymbol{a}_o = 10\boldsymbol{j}$. Therefore, applying Eq. (A.3) gives:

$$\boldsymbol{a}_C = 10\boldsymbol{j} + (-10)\boldsymbol{k} \times (\boldsymbol{i} + \boldsymbol{j}) - 16(\boldsymbol{i} + \boldsymbol{j}) = -6\boldsymbol{i} - 16\boldsymbol{j}$$

At point C in the coupler,

$$\boldsymbol{t} = \frac{-3\boldsymbol{i} + 4\boldsymbol{j}}{\sqrt{3^2 + 4^2}} = -\frac{3}{5}\boldsymbol{i} + \frac{4}{5}\boldsymbol{j}$$

and

$$\boldsymbol{n} = \boldsymbol{k} \times \boldsymbol{t} = -\frac{3}{5}\boldsymbol{j} - \frac{4}{5}\boldsymbol{i}$$

so, applying Eq. (A.5)

$$\rho_C = \frac{3^2 + 4^2}{\left(-\frac{4}{5}\boldsymbol{i} - \frac{3}{5}\boldsymbol{j}\right) \bullet (-6\boldsymbol{i} - 16\boldsymbol{j})} = 1.736$$

Applying Eq. (A.3):

$$\boldsymbol{a}_D = 10\boldsymbol{j} + (-10)\boldsymbol{k} \times 2\boldsymbol{i} - 16\,(2\boldsymbol{i}) = -32\boldsymbol{i} - 10\boldsymbol{j}$$

At D

$$\boldsymbol{t} = \frac{\boldsymbol{i} + 8\boldsymbol{j}}{\sqrt{1^2 + 8^2}}$$

and

$$\boldsymbol{n} = \boldsymbol{k} \times \frac{1}{\sqrt{65}}(\boldsymbol{i} + 8\boldsymbol{j}) = \frac{1}{\sqrt{65}}(\boldsymbol{j} - 8\boldsymbol{i})$$

Applying Eq. (A.5):

$$\rho_D = \frac{1^2 + 8^2}{\frac{1}{\sqrt{65}}(\boldsymbol{j} - 8\boldsymbol{i}) \bullet (-32\boldsymbol{i} - 10\boldsymbol{j})} = 2.130$$

Using this data, C* and D* are located as shown in Fig. A.4. Note that a minus sign on either ρ_C or ρ_D would indicate that the center of curvature is located in the -$\boldsymbol{n}$ direction.

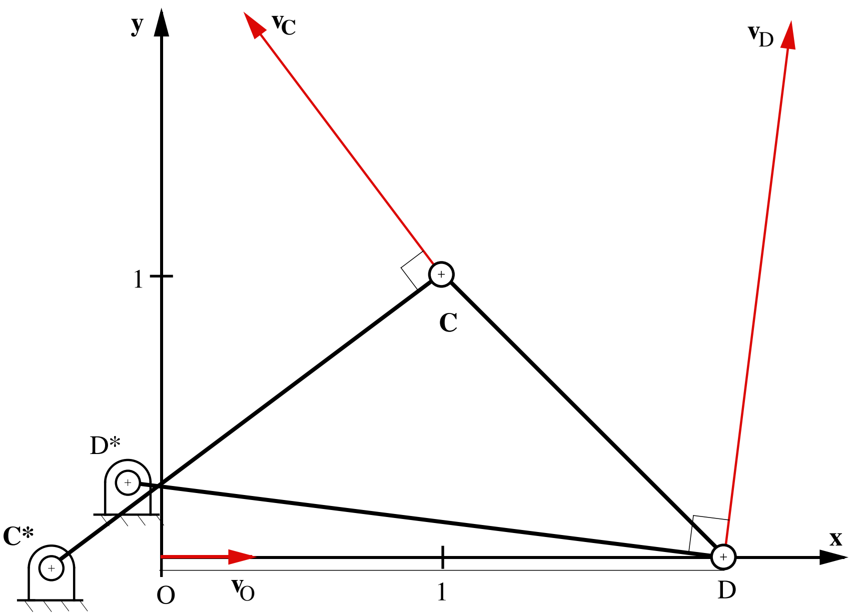


Figure A.4: The solution to Example A.3. C* and D* are selected to give the desired velocity and acceleration fields.

A.3.3 Inflection Circle

We found that, for three finitely separated positions, there are an infinite number of points whose three positions all lie on the same straight line and that they are distributed on a circle which passes through all three image poles. Let us seek the equivalent result for 3 infinitesimally separated positions: namely, the locus of points that, for a given velocity and acceleration state, have paths with locally infinite radius of curvature. Another way of stating this is the locus of points whose paths have points of inflexion at the instant of passing through the design position.

Looking at Eq. (A.5), we see that ρ approaching infinity implies $\boldsymbol{n} \bullet \boldsymbol{a}_A = 0$. In the general case, $\boldsymbol{a}_A$ will be nonzero. Then, since $\boldsymbol{n}$ is normal to $\boldsymbol{v}_A$, this implies that $\boldsymbol{v}_A$ and $\boldsymbol{a}_A$ have the same or opposite directions. Hence

$$\boldsymbol{v}_A \times \boldsymbol{a}_A = 0$$

Applying Eqs. (A.1) and (A.3)

$$(\boldsymbol{v}_O + \omega\, \boldsymbol{k} \times \boldsymbol{r}_{A/O}) \times (\boldsymbol{a}_O + \alpha\, \boldsymbol{k} \times \boldsymbol{r}_{A/O} - \omega^2 \boldsymbol{r}_{A/O}) = 0 \qquad \text{(A.6)}$$

For the analysis, we may select the origin of coordinates to be at any location that we like. It will simplify the results if we move the origin to the instantaneous center, I, between the moving body and the frame. Then $\boldsymbol{v}_O$ becomes zero, and $\boldsymbol{a}_O = \boldsymbol{a}_I$. That is, $\boldsymbol{a}_O$ becomes the acceleration of the point in the moving body which is at the instantaneous center. Equation (A.6) then becomes

$$(\omega\boldsymbol{k} \times \boldsymbol{r}_{A/I}) \times (\boldsymbol{a}_I + \alpha\, \boldsymbol{k} \times \boldsymbol{r}_{A/I} - \omega^2\boldsymbol{r}_{A/I}) = 0$$

or

$$(\boldsymbol{k} \times \boldsymbol{r}_{A/I}) \times \boldsymbol{a}_I - \omega^2\, (\boldsymbol{k} \times \boldsymbol{r}_{A/I}) \times \boldsymbol{r}_{A/I} = 0$$

or

$$(\boldsymbol{k} \times \boldsymbol{r}_{A/I}\,) \times \boldsymbol{a}_I + \omega^2(r_{A/I})^2\boldsymbol{k} = 0$$

Let the angle between $\boldsymbol{a}_I$ and $\boldsymbol{r}_{A/I}$ be γ_A (see Fig. A.5), where γ_A is measured from $\boldsymbol{a}_I$ to $\boldsymbol{r}_{A/I}$. Then

$$(\boldsymbol{k} \times \boldsymbol{r}_{A/I}) \times \boldsymbol{a}_I = -\, r_{A/I}a_I \sin\,(\gamma_A + \pi/2)\, \boldsymbol{k} \tag{A.7}$$

since $|\boldsymbol{k} \times \boldsymbol{r}_{A/I}\,| = r_{A/I}$ and the angle between $\boldsymbol{k} \times \boldsymbol{r}_{A/I}$ and $\boldsymbol{a}_I$ is $-(\gamma_A + \pi/2)$. Hence

$$\omega^2(r_{A/I})^2\boldsymbol{k} - a_I\, r_{A/I} \cos\gamma_A\, \boldsymbol{k} = 0$$

or

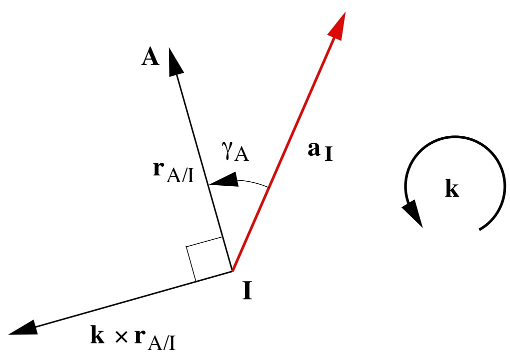


Fig. A.5: The geometry of the vectors in Eq. (A.7).

$$r_{A/I} = \frac{a_I}{\omega^2}\cos\gamma_A \tag{A.8}$$

This is the equation of a circle passing through I with diameter

$$D = \frac{a_I}{\omega^2}\,. \tag{A.9}$$

The center of the circle through I is located on a line from I and in the $\boldsymbol{a}_I$ direction. This circle is called the *inflection circle* and is represented in Fig. A.6. The inflection circle can be viewed as the limit of the image pole circle for three finitely separated positions as those positions become infinitesimally close. Just as the image pole circle (circle of sliders) is the locus of circle points whose three positions lie on a straight line, the inflection circle is the locus of points whose paths are locally straight.

We now seek an expression for the radius of curvature of the path of any point, A, in terms of the variables used in Eq. (A.7). Equation (A.5) gives

$$\rho = \frac{v_A^2}{\boldsymbol{n} \bullet \boldsymbol{a}_A}$$

Substituting from Eqs. (A.1) and (A.3) for $\boldsymbol{v}_A$ and $\boldsymbol{a}_A$ with origin at the instantaneous center

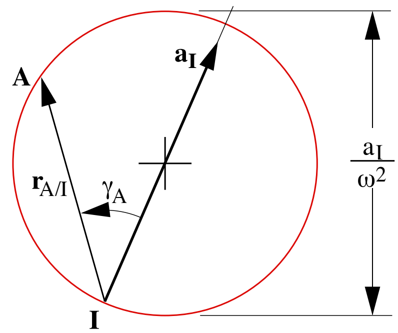


Fig. A.6: The inflection circle, for given *I*, a_I, *r*, ω, and γ_A.

$$\boldsymbol{v}_A = 0 + \omega \boldsymbol{k} \times \boldsymbol{r}_{A/I}, \ v_A{}^2 = \omega^2 \ r_{A/I}^2$$

Now

$$\mathbf{n} = \mathbf{k} \times \frac{\mathbf{v}_A}{v_A} = -\frac{\omega \mathbf{r}_{A/I}}{\omega r_{A/I}} = -\frac{\mathbf{r}_{A/I}}{r_{A/I}}$$

and from Eq. (A.3),

$$\mathbf{a}_A = \mathbf{a}_I + \alpha\, \mathbf{k} \times \mathbf{r}_{A/I} - \omega^2 \mathbf{r}_{A/I}$$

so

$$\boldsymbol{n} \bullet \boldsymbol{a}_A = -\frac{\boldsymbol{r}_{A/I}}{r_{A/I}} \bullet \boldsymbol{a}_I + 0 + \omega^2 r_{A/I}$$

Referring to Fig. A.6

$$\boldsymbol{r}_{A/I} \bullet \boldsymbol{a}_I = r_{A/I} a_I \cos\gamma_A$$

so

$$\boldsymbol{n} \bullet \boldsymbol{a}_A = -a_I \cos\gamma_A + \omega^2 r_{A/I}$$

and

$$\rho = \frac{\omega^2 r_{A/I}^2}{\omega^2 r_{A/I} - a_I \cos\gamma_A}$$

Now, if D is the diameter of the inflection circle, Eq. (A.9) gives

$$D = \frac{a_I}{\omega^2}$$

and so

$$\rho = \frac{r_{A/I}^2}{r_{A/I} - D\cos\gamma_A} \qquad \text{(A.10)}$$

Equation (A.10) is one form of the Euler-Savary equation. The Euler-Savary Equation is very useful because, given the instantaneous center and inflection circle, it can be used to locate the center point corresponding to any given circle point, or vice-versa. The inflection circle is readily constructed for a given four-bar linkage, and it is, therefore, more convenient to work with the inflection circle than with the variables ω and $\boldsymbol{a}_I$.

The geometric meaning of the Euler-Savary Equation is discernible by referring to Fig. A.7. Let A be the point whose path curvature is sought. If we use directed line segments, $r_{A/I}$ points from I to A, and $r_{A/A*}$ points from $A*$ to A. Also, $r_{JA/I} = D\cos\gamma_A$ where J_A is the location where a ray from I to A crosses the inflection circle. Hence, if $A*$ is the center of curvature of the path of point A, then $\rho = r_{A/A*}$ and

$$r_{A/A^*} = \frac{r_{A/I}^2}{r_{A/I} - r_{JA/I}}.$$

Now $r_{A/I} - r_{Q/I} = r_{A/Q}$ so

$$\frac{r_{A/A^*}}{r_{A/I}} = \frac{r_{A/I}}{r_{A/JA}} \tag{A.11}$$

that can be viewed as the geometric form of the Euler-Savary Eq. (A.10).

A.3.4 Different Forms for the Euler-Savary Equation

The Euler-Savary Equation can be expressed in several different ways, and the different forms are useful depending on the known quantities when a problem is formulated. For example, another form can be derived from Eq. (A.11) as follows:

$$r_{A/A^*} = r_{A/I} + r_{I/A^*} = \frac{r_{A/I}^2}{r_{A/JA}} = \frac{r_{A/I}^2}{r_{A/I} + r_{I/JA}}$$

or

$$(r_{A/I} + r_{I/A^*})(r_{A/I} + r_{I/JA}) = r_{A/I}^2 + (r_{A/I})(r_{I/JA}) + (r_{I/A^*})(r_{A/I}) + (r_{I/A^*})(r_{I/JA}) = r_{A/I}^2$$

Simplifying

$$(r_{A/I})(r_{I/JA}) + (r_{I/A^*})(r_{A/I}) + (r_{I/A^*})(r_{I/JA}) = 0$$

Now division by $(r_{A/I})(r_{I/JA})(r_{I/A^*})$ gives

$$\frac{1}{r_{I/A^*}} + \frac{1}{r_{I/JA}} + \frac{1}{r_{A/I}} = 0$$

or

$$\frac{1}{r_{JA/I}} = \frac{1}{r_{A/I}} - \frac{1}{r_{A^*/I}} \tag{A.12}$$

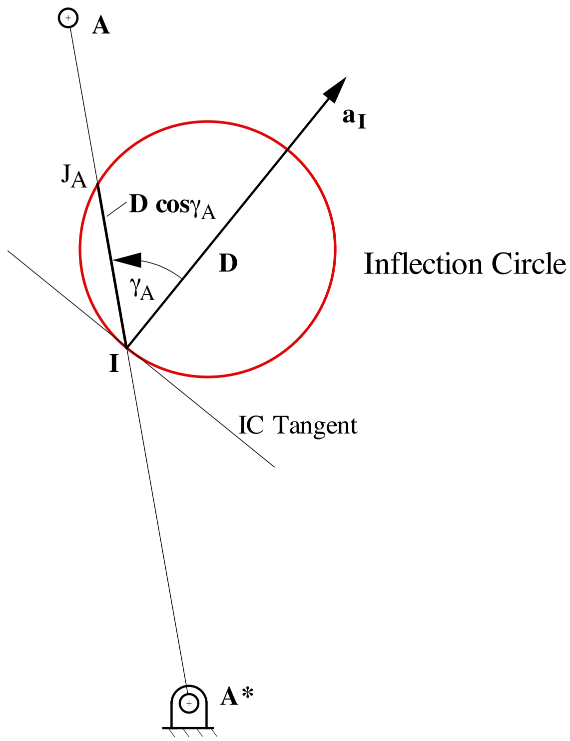
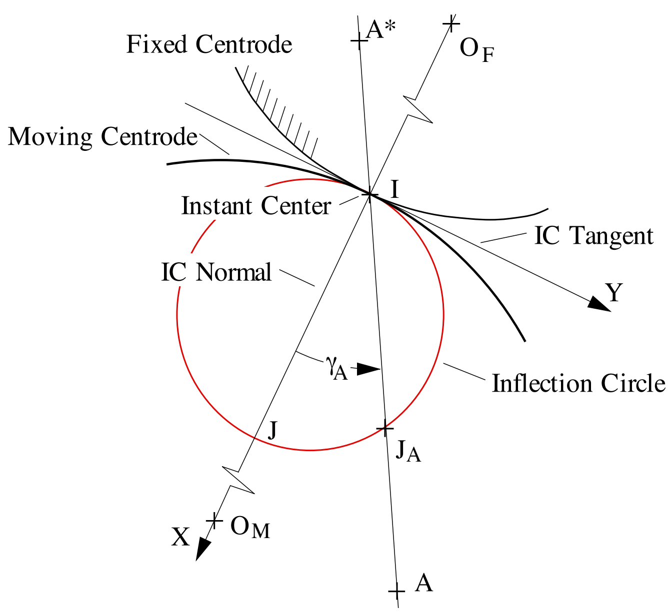


Fig. A.7: The geometric relationship of the inflection circle with a center and circle point pair A*A.

Some of the different forms for the Euler-Savary Equation are summarized in Table A.1. The terms used in Table A.1 are shown schematically in Fig. A.8. Most of the forms can be derived directly from Eq. (A.11) as was done in the case of Eq. (A.12); however, several of the forms are based on O_M and O_F, the centers of curvature of the moving and fixed centrodes corresponding to the instant center. These forms are derived by Hall[2]. Each form of the equation is based on a



[2]Hall, A.S., *Kinematics and Linkage Design*. Balt Publishers, West Lafayette, IN (1961).

Fig. A.8: Summary of terms for Euler-Savary equation

Table A.1 Summary of forms of Euler-Savary equation

Using the ray I-A, different forms of the Euler-Savary Equation are:

$$r_{A/A^*} = \frac{r_{A/I}^2}{r_{A/JA}} \qquad \frac{1}{r_{OM/I}} - \frac{1}{r_{OF/I}} = \frac{1}{r_{J/I}}$$

$$r_{A/A^*} = \frac{r_{A/I}^2}{r_{A/I} - r_{JA/I}} \qquad \frac{1}{r_{JA/I}} = \frac{1}{r_{A/I}} - \frac{1}{r_{A^*/I}}$$

$$r_{A/A^*} = \frac{r_{A/I}^2}{r_{A/I} - r_{J/I}\cos\gamma_A} \qquad r_{A/I} = \frac{r_{JA/I}\, r_{A^*/I}}{r_{JA/I} + r_{A^*/I}}$$

$$\frac{1}{r_{OM/I}} - \frac{1}{r_{OF/I}} = \left(\frac{1}{r_{A/I}} - \frac{1}{r_{A^*/I}}\right)\cos\gamma_A \qquad r_{JA/I} = r_{A/I} - \frac{r_{A/I}^2}{r_{A/A^*}^2} r_{A/A^*}$$

single ray through the instant center I. Therefore, relative to the ray, each vector can be treated as a signed (±) number. One direction from I can be taken arbitrarily as positive; distances in the other direction are automatically taken as negative. Examples of different locations of circle points and center points are shown in Fig. A.9.

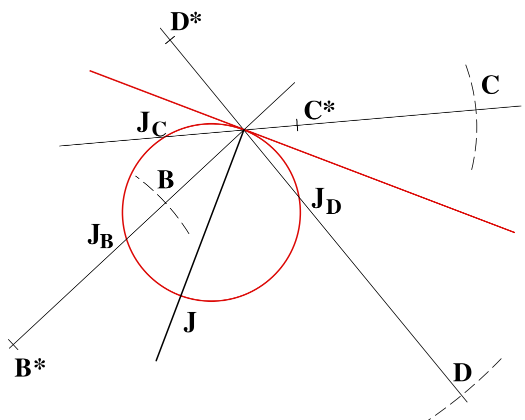


Fig. A.9: Locations for different points according to the Euler-Savary equation.

Example A.4 (Locating the Inflection Circle for Four-Bar Linkage)

Problem:

Locate the inflection circle for the four-bar linkage shown in Fig. A.10.

Solution:

To find the inflection circle, we need to find three points lying on it. Three points that can be found from the information given are I, J_A, and J_B. First locate the instant center I. From Chapter 3, the location is where an extension of AA^* intersects the line defined by BB^*. Next find J_A. This can be found by rewriting Eq. (A.11) as

$$r_{A/JA} = \frac{r_{A/I}^2}{r_{A/A^*}} \tag{A.13}$$

From the geometry given in Fig. A.10, $r_{A/I} = AB\sin(30^\circ) = 2$. Substituting numbers into Eq. (A.13), $r_{A/JA} = \frac{r_{A/I}^2}{r_{A/A^*}} = \frac{2^2}{4} = 1$ in the direction of r_{A/A^*}. This locates J_A between A and A^*. Next compute J_B using

$$r_{B/J_B} = \frac{r_{B/I}^2}{r_{B/B^*}}$$

From the geometry given in Fig. A.10, $r_{B/I} = AB\cos(30^\circ) = 2\sqrt{3}$. Substituting numbers into Eq. (A.13) again gives $r_{B/J_B} = \frac{r_{B/I}^2}{r_{B/B^*}} = \frac{(2\sqrt{3})^2}{4\sqrt{3}} = \sqrt{3}$ in the direction of r_{B/B^*}. This locates J_B between B and B^* also. Given I, J_A, and J_B, the inflection circle can be drawn as shown in Fig. A.10.

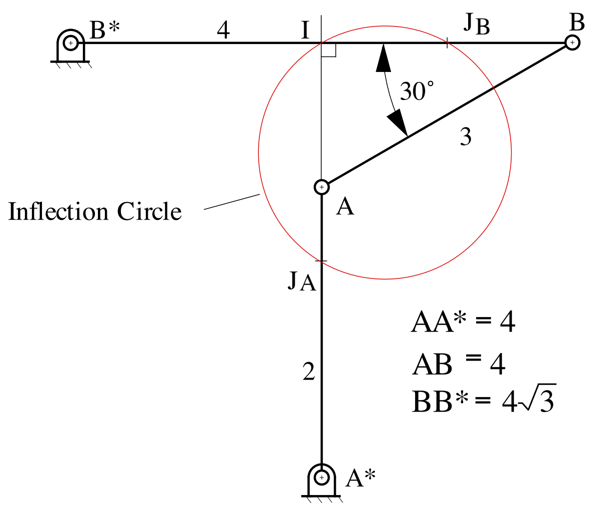


Fig. A.10: Inflection circle for four-bar linkage in Example A.4

Example A.5 (Inflection Circle for Slider-Crank Mechanism)

Problem:

Locate the inflection circle for the slider-crank mechanism shown in Fig. A.11. The link dimensions are $AA^* = 2$ m and $AB = 4$ m.

Solution:

Again, to find the inflection circle, we need to find three points lying on it. Three points which can be found from the information given are I, J_A, and J_B. First locate the instant center I using the procedure given in Chapter 3. The distance AB is given by

$$AB = 2\cos 30 + \sqrt{1^2 + 2^2} = 3.968$$

and $r_{A^*/I}$ is given by

$$r_{A^*/I} = 3.968 / \cos 30 = 4.582.$$

Also,

$$r_{A/I} = 4.582 - 2 = 2.582$$

and

$$r_{B/I} = (r_{A/I} + r_{A/A^*}) * \sin(30°) = 2.291$$

Next find J_A using Eq. (A.13). For the values given, $r_{A/JA} = \dfrac{r_{A/I}^2}{r_{A/A^*}} = \dfrac{2.582^2}{2} = 3.333$ in the direction of r_{A/A^*}. Therefore, A^* is between A and J_A as shown in Fig. A.11.

Next compute J_B. From Eq. (A.13), $r_{B/JB} = \dfrac{r_{B/I}^2}{r_{B/B^*}} = \dfrac{2.291^2}{\infty} = 0$. That is, J_B is located at B. We could have determined this by inspection because point B traces a straight path on the frame. Therefore, B must be on the inflection circle by definition. Given I, J_A, and J_B, the inflection circle can be drawn as shown in Fig. A.11.

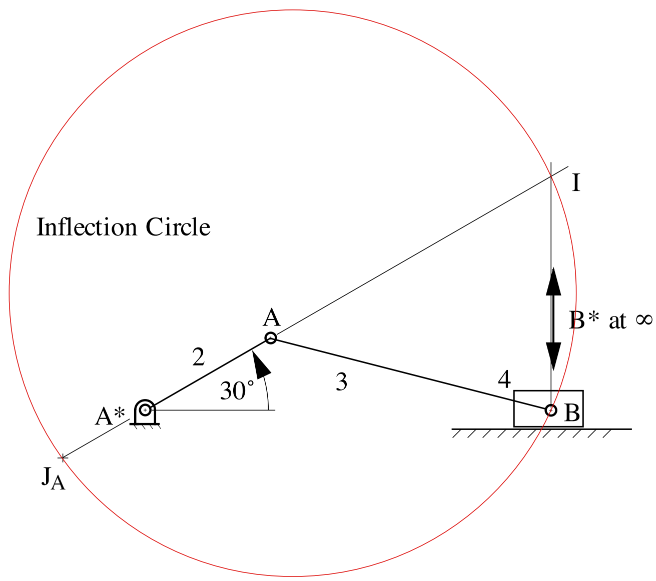


Fig. A.11: Inflection circle for slider-crank mechanism in Example A.4.

Example A.5 (Inflection Circle and Radius of Curvature)

Problem:

Determine the radius of curvature of the path that point C_3 in Fig. A.12 traces on the frame. Link 3 rolls on link 4 without slipping. The dimensions for the linkage are as follows: AA^* = 1 cm, B^*A^* = 1, AC = 2 cm, and the radius of the roller is 0.2 cm.

Solution:

To solve the problem, we first need to find the inflection circle. As in Examples A.2 and A.3, we need to find three points lying on the inflection circle to define it. Three points which can be found from the information given are I, J_A, and J_B. Point B_3 is not indicated directly on the drawing, however, we can locate B_3 by visualizing the path that B* traces on link 3. That path is a straight line; therefore, the center of curvature of the path is at infinity. Points B and B^* switch roles when we invert the motion and make link 3 the reference and allow the frame to move. Thus, B_3 is the center of curvature of the path of B^* relative to link 3, and B^* is the center of curvature of the path of B_3 relative to the frame. Therefore, in this problem, B_3 is at infinity in the direction indicated in Fig. A.12.

Locate the instant center, I, by finding the intersection of the rays through BB^* and AA^*. To find the intersection, the angle ϕ is required. From geometry, this is given by

$$\phi = \cos^{-1}\left(\frac{0.2}{1.414}\right) = 81.869^{\circ}$$

Then,

$$r_{A^*/I} = r_{A^*/B^*}\tan(\phi - 45) = 1\tan(36.869) = 0.750$$

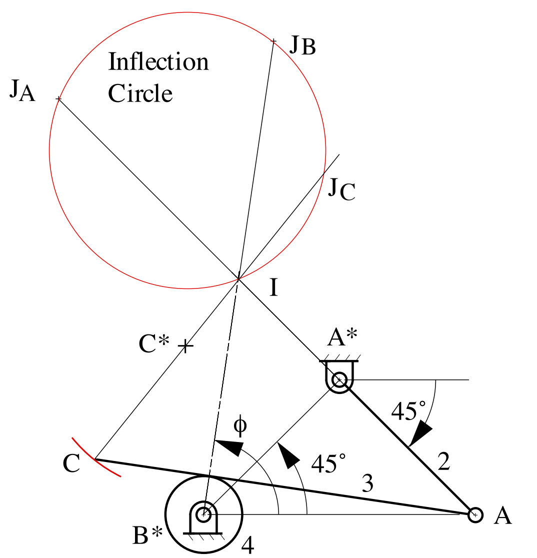


Fig. A.12: Inflection circle for Example 4.11

Next find J_A. using Eq. (A.13). That is, $r_{A/JA} = \frac{r_{A/I}^2}{r_{A/A^*}} = \frac{(1+0.75)^2}{1} = 3.062$ in the direction of r_{A/A^*}. Therefore, A^* is between A and J_A as shown in Fig. A.12.

To find the location of J_B, we cannot use Eq. (A.13) because B is at infinity. Instead, we can use the form of the equation given by Eq. (A.12). That is,

$$\frac{1}{r_{JB/I}} = \frac{1}{r_{B/I}} - \frac{1}{r_{B^*/I}}$$

or

$$\frac{1}{r_{JB/I}} = \frac{1}{\infty} - \frac{1}{r_{B^*/A^*}/\cos(\phi - 45)}$$

or

$$r_{JB/I} = -r_{B^*/A^*}/\cos(\phi - 45) = -1/\cos(36.869) = -1.250$$

or 1.250 in the <u>opposite</u> direction of $r_{B^*/I}$. Therefore, I is between B^* and J_B as shown in Fig. A.12.

To locate the center C^*, we must first find J_C by drawing a ray from C through I as shown in Fig.

A.12. We can measure $r_{I/JC}$ directly to be 0.711. Also, $r_{C/I} = 1.197$. Then from Eq. (A.13),

$$r_{C/JC} = \frac{r_{C/I}^2}{r_{C/C^*}}$$

or $r_{C/C^*} = \dfrac{r_{C/I}^2}{r_{C/JC}} = \dfrac{(1.197)^2}{(1.197 + 0.711)} = 0.752$ in the same direction as $r_{I/JC}$

Therefore, C* is between C and I. The location is shown in Fig. A.12. The approximate path of C is also drawn in Fig. A.12.

A.4 Relationship Among IC, Centrodes, IC Tangent, and IC Velocity

The relative motion between two rigid bodies is equivalent to two curves called centrodes rolling on each other as discussed in Chapter 2. One centrode is fixed to one body, and the second is fixed to the other body. This is represented in Fig. A.13 for the coupler of a four-bar linkage.

The point of contact is the instant center, and the centrodes are the paths of the instant centers on the two bodies. The instant center (IC) tangent is the common tangent to the two centrodes. The IC velocity is the instantaneous velocity with which the IC shifts; it is along the IC tangent. Note that the point that has the IC velocity will belong to neither of the rigid bodies being considered. Relative to the two bodies, the IC is at a different location for each relative position of the two bodies. This situation is shown in Fig. A.14. In that figure, the instant center I_{13} is in a different location relative to links 1 and 3 for each position of the linkage. The path of the instant center is defined by the path of point I_5 relative to the frame where link 5 is the ball captured between the two yokes in Fig. A.14. This path will be the fixed centrode. For any instantaneous position, the location of point I_5 coincides with the instant center, I_{13}, and the velocity of I_5 is the IC velocity discussed above.

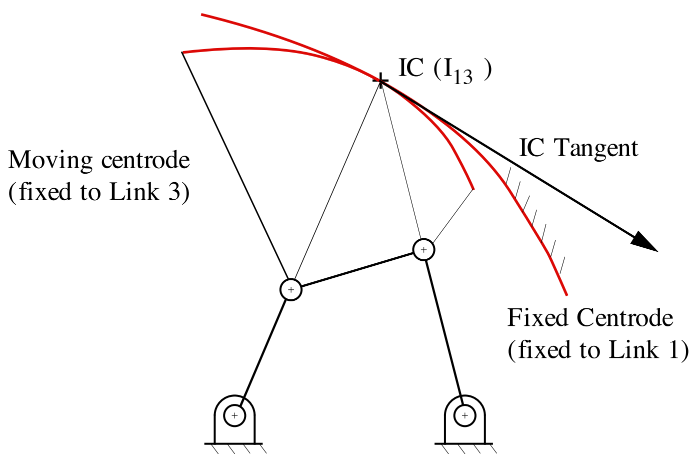


Fig. A.13: Location of instant center I_{13} and centrodes for a four-bar linkage.

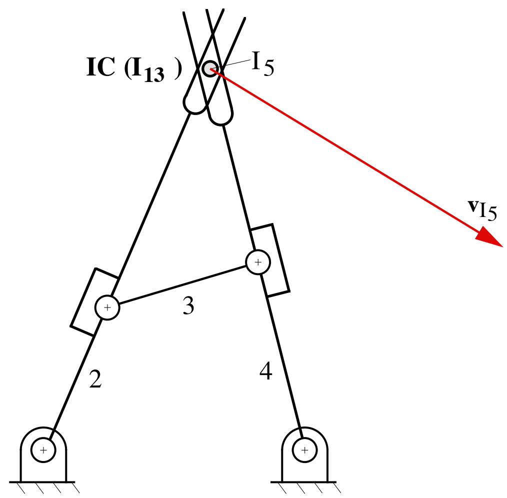


Fig. A.14: The path of I_{13} can be traced by I_5 as shown. Here, link 5 is the ball captured by the two yokes on links 2 and 4.

A.5 Analytical Form for Euler Savary Equation

The approach used in Examples A.3 - A.5 uses one of the forms of the Euler-Savary Equation given in Table A.1. These equations lend themselves to the graphical solution of the Euler-Savary Equation. To use the equations, we must establish a positive direction and identify that direction in the calculations. When programming the equations, it is convenient to work initially with points or absolute vectors rather than relative vectors. From the absolute vectors, the vectors in Table A.1 can be established. For example,

$$r_{A/A^*} = |\mathbf{r}_{A/A^*}| = |\mathbf{r}_A - \mathbf{r}_{A^*}| \qquad r_{A/I} = |\mathbf{r}_{A/I}| = |\mathbf{r}_A - \mathbf{r}_I|$$

$$r_{A/JA} = |\mathbf{r}_{A/JA}| = |\mathbf{r}_A - \mathbf{r}_{JA}| \qquad r_{J/I} = |\mathbf{r}_{J/I}| = |\mathbf{r}_J - \mathbf{r}_I|$$

etc. With these substitutions, the equations in Table A.1 can be programmed easily to compute the unknowns. MATLAB routines for the most common calculations are on the disk included with this book. Combinations of these routines can be used to write programs for finding the inflection circle and determining the center of curvature of selected points on different links. A routine for making these calculations are given for a four-bar linkage.

A.6 The Bobillier Constructions

As indicated in Example A.3, if we have a four-bar linkage, we can determine the inflection circle by locating J_A and J_B. However, calculations are required to locate these two points. The Bobillier constructions allow the inflection circle to be determined without calculations. The Bobillier constructions are graphical solutions of the Euler-Savary equation for a four-bar linkage. That is, they permit the location of the center point corresponding to a given circle point for three infinitesimally separated positions.

A.6.1 Bobillier's Theorem

Bobillier's theorem states that the angle between the centrode tangent at the instantaneous center of the coupler relative to the base of a four-bar linkage and one of the cranks is equal to the angle between the other crank and the collineation axis. The collineation axis is the line joining the instantaneous center of the coupler relative to the base to the instantaneous center of one crank relative to the other as shown in Fig. A.15. This theorem permits easy location of the centrode tangent. A line normal to the centrode tangent at the instant center gives a locus for the center of the inflection circle.

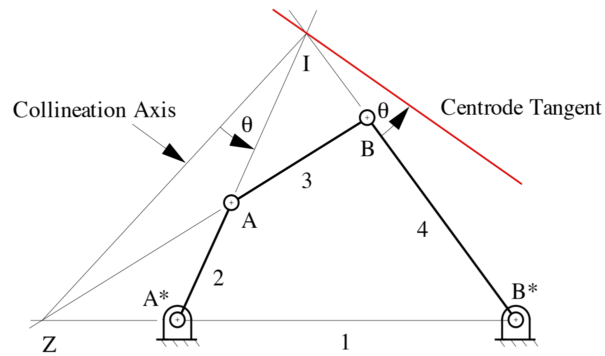


Fig. A.15: Statement of Bobillier's Theorem. The theorem states that the angles marked θ are equal.

Proof

For a four-bar linkage, we can find the IC tangent for I_{13} by a simple relative velocity analysis. Referring to Fig. A.16, let the instant center location for I_{13} be designated simply as I. Also, let I_5 be a point on rigid body 5 which traces the path of the instant center as shown in Figs. A.14 and A.16. Then the following relationships apply:

$${}^1\boldsymbol{v}_{I5} = {}^1\boldsymbol{v}_{I2} + {}^1\boldsymbol{v}_{I5/I2} = {}^1\boldsymbol{v}_{I4} + {}^1\boldsymbol{v}_{I5/I4}$$

$${}^1\boldsymbol{v}_{I4/I2} = {}^2\boldsymbol{v}_{I4/I2} = {}^2\boldsymbol{v}_{I4/\text{any pt. in system 2}}$$

$$= {}^2\boldsymbol{v}_{I4/\text{any pt. in system 4 with zero velocity relative to System 2}} = {}^2\boldsymbol{v}_{I4/(I24)_2} = {}^2\boldsymbol{v}_{I4/(I24)_4}$$

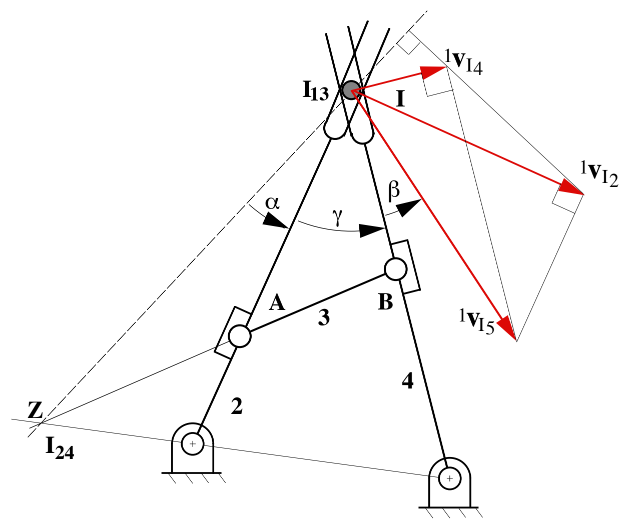


Fig. A.16: Velocity polygon for determining the velocity of the instant center, I_5.

Now,

${}^1\boldsymbol{v}_{I_2}$ is perpendicular to line AI,

${}^1\boldsymbol{v}_{I_4}$ is perpendicular to line BI,

${}^1\boldsymbol{v}_{I_5/I_2}$ is parallel to line AI,

${}^1\boldsymbol{v}_{I_5/I_4}$ is parallel to line BI, and

$^1\boldsymbol{v}_{I_2/I_4}$ is perpendicular to the line from I_{24} to I_{13} (Line *ZI*).

Because of the right angles indicated, the ends of vectors $^1\boldsymbol{v}_{I_2}$ and $^1\boldsymbol{v}_{I_4}$ lie on a circle with $^1\boldsymbol{v}_{I_5}$ as the diameter. A detailed representation of the angles involved is shown in Fig. A.17. Because quadrilateral Iacd is inscribed in a circle, two observations can be made from plane geometry:

a) Opposite angles of the quadrilateral are supplementary
b) All angles inscribed by the same chord segment (or equal segments) are equal.

Therefore,

$$\gamma + \beta = \frac{\pi}{2} - \rho$$

$$\gamma + \eta = \frac{\pi}{2} - \alpha$$

Then,

$$\gamma - \frac{\pi}{2} = -(\beta + \rho)$$

$$\gamma - \frac{\pi}{2} = -(\eta + \alpha)$$

and

$$\beta + \rho = \eta + \alpha$$

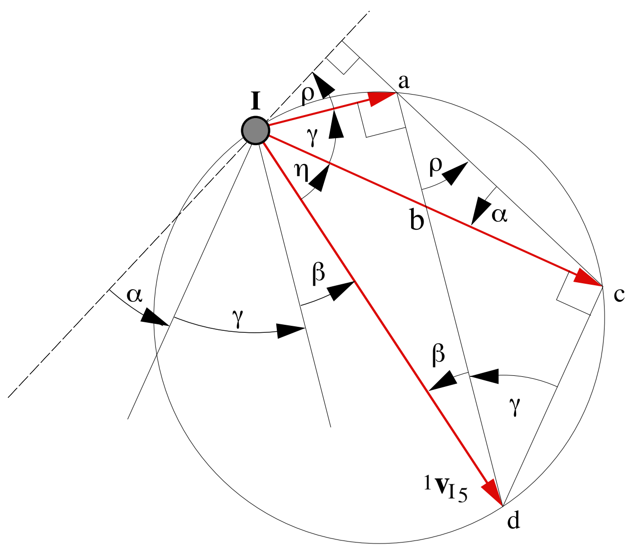


Fig. A.17: Details of the velocity polygon in Fig. A.16.

Also triangles dIc and dac contain a common chord line as a side. Therefore, $\rho = \eta$, which requires that $\alpha = \beta$. Comparing Figs. A.15 and A.16, it is clear that $a = \beta = \theta$. which proves the theorem.

A consequence of the Bobillier theorem is that the direction of the velocity of I_5 is a purely geometric quantity as it should be since the direction of the tangent to the centrodes at the contact point (instant center location) is a purely geometric quantity.

A.6.2 First Bobillier Construction

Given the centrode tangent and inflexion circle, construct the center of curvature of the path of any nominated point. The steps are given below, and the construction is shown in Fig. A.18.

Steps

1) Select the circle point C
2) Locate point J on the opposite end of the diameter of the circle from I.
3) Draw line CI and construct the normal to CI at I.
4) Locate point G at the intersection of the normal to CI at I and line CJ.
5) Construct the normal to the centrode tangent through point G.
6) Locate center point C^* at the intersection of the normal to the centrode tangent through G and line CI.

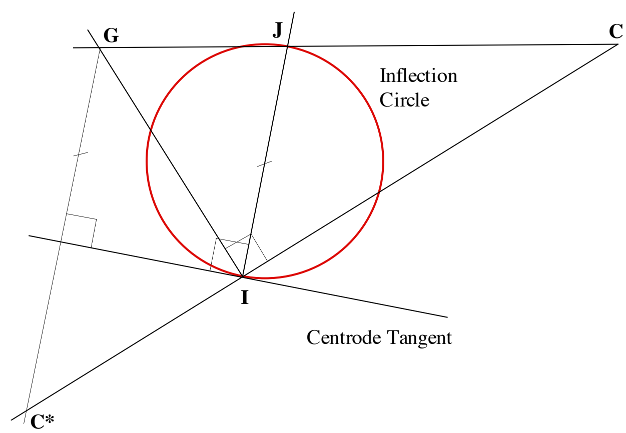


Fig. A.18: First Bobillier construction.

Proof

$$r_{C/I} = IC,\ \ \rho = r_{C/C^*},\ \ D = IJ \text{ and } \gamma_C = \angle JIC$$

Triangles IJC and C^*GC are similar so

$$\frac{C^*C}{IC} = \frac{C^*G}{IJ}$$

and $\angle GC^*I = \angle JIC = \gamma_C$.

Therefore

$$C^*G = \frac{C^*I}{\cos\gamma_C} = \frac{C^*C - IC}{\cos\gamma_C} = \frac{\rho - r_{C/I}}{\cos\gamma_C}$$

Also

$$\frac{C^*C}{IC} = \frac{\rho}{r_{C/I}} = \frac{C^*G}{IJ} = \frac{\rho - r_{C/I}}{D\cos\gamma_C}$$

giving

$$\rho D\cos\phi = \rho r_{C/I} - r_{C/I}^2$$

or

$$\rho = \frac{r_{C/I}^2}{r_{C/I} - D\cos\gamma_C}$$

which is the Euler-Savary equation (Eq. A.10).

A.6.3 Second Bobillier Construction

Given the inflexion circle and the instantaneous center, find the center points corresponding to two nominated circle points. The steps are given below and the construction is given in Fig. A.19.

Steps

1) Select circle points C and D and draw line CD.

2) Join points C and D to the instantaneous center I to locate points J_C and J_D at the intersections of the junction lines with the inflexion circle.

3) Join points J_C and J_D to locate point E at the intersection of lines CD and J_CJ_D.

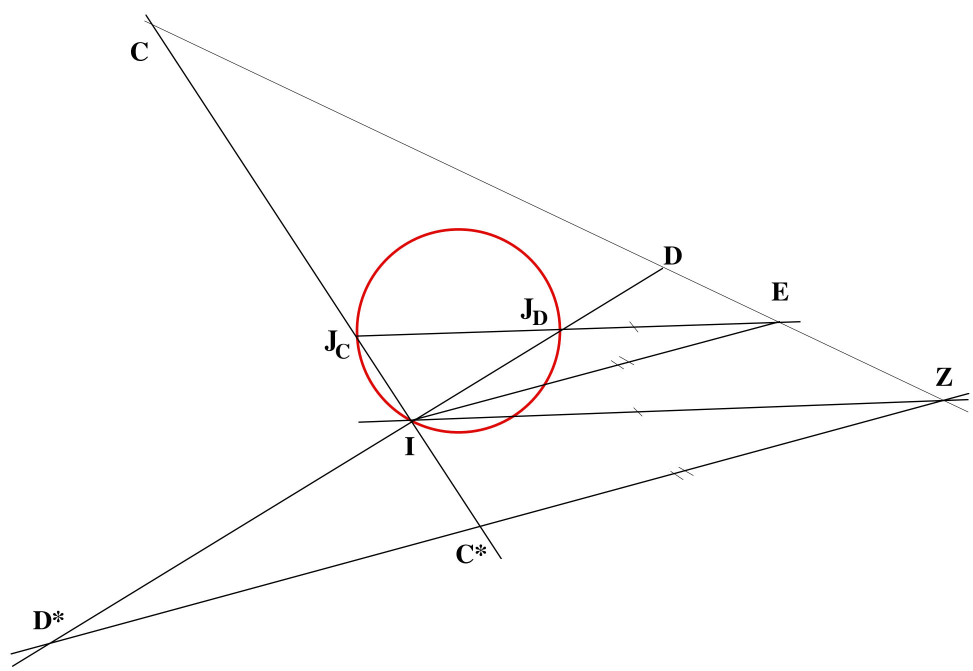


Fig. A.19: The second Bobillier construction.

4) Join I to E.

5) Draw a line parallel to J_CJ_D through I. Its intersection with line CD gives point Z. The collineation axis is line IZ.

6) Draw a line through Z parallel to IE. Its intersections with lines CI and DI give the center points C^* and D^* respectively.

Proof

Triangle ICE is similar to triangle C^*CZ

so $\quad \dfrac{IC}{C*C} = \dfrac{IE}{C*Z}$.

Also triangle IJ_CE is similar to triangle $C*IZ$

so $\quad \dfrac{IE}{C*Z} = \dfrac{IJ_C}{C*I}$

hence $\quad \dfrac{IC}{C*C} = \dfrac{IJ_C}{C*I}$

Now

$$\mathrm{IC} = \mathrm{r}_{\mathrm{C/I}},\ \mathrm{C*C} = \rho,\ \mathrm{IJ_C} = \mathrm{D}\cos\gamma_{\mathrm{C}},\ \ \mathrm{C*I} = \rho - \mathrm{r}$$

so $\quad \dfrac{\mathrm{r}_{\mathrm{C/I}}}{\rho} = \dfrac{\mathrm{D}\cos\gamma_{\mathrm{C}}}{\rho - \mathrm{r}_{\mathrm{C/I}}}$ or $\rho = \dfrac{\mathrm{r}^2_{\mathrm{C/I}}}{\mathrm{r}_{\mathrm{C/I}} - \mathrm{D}\cos\gamma_{\mathrm{C}}}$.

A similar proof holds for point *D*.

Bobillier's second construction is of greater importance when used in reverse. It then becomes a means of constructing the inflexion circle of a given four-bar linkage. The steps are given below and the construction is shown in Fig. A.20.

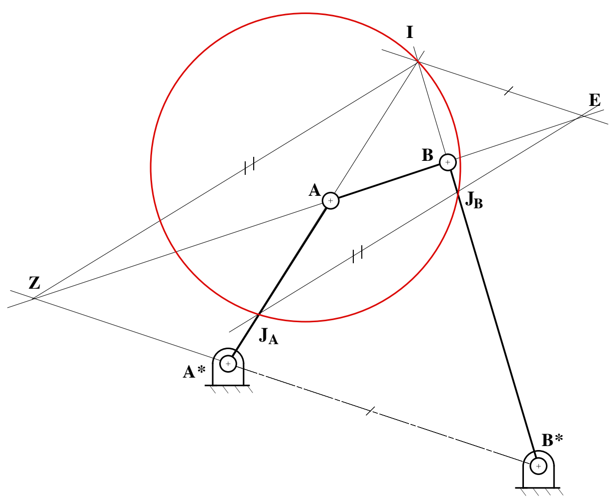


Fig. A.20: The second Bobillier construction used in reverse to find the inflection circle of a given four-bar linkage.

Steps

1) Locate the instantaneous centers *I* and *Z* and draw the collineation axis *IZ*.

2) Draw a line through *I* parallel to line *A*B**. Its intersection with line *AB* is *E*.

3) Draw a line through *E* parallel to *IZ*. Its intersections with lines *A*A* and *B*B* are points J_A, J_B, respectively.

4) Draw a circle through points *I*, J_A, J_B. This is the inflexion circle.

Appendix B
Procedure for Drawing Conjugate Tooth Form.

B.1 General Conjugate Tooth Forms

The fundamental law of gearing requires that when two gears are in contact, the angular velocity ratio is inversely proportional to the lengths of the two line segments created by the intersection of the common normal to the two contacting surfaces and the line of centers. This ratio is constant if the common normal intersects the centerline at a fixed point, the pitch point. The tooth forms satisfying this condition are said to be conjugate. The flat sided rack and involute tooth form are one example of conjugate tooth forms; however, there are an infinite number of other tooth forms which can be conjugate. In this section, we will generalize the procedure given in Section 10.12 of the textbook to develop a procedure for finding the tooth form, which is conjugate to a general tooth form.

The information in Appendix B was originally contained in the textbook; however, it was removed because of page constraints. Therefore, the entire development is given here. The program that will draw a conjugate gear is described in Chapter 5 of this manual.

B.1.1 Required Geometric Parameters

Several parameters must be known about both gears to determine the unknown tooth form which is conjugate to the known tooth form. These parameters include the pitch radii, tooth numbers, and gear type (i.e., internal or external), and a mathematical function for the gear tooth form on the known gear. If the function is not known directly, it is possible to fit a spline to a set of points describing the tooth form. Ultimately, it is necessary to be able to compute the normal vector to the known gear tooth at each location.

B.1.2 Determination of the Point of Contact

Assume that the known gear is gear 2 and the unknown gear is gear 3. Each gear will have a coordinate system attached as shown in Fig. B.1, and the local gear geometry will be defined relative to the coordinate system fixed to each gear. To satisfy the fundamental law of gearing, the line normal to the tooth surfaces must pass through the pitch point as shown by line *AP* in Fig. B.1. The line segment *AP* is a straight line which has the following equation:

$$y = mx + b$$

Here, m is the slope of the line which is the direction of the normal to the known gear at point *A* and *b* is the *y* intercept. An expression for *b* can be found by recognizing that the line passes through point *A*. Therefore,

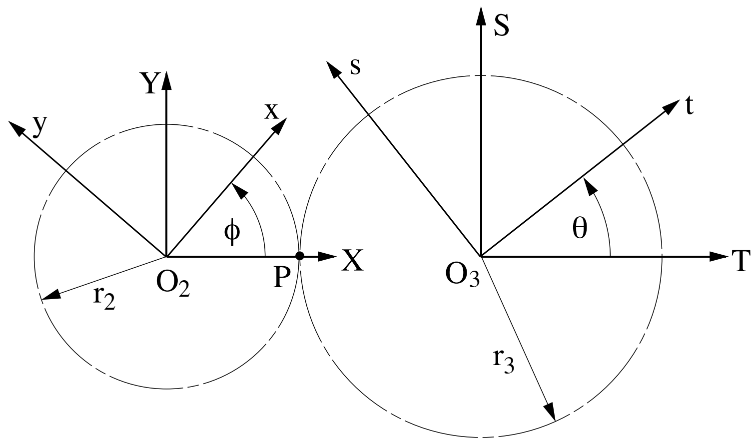


Fig. B.1: Coordinate systems on two gears.

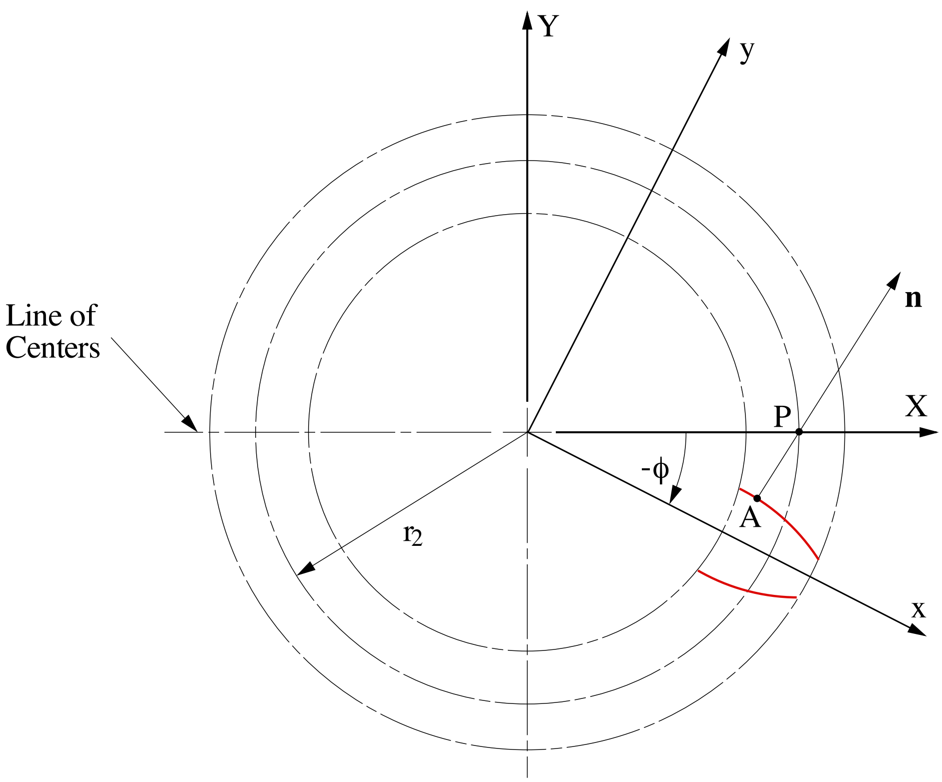


Fig. B.2: Line through pitch point and normal to tooth profile.

$$y_A = mx_A + b$$

or

$$b = y_A - mx_A$$

If the slope of the normal is represented by

$$n_A = \left(\frac{n_{A_y}}{n_{A_x}}\right)$$

then an expression for the line segment is given by

$$y = (x - x_A)n_A + y_A \tag{B.1}$$

Two special cases exist for the line. The first occurs when the line is horizontal. Then $y = y_A$ regardless of x. When the line is vertical, then $x = x_A$ for all values of y.

Note that in Eq. (B.1), we assume that the components of the normal vector are known. If only an equation for the tooth profile is known, we can obtain the normal to the curve at any point by differentiation. For example, if the tooth profile is given by

$$y = F(x),$$

then the slope of the normal vector is given by

$$n = -\frac{1}{dy/dx}$$

where the derivative is evaluated at the point of interest. If x and y are given as parametric expressions, for example,

$$y = r\sin\theta$$
$$x = r\cos\theta$$

then the slope of the normal can be computed from

$$n = -\frac{1}{dy/dx} = -\frac{1}{\left(\frac{dy}{d\theta}\right)\Big/\left(\frac{dx}{d\theta}\right)}$$

Referring to Fig. B.2, the line AP through P can be written relative to the coordinate system attached to gear 2 as

$$\begin{aligned} x_P &= r_2\cos\phi \\ y_P &= -r_2\sin\phi \end{aligned} \tag{B.2}$$

where r_2 is the pitch circle radius of gear 2. The negative sign on y_P is present because ϕ is negative. Substituting Eqs (B.2) into Eq. (B.1), gives

$$-r_1\sin\phi = (r_1\cos\phi - x_A)n_A + y_A$$

or

$$r_1\sin\phi + r_1 n_A\cos\phi + y_A - x_A n_A = 0 \tag{B.3}$$

In a typical problem, both x_A and y_A will be specified. This will correspond to the contact point location for both gears, although the x_A and y_A specified will be relative to gear 2. We must find the coordinates relative to gear 3 to find the point on gear 3 which is conjugate to the point on the gear. However, to do this, we must first find the angle ϕ.

The angle ϕ can be found using the procedures given in Chapter 3 of the textbook. To begin, make the following substitutions

$$\cos\phi = \frac{\left[1-\tan^2\left(\frac{\phi}{2}\right)\right]}{\left[1+\tan^2\left(\frac{\phi}{2}\right)\right]}$$

$$\sin\phi = \frac{\left[2\tan\left(\frac{\phi}{2}\right)\right]}{\left[1+\tan^2\left(\frac{\phi}{2}\right)\right]}$$

$$\tau = \tan\left(\frac{\phi}{2}\right)$$

$$A = y_A - x_A n_A$$

$$B = n$$

$$C = n n_A$$

Then, the equation to be solved is

$$A + B\sin\phi + C\cos\phi = 0 = A + B\left[\frac{2\tau}{1+\tau^2}\right] + C\left[\frac{1-\tau^2}{1+\tau^2}\right]$$

and the solution is

$$\tau = \frac{-B + \sqrt{B^2 - A^2 + C^2}}{A - C}$$

and

$$\phi = 2\tan^{-1}\tau$$

Note that all points on the known gear may not be possible choices for a contact point. If the candidate point chosen is an impossible contact point, $B^2 - A^2 + C^2$ will be negative.

To locate the angle ϕ for all possible points x_A and y_A, it is only necessary to begin at one end of the known contour and increment x until the other end is reached. The increments of x need not be uniform.

B.1.3 Coordinate Transformations

Once the point of contact is located, it becomes necessary to transform the coordinates from gear 2 to gear 3. The transformation will involve the following parameters:

C_d = center distance for two gears

θ_0 = initial angle for axis t on gear 3 when the angle ϕ is zero.

N_2 = number of teeth on gear 2

N_3 = number of teeth on gear 3

The center distance is given by

$$C_d = r_2 + ir_3 \tag{B.4}$$

where "i" is equal to 1 for an external gear and -1 for an internal gear.

The initial angle θ_0 for the t axis on gear 3 is π minus the angle that subtends an arc which is one half of the tooth thickness measured at the pitch circle. This angle is equal to

$$\theta_0 = \pi - \pi / N_3 = \pi\left(1 - \frac{1}{N_3}\right) \tag{B.5}$$

The angles θ and ϕ are related by the ratio of the pitch radii. As ϕ increases, θ decreases for external gears. The resulting relationship is

$$\theta = -i\phi\frac{r_2}{r_3} + \theta_0 \tag{B.6}$$

The coordinates must be transformed through four sets of coordinate systems: local *xy* to global *XY*, to global *TS* , and finally to local *ts*. Referring to Fig. B.8, the *xy* and *XY* coordinate systems pertain to gear 2; whereas the *TS* and ts systems refer to gear 3. The *x* coordinate axis is along the center line of the tooth in gear 2 while the - *t* axis is along the centerline of the contacted gear on gear 3. The three successive transformations are given in the following.

$$\begin{Bmatrix} X \\ Y \end{Bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}\begin{Bmatrix} x \\ y \end{Bmatrix}; \quad \begin{Bmatrix} T \\ S \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{Bmatrix} X \\ Y \end{Bmatrix} - \begin{Bmatrix} C_d \\ 0 \end{Bmatrix}; \quad \begin{Bmatrix} t \\ s \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{Bmatrix} T \\ S \end{Bmatrix}$$

The overall transformation is

$$\begin{Bmatrix} t \\ s \end{Bmatrix} = \begin{bmatrix} \cos(\theta - \phi) & \sin(\theta - \phi) \\ -\sin(\theta - \phi) & \cos(\theta - \phi) \end{bmatrix}\begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} -C_d\cos\theta \\ C_d\sin\theta \end{Bmatrix} \tag{B.7}$$

These equations define the conjugate tooth form relative to gear 3.

Example B.1 (Conjugate Tooth Form for Straight Toothed-Gearing)

Problem:

One tooth form which has been used on very large gears such as the ring gear on draglines is a straight-toothed form. This is the same form as is used on a simple rack except that the pitch curve is a circle instead of a straight line. Therefore, the conjugate tooth form is not an involute. For the problem, assume that gear 2 has a pitch diameter (d_2) of 20 feet and the diametral pitch (D_p) is 5 teeth per foot of pitch diameter. The tooth surface is inclined with the centerline at an angle of ϕ = 25˚. The mating gear (gear 3) is an external gear with 30 teeth (N_2). The addendum constant (a_2, a_3) for each gear is 1, and the dedendum constant (b_2, b_3) is 1.2. Find the tooth form which is conjugate to gear 2 so that there will be a constant velocity ratio between the two gears.

Solution:

To find the conjugate tooth form, we must first find an expression for the coordinates of points on the gear tooth and for the components of the normal vectors. Figure B.3 shows an enlarged view of gear 2. The equations for the gear are similar to those for the hob in Example 8.5.

Before developing the equations, it is useful to compute several parameters. These are:

A_2 = Addendum of gear 2 = $\dfrac{a_2}{D_p} = \dfrac{1}{5} = 0.2\ ft$

B_2 = Dedendum of gear 2 = $\dfrac{b_2}{D_p} = \dfrac{1.2}{5} = 0.24\ ft$

γ = tooth angle = 25°

ℓ_t = gear 2 tooth thickness at tip = $\ell_t = \dfrac{\pi}{2D_p} - 2A_2 \tan\gamma = \dfrac{\pi}{2(5)} - 2(0.2)\tan 25 = 0.128\ ft$

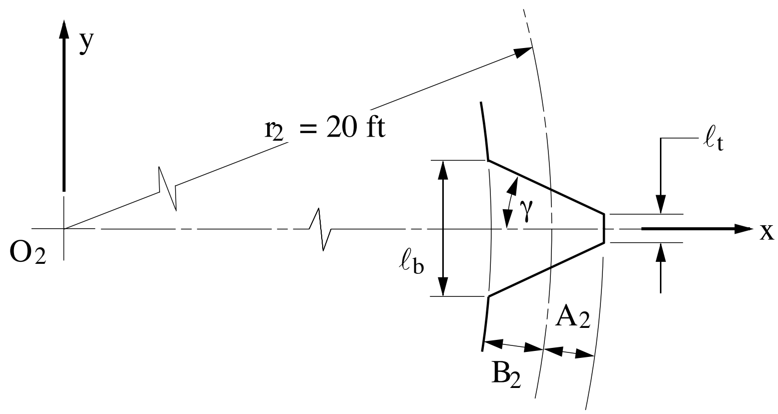


Fig. B.3: One tooth from pin gear.

ℓ_b = gear 2 tooth thickness at tip = $\ell_b = \dfrac{\pi}{2D_p} + 2B_2 \tan\gamma = \dfrac{\pi}{2(5)} + 2(0.24)\tan 25 = 0.538\ ft$

From that figure,

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{Bmatrix} r_2 - B_2 + \beta \\ \dfrac{\ell_b}{2} - \beta \tan\gamma \end{Bmatrix}; \quad \begin{Bmatrix} n_x \\ n_y \end{Bmatrix} = \begin{Bmatrix} \sin\gamma \\ \cos\gamma \end{Bmatrix}; \quad 0 \le \beta \le (A_2 + B_2) \tag{B.8}$$

We need consider only one side of the driving tooth because only one side will contact the corresponding tooth on gear 3 for a given direction of rotation. We can reflect the tooth about its centerline to find the other half.

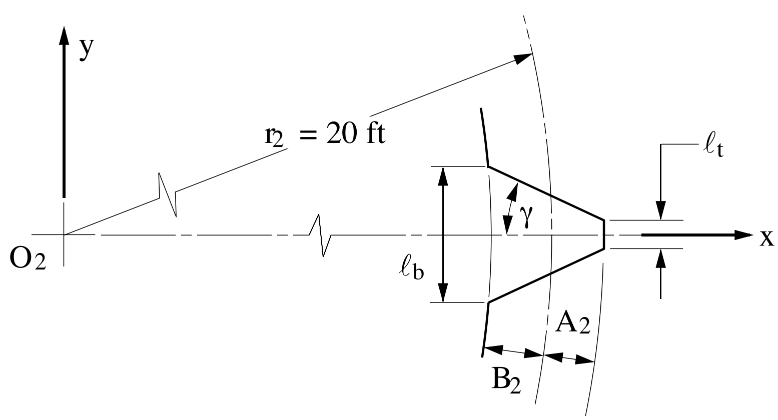


Fig. B.3: One tooth from pin gear.

The number of teeth on gear 2 is

$$N_2 = d_2 D_p = 20(5) = 100$$

The pitch radius of gear 3 is given by

$$r_3 = \frac{N_3}{N_2} r_2 = \frac{30}{100} 10 = 3 \; ft$$

and the center distance is given by Eq. (B.4) as

$$C_d = r_2 + i r_3 = 10 + (+1)(3) = 13 \; ft$$

The initial angle θ_0 is given by Eq. (B.5) as

$$\theta_0 = \pi\left(1 - \frac{1}{N_2}\right) = \pi\left(1 - \frac{1}{30}\right) = 3.037 \, rad$$

To find the conjugate gear form, it is only necessary to increment β from 0 to $(A_2 + B_2)$ and compute the (x, y) coordinates of the points and normals using Eq. (B.8). The angle ϕ corresponding to the selected point can then be found by solving Eq. (B.3) using the procedure given above.

The angles θ for a given value of ϕ is given by Eq. (B.6):

$$\theta = -i\phi \frac{r_1}{r_2} + \theta_0 = -\frac{2}{3}\phi + 0.209$$

Knowing θ and ϕ, the coordinates of the conjugate point on gear 3 are given by Eq. (6.39) or

$$\begin{Bmatrix} t \\ s \end{Bmatrix} = \begin{bmatrix} \cos(\theta - \phi) & \sin(\theta - \phi) \\ -\sin(\theta - \phi) & \cos(\theta - \phi) \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} + \begin{Bmatrix} -C_d \cos\theta \\ C_d \sin\theta \end{Bmatrix}$$

Once the values of t, s on gear 3 are known for each value of x, y, on gear 2, the tooth form on gear 3 can be computed. Clearly, this procedure is best done using a computer program to determine the tooth profile of gear 3.