

## Animation Programming with QuickBASIC to Aid the Teaching of Kinematics

Ing-Chang Jong, Stanley N. Onggowijaya  
University of Arkansas

### Abstract

Animation of mechanical systems is an effective means to enhance the learning of dynamics. Nevertheless, when the available software lacks the flexibility to animate a desired specific task, such as the generation of the locus of the velocity center of the coupler link of a four-bar linkage mechanism, what alternatives does an instructor have? This paper provides instructors of mechanics with an alternative to accomplish the said desired task by presenting a computer program written in QuickBASIC, which is easy to program, very flexible, low overhead, and widely available. The program is intended to complement available canned animation software in the teaching of kinematics. The paper provides an unabridged listing of the QuickBASIC program with helpful remarks. A sequence of sample pictures on the computer screen is included to illustrate the animation of the mechanism and the simultaneous plotting of the space centrode of the coupler.

### Introduction

Suppose that the crank  $AB$  of the four-bar linkage mechanism in Fig. 1 rotates with a counterclockwise angular velocity. This linkage is a crank-rocker mechanism since it has no range of lockup positions. It can readily be shown that the motion of this mechanism is subject to the following two constraint equations:

$$L_1 \cos \theta_1 + L_2 \cos \theta_2 + L_3 \cos \theta_3 - L_4 = 0 \quad (1)$$

$$L_1 \sin \theta_1 + L_2 \sin \theta_2 + L_3 \sin \theta_3 = 0 \quad (2)$$

where  $L_1 = \overline{AB} = 0.15$  m,  $L_2 = \overline{BD} = 0.25$  m,  $L_3 = \overline{DE} = 0.5$  m, and  $L_4 = \overline{AE} = 0.6$  m.

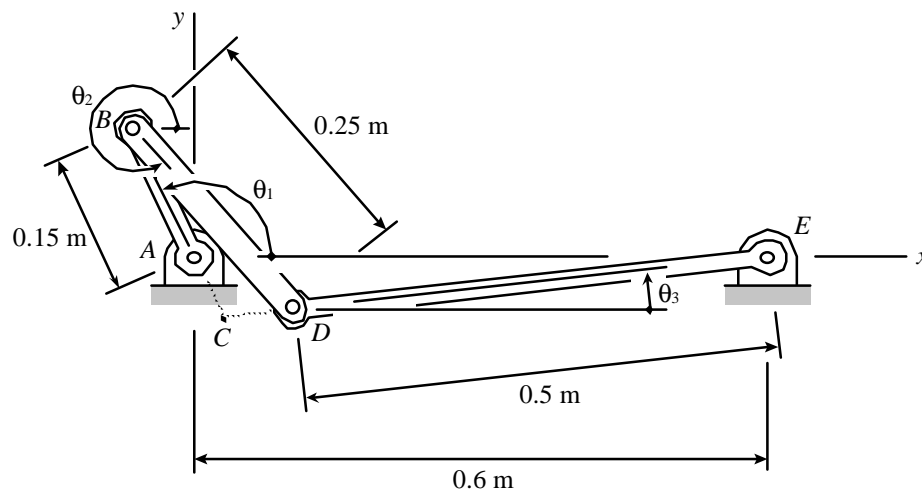


Fig. 1 A four-bar linkage with no lockup positions

The velocity center  $C$  of the coupler  $BD$  is simply the point of intersection of the lines  $AB$  and  $DE$  in Fig. 1, whereas the space centrode of the coupler is the locus of  $C(x_c, y_c)$ , whose coordinates are

$$x_c = -\frac{L_4 \tan \theta_3}{\tan \theta_1 - \tan \theta_3} \quad (3)$$

$$y_c = -\frac{L_4 \tan \theta_1 \tan \theta_3}{\tan \theta_1 - \tan \theta_3} \quad (4)$$

Clearly, the animation of the mechanism and the plotting of the space centrode of the coupler require the solutions of  $\theta_2$  and  $\theta_3$  from Eqs. (1) and (2) for every given value of  $\theta_1$ . The solutions are obtained by using the Newton-Raphson Method for nonlinear systems.<sup>1</sup> When the crank  $AB$ , the coupler  $BD$ , and the output  $DE$  are collinear with the ground link  $AE$ , the positions of the velocity center  $C$  could not be determined by the traditional *method of velocity center*. The positions of  $C$  were determined and discussed in detail by Jong, Reynolds, and Adams.<sup>2</sup> For the present system in the *collinear* configuration, it can be shown that  $y_c = 0$  and

$$x_c = \overline{AC} = \frac{3}{4 \pm 8\sqrt{2}} - 0.15 \quad (5)$$

which are singularity values for the system. Thus, in the collinear configuration, the velocity center  $C$  of the coupler link is located at  $C(0.04590291, 0)$  m when  $\theta_1 = 180^\circ$ , and at  $C(-0.56018862, 0)$  m when  $\theta_1 = 540^\circ$ .

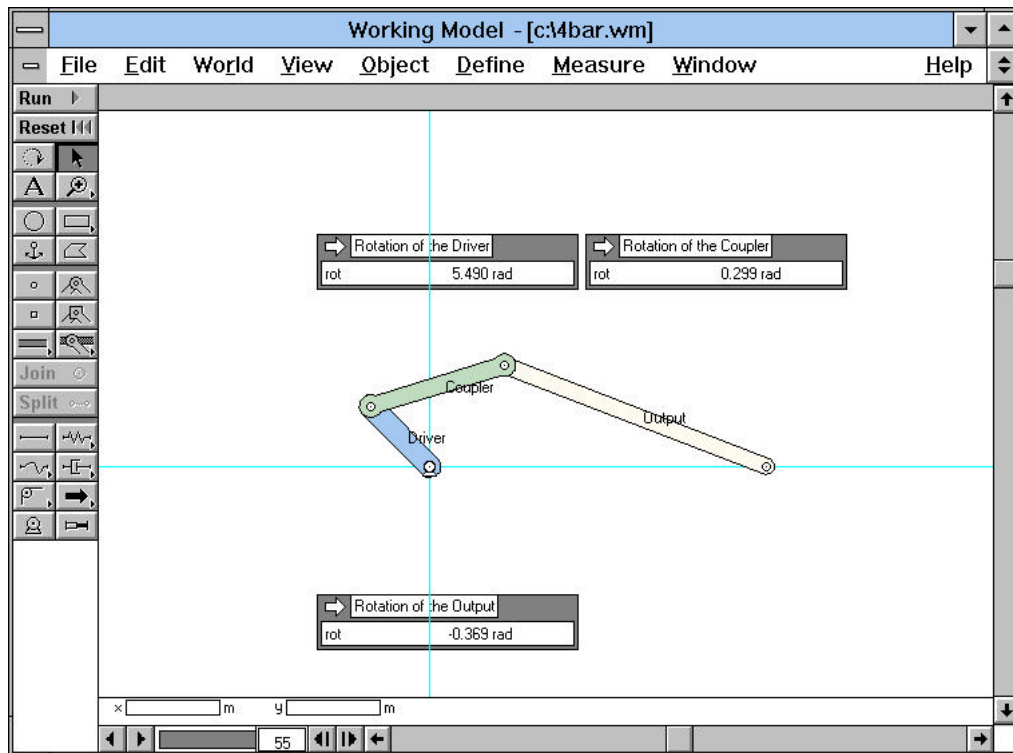


Fig. 2 Animation of a four-bar linkage mechanism using *Working Model*

## Animation of Four-Bar Linkage Mechanism and Plotting of Space Centrode

With the software *Working Model*,<sup>3</sup> one can readily animate the system in Fig. 1 as illustrated in Fig. 2, where the driver was set to start its counterclockwise rotation from the collinear configuration. However, the current version of this software does not lend itself to the animation of the generation of the space centrode of the coupler. To see the generation of the space centrode, one needs to resort to an alternative. We note that QuickBASIC<sup>4</sup> is widely available to users of IBM PC's, PC compatibles, and SoftWindows on Power Macintoshes. Because of its prevalence and built-in strength in graphics, QuickBASIC is an attractive choice to complement *Working Model*.

The key technique in producing a variety of animation effects is to *alternately* erase old lines and draw new lines in many different locations on the screen. To erase an old line, one simply redraws the old line *in the color of the background*. For example, to erase a line *AB* on a black background, one simply redraws the line *AB* in *black*. Naturally, to draw a new line, one draws it in a desired color that is *different* from the color of the background. Readers may refer to a QuickBASIC manual<sup>4</sup> for detailed descriptions of the various graphics statements. The program in this paper is written for a PC with VGA display adapter (SCREEN 12, 640-by-480-pixel resolution, 16 colors). Note that colors are specified along with other parameters in graphics statements. The codes for the 16 available colors are:

0 = black	1 = blue	2 = green	3 = cyan
4 = red	5 = magenta	6 = brown	7 = white
8 = gray	9 = light blue	10 = light green	11 = light cyan
12 = light red	13 = light magenta	14 = yellow	15 = bright white

The syntaxes of statements for the graphics and cursor control employed in the program are

```
CIRCLE [STEP](x, y), radius [, [color][, [start][, [stop][, aspect]]]]
COLOR [foreground]
LINE [STEP](x1, y1)–[STEP](x2, y2)[, [color][, [B[F]]], style]]
PAINT [STEP](x, y)[, [color][, [border][, background]]]
SCREEN [mode][, [color][, [active][, visible]]]
LOCATE [row][, [column][, [visible][, [start][, stop]]]]
```

Note that the *origins* of both the 640-by-480-pixel graphics screen and the 30-row-by-80-column text screen are at the upper left corner of the screen. Furthermore, note that positive *x* is to the right of the origin on the screen and in Fig. 1, whereas positive *y* is below the origin on the screen but above the origin in Fig. 1.

### A. QuickBASIC Program (Unabridged)

```
PROGRAM$ = "4BAR.BAS"
```

```
REM *** Program Name: ANIMATION & PLOTTING FOR A MECHANISM *****
REM *** Lengths of links and positions of A & E on the 640 x 480 screen ===
DEFDBL A-Z: L1 = .15: L2 = .25: L3 = .5: L4 = .6
Ax = 230: Ay = 240: Ex = 230 + L4 * 300: Ey = 240
```

```
REM === Setting IN = Increment of thetal (T1) in degrees in the animation
IN = 2: M = 180 / IN + 1: N = 540 / IN + 1: REM M & N: for singularities.
```

```

REM *** MAIN PROGRAM *****

CLS : START = TIMER: SCREEN 12
COLOR 15: PRINT "ANIMATION OF A FOUR-BAR LINKAGE MECHANISM &"
COLOR 15: PRINT "PLOTTING OF THE SPACE CENTRODE OF THE COUPLER"
DM = CINT(720 / IN) + 1: IF CINT(720 / IN) < 720 / IN THEN DM = DM + 1
DIM TS1(DM), TS2(DM), TS3(DM), XS(DM), YS(DM)
REM === Initial Values for Iterative Solutions =====
Bx = 50: By = 50: Dx = 50: Dy = 50: X = 50: Y = 50: S = 1
PI = 3.14159265#: DC = 180 / PI: INR = IN / DC: GPA = INR / 10
T1 = 0: T2 = 1.5 * PI: T3 = .2 * PI
REM === Iterative Solutions and printing of output on the screen =====
FOR I = 1 TO DM: X0 = T2: Y0 = T3: CC = 1: MC = 1
  GOSUB SOLVE: LOCATE 28, 38: COLOR 12: PRINT "θ1 = ";
  COLOR 15: PRINT USING "###.##° "; T1 * DC;
  COLOR 12: PRINT "θ2 = ";
  COLOR 15: PRINT USING "###.##° "; T2 * DC;
  COLOR 12: PRINT "θ3 = ";
  COLOR 15: PRINT USING "###.##° "; T3 * DC;
  TS1(I) = T1: TS2(I) = T2: TS3(I) = T3: GOSUB PLOT
  REM === Use of kinematic coefficients =====
  H2 = L1 * SIN(T3 - T1) / (L2 * SIN(T2 - T3))
  H3 = L1 * SIN(T1 - T2) / (L3 * SIN(T2 - T3))
  T1 = I * INR: T2 = T2 + H2 * INR: T3 = T3 + H3 * INR
  WHILE CINT(T1 * DC) = 180 OR CINT(T1 * DC) = 540
    T1 = T1 + GPA: T2 = T2 + H2 * GPA: T3 = T3 + H3 * GPA: WEND
  IF T1 > 720 / DC THEN T1 = 720 / DC
NEXT I: FIN = TIMER
LOCATE 28, 1: PRINT SPC(37); : SEC = FIN - START
LOCATE 28, 1: COLOR 9: PRINT "Elapsed time:";
PRINT USING "###.## "; SEC; : PRINT "seconds"
COLOR 14: PRINT "Press any key to connect dots and see asymptotes.";
DO WHILE INKEY$ = ""
LOOP: FOR I = 1 TO DM - 1: LINE (XS(I), YS(I))-(XS(I + 1), YS(I + 1)), 3
FOR J = 1 TO 1000: NEXT J: NEXT I: LINE (Ax, 80)-(Ax, 400), 8
LOCATE 30, 1: COLOR 15: PRINT "Press any key to continue.";
DO WHILE INKEY$ = ""
LOOP: END: REM *** END OF MAIN PROGRAM =====

REM *** SUBROUTINE *****
SOLVE:
REM === Numerical Solution Using Newton-Raphson Method =====
DO WHILE CC > .00001
  U0 = L1 * COS(T1) + L2 * COS(T2) + L3 * COS(T3) - L4
  V0 = L1 * SIN(T1) + L2 * SIN(T2) + L3 * SIN(T3)
  U0T2 = -L2 * SIN(T2): U0T3 = -L3 * SIN(T3)
  V0T2 = L2 * COS(T2): V0T3 = L3 * COS(T3)
  T2 = X0 - (U0 * V0T3 - V0 * U0T3) / (U0T2 * V0T3 - U0T3 * V0T2)
  T3 = Y0 + (U0 * V0T2 - V0 * U0T2) / (U0T2 * V0T3 - U0T3 * V0T2)
  IF T2 > 2 * PI THEN T2 = T2 - 2 * PI
  IF T3 > 2 * PI THEN T3 = T3 - 2 * PI
  IF T2 < 0 THEN T2 = T2 + 2 * PI
  IF T3 < 0 THEN T3 = T3 + 2 * PI
  CC1 = ABS((T2 - X0) / X0): CC2 = ABS((T3 - Y0) / Y0)
  IF CC1 > CC2 THEN CC = CC1
  IF CC1 < CC2 THEN CC = CC2
  X0 = T2: Y0 = T3: MC = CINT(MC) + 1: IF MC = 101 THEN CC = .000001
LOOP: RETURN: REM *** END OF SUBROUTINE "SOLVE" =====

```

```

REM *** SUBROUTINE *****
PLOT:

REM === Erasing Old Lines & Points =====
LINE (Ax, Ay - 1)-(Bx, By - 1), 0: LINE (Ax, Ay)-(Bx, By), 0
LINE (Ax, Ay + 1)-(Bx, By + 1), 0: LINE (Bx, By - 1)-(Dx, Dy - 1), 0
LINE (Bx, By)-(Dx, Dy), 0: LINE (Bx, By + 1)-(Dx, Dy + 1), 0
LINE (Dx, Dy - 1)-(Ex, Ey - 1), 0: LINE (Dx, Dy)-(Ex, Ey), 0
LINE (Dx, Dy + 1)-(Ex, Ey + 1), 0: LINE (Bx, By)-(X, Y), 0
LINE (Dx, Dy)-(X, Y), 0: LINE (X - 4, Y - 4)-(X + 4, Y + 4), 0, BF
LINE (Bx - 3, By - 3)-(Bx + 3, By + 3), 0, BF
LINE (Dx - 3, Dy - 3)-(Dx + 3, Dy + 3), 0, BF

REM === Calculating Coordinates of New Points =====
Bx = L1 * 300 * COS(T1) + 230
By = -L1 * 300 * SIN(T1) + 240
Dx = -L3 * 300 * COS(T3) + 230 + L4 * 300
Dy = L3 * 300 * SIN(T3) + 240
XT = -L4 * TAN(T3) / (TAN(T1) - TAN(T3))
YT = -XT * TAN(T1)
X = XT * 300 + 230: Y = YT * 300 + 240
XS(S) = X: YS(S) = Y
XS(M) = .04590291# * 300 + 230: YS(M) = 0 * 300 + 240
XS(N) = -.56018862# * 300 + 230: YS(N) = 0 * 300 + 240

REM === Drawing New Points & Lines =====
FOR G = 1 TO I: CIRCLE (XS(G), YS(G)), 1, 14: NEXT
LINE (Bx, By)-(X, Y), 7: LINE (Dx, Dy)-(X, Y), 7
LINE (X, Y - 4)-(X, Y + 4), 14: LINE (X - 4, Y)-(X + 4, Y), 14
LINE (Ax, Ay - 1)-(Bx, By - 1), 12: LINE (Ax, Ay)-(Bx, By), 12
LINE (Ax, Ay + 1)-(Bx, By + 1), 12: LINE (Bx, By - 1)-(Dx, Dy - 1), 15
LINE (Bx, By)-(Dx, Dy), 15: LINE (Bx, By + 1)-(Dx, Dy + 1), 15
LINE (Dx, Dy - 1)-(Ex, Ey - 1), 9: LINE (Dx, Dy)-(Ex, Ey), 9
LINE (Dx, Dy + 1)-(Ex, Ey + 1), 9: LINE (0, Ay)-(640, Ay), 8

REM === Drawing the Fixed Support at A =====
CIRCLE (Ax, Ay), 2, 15: PAINT (Ax, Ay), 9, 15:
CIRCLE (Ax, Ay), 4, 15, 0, PI
LINE (Ax - 4, Ay)-(Ax - 4, Ay + 4), 15
LINE (Ax + 4, Ay)-(Ax + 4, Ay + 4), 15
LINE (Ax - 6, Ay + 4)-(Ax + 6, Ay + 4), 15
LOCATE 15, 27: COLOR 12: PRINT "A"

REM === Drawing the Fixed Support at E =====
CIRCLE (Ex, Ey), 2, 15: PAINT (Ex, Ey), 9, 15
CIRCLE (Ex, Ey), 4, 15, 0, PI
LINE (Ex - 4, Ey)-(Ex - 4, Ey + 4), 15
LINE (Ex + 4, Ey)-(Ex + 4, Ey + 4), 15
LINE (Ex - 6, Ey + 4)-(Ex + 6, Ey + 4), 15
LOCATE 17, 52: COLOR 12: PRINT "E"

REM === Drawing the Moving Joints at B & D =====
CIRCLE (Bx, By), 2, 15: PAINT (Bx, By), 9, 15
CIRCLE (Dx, Dy), 2, 15: PAINT (Dx, Dy), 9, 15

REM === Preparing for the Next Iteration =====
S = CINT(S) + 1: RETURN
REM *** LAST LINE OF PROGRAM =====

```

## B. Output of the QuickBASIC Program: Sample Pictures in the Animation

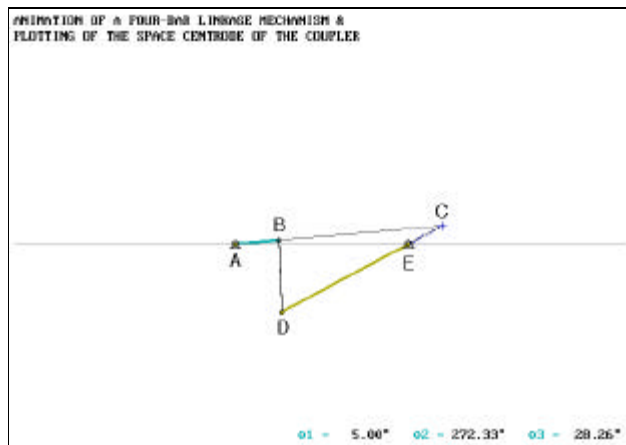
Upon running the preceding QuickBASIC program, the motion of the four-bar linkage mechanism and the plotting of the centrode of the coupler will be simulated on the computer screen. Each picture frame carries a heading in the upper left part of the frame, which says

ANIMATION OF A FOUR-BAR LINKAGE MECHANISM &  
PLOTTING OF THE SPACE CENTRODE OF THE COUPLER

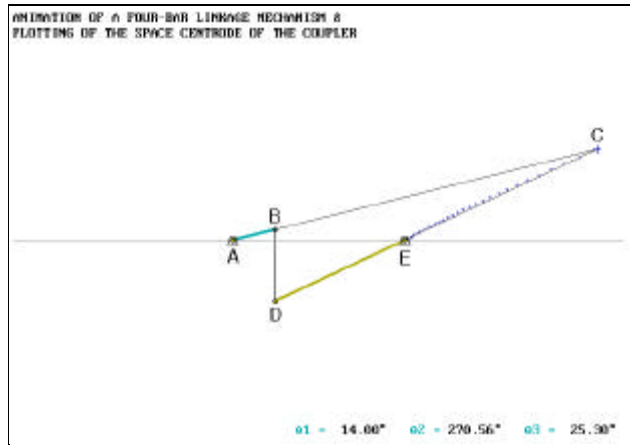
As the animation progresses, the current values of  $\theta_1$ ,  $\theta_2$ , and  $\theta_3$  are displayed in the lower right part of the frame for each step. For example, the lower right part of Picture (a) says that

$$\theta_1 = 5.00^\circ \quad \theta_2 = 272.33^\circ \quad \theta_3 = 28.26^\circ$$

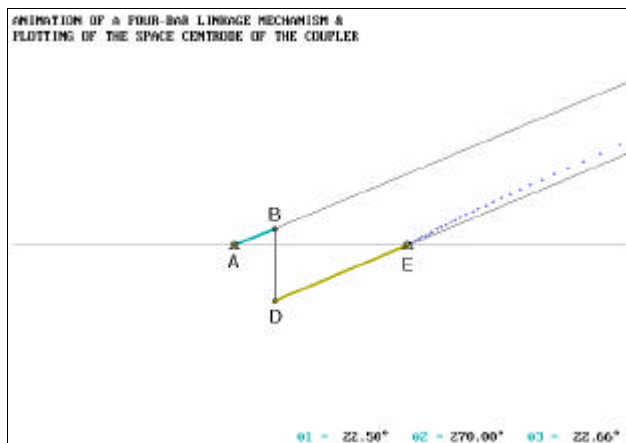
Although only the points  $A$  and  $E$  are automatically labeled, the points  $A$ ,  $B$ ,  $C$ ,  $D$ , and  $E$  on the following captured pictures are manually re-labeled in this presentation. A sequence of sample pictures on the computer screen is shown in Pictures (a) through (r).



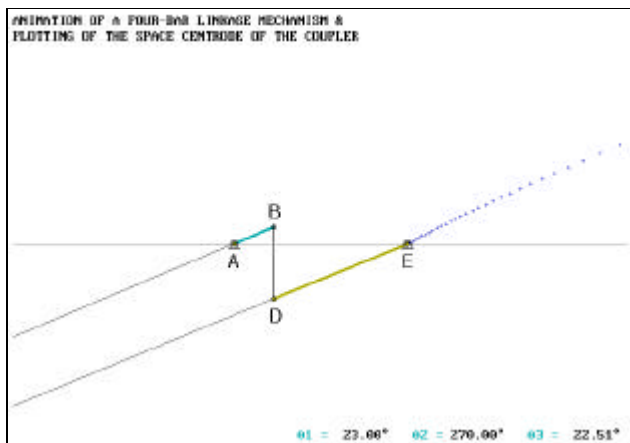
Picture (a)  $\theta_1 = 5^\circ$



Picture (b)  $\theta_1 = 14^\circ$

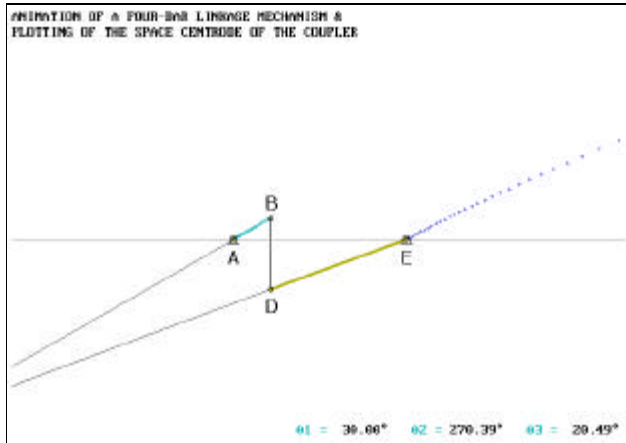


Picture (c)  $\theta_1 = 22.5^\circ$

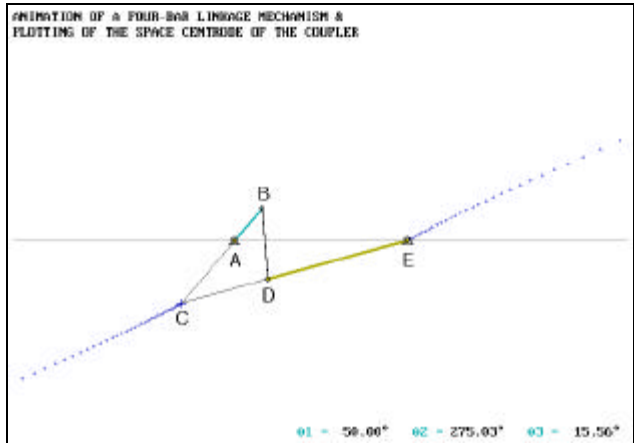


Picture (d)  $\theta_1 = 23^\circ$

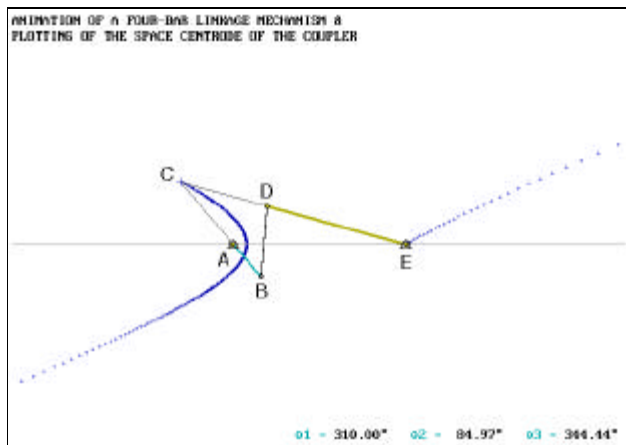
Sample Pictures in the Animation (Continued)



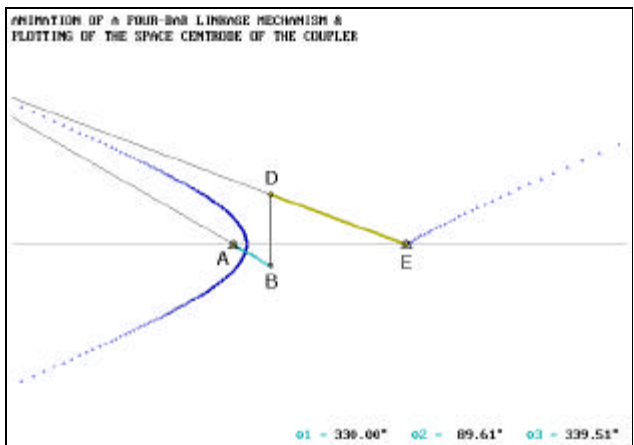
Picture (e)  $\theta_1 = 30^\circ$



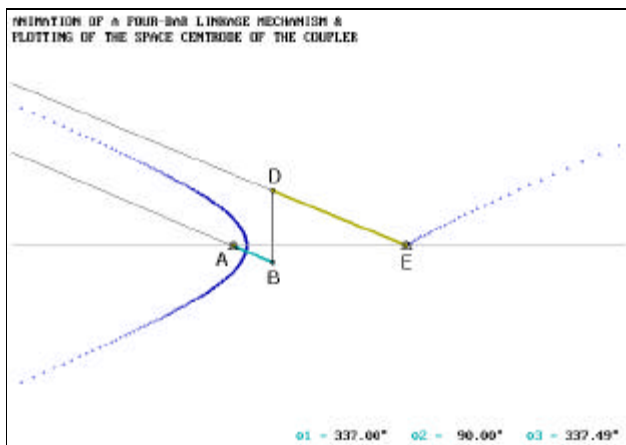
Picture (f)  $\theta_1 = 50^\circ$



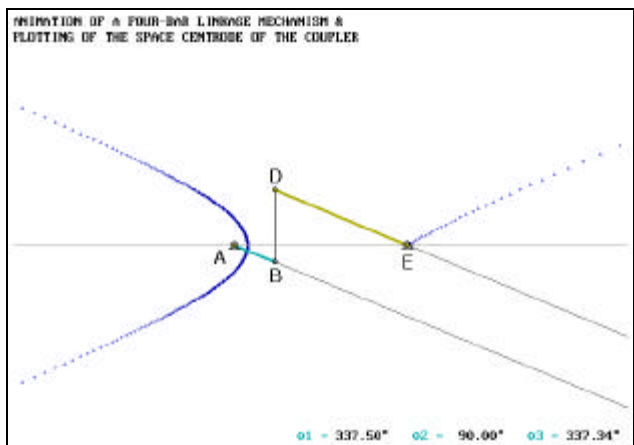
Picture (g)  $\theta_1 = 310^\circ$



Picture (h)  $\theta_1 = 330^\circ$

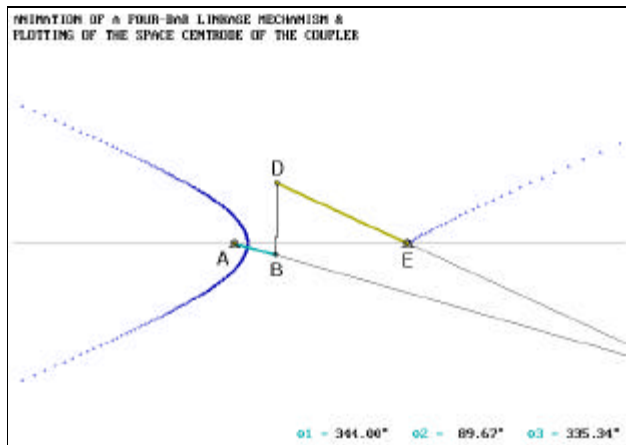


Picture (i)  $\theta_1 = 337^\circ$

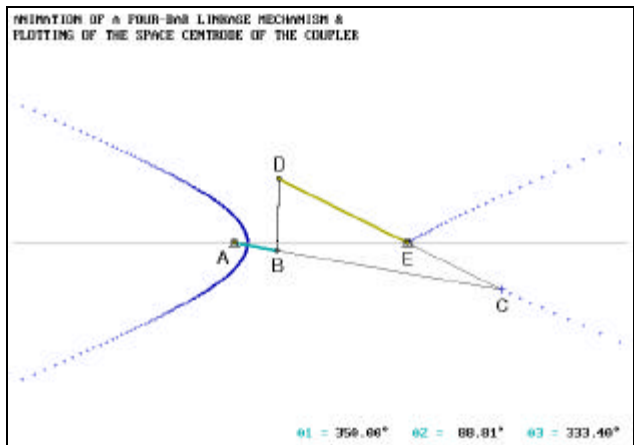


Picture (j)  $\theta_1 = 337.5^\circ$

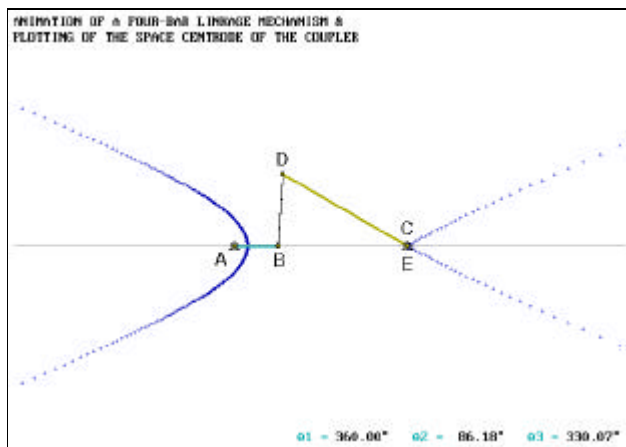
Sample Pictures in the Animation (Continued)



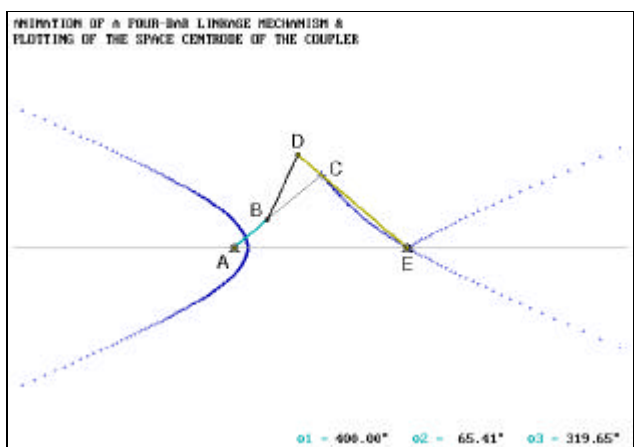
Picture (k)  $\theta_1 = 344^\circ$



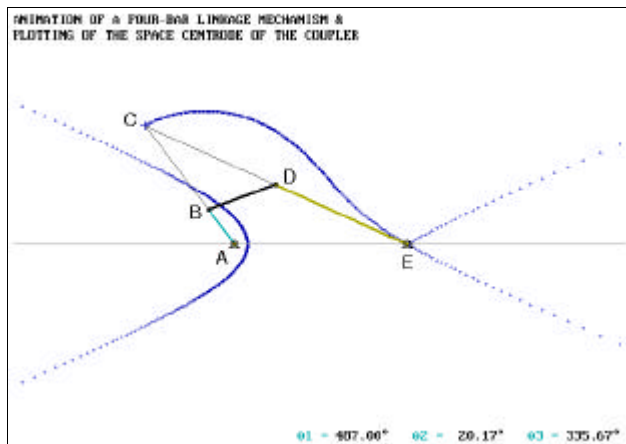
Picture (l)  $\theta_1 = 350^\circ$



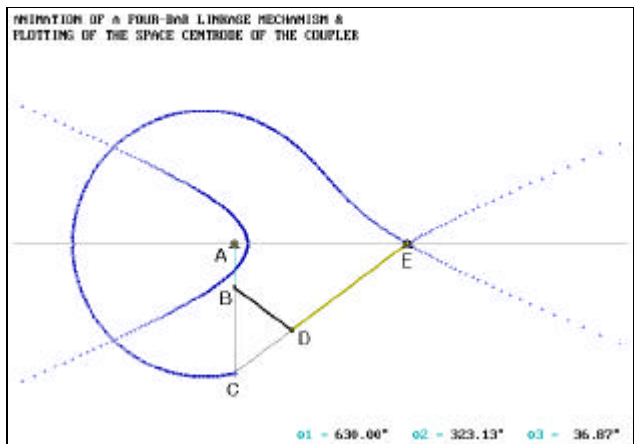
Picture (m)  $\theta_1 = 360^\circ$



Picture (n)  $\theta_1 = 400^\circ$



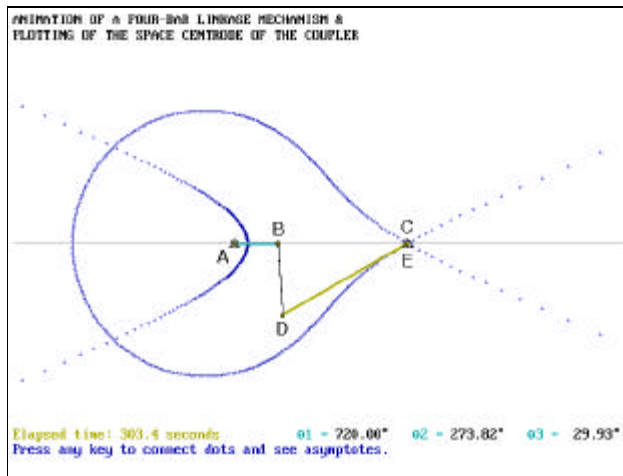
Picture (o)  $\theta_1 = 487^\circ$



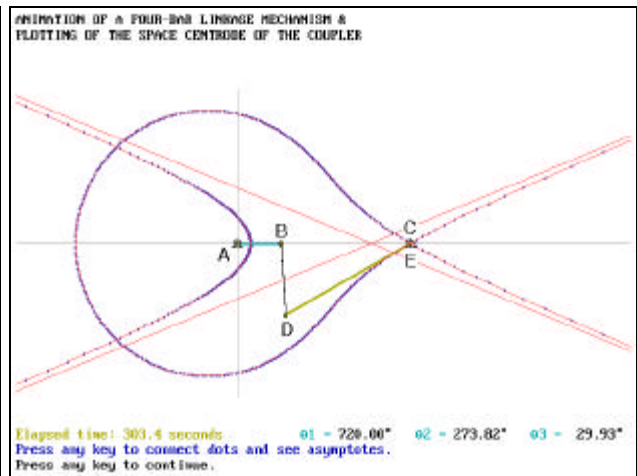
Picture (p)  $\theta_1 = 630^\circ$



## Sample Pictures in the Animation (Continued)



Picture (q)  $\theta_1 = 720^\circ$



Picture (r)  $\theta_1 = 720^\circ$  and asymptotes

### Salient Features of Space Centrode

The preceding sequence of sample pictures in the output of the QuickBASIC program reveals several salient features. The jumps and meander in the locus of the velocity center  $C$  of the coupler are indeed spectacular. It is of interest to point out the following salient features:

- As the crank  $AB$  rotates counterclockwise from the position  $\theta_1 = 0^\circ$ ,  $C$  starts to move out from the support at  $E$ .
- For  $0^\circ < \theta_1 < 22.62^\circ$ ,  $C$  moves from  $E$  to travel in the upper right direction to infinity in the first quadrant. See Pictures (a) through (c).
- As the crank passes the position  $\theta_1 = 22.62^\circ$ ,  $C$  jumps from infinity in the first quadrant down to infinity in the third quadrant. See Pictures (c) and (d).
- For  $22.62^\circ < \theta_1 < 337.38^\circ$ ,  $C$  moves from infinity in the third quadrant to traverse the space centrode to finally approach infinity in the second quadrant. See Pictures (d) through (i).
- As the crank passes the position  $\theta_1 = 337.38^\circ$ ,  $C$  jumps from infinity in the second quadrant down to infinity in the fourth quadrant. See Pictures (i) and (j).
- For  $337.38^\circ < \theta_1 < 360^\circ$ ,  $C$  moves from infinity in the fourth quadrant to travel in the upper left direction to the support at  $E$ . See Pictures (j) through (m).
- For  $360^\circ < \theta_1 < 720^\circ$  (i.e., in the second revolution of the crank),  $C$  moves from the support at  $E$  to traverse the space centrode in the counterclockwise direction to return to the support at  $E$ . See Pictures (m) through (q).
- The asymptotes of the space centrode are shown in Picture (r).
- For  $\theta_1 > 720^\circ$ , the above features will be repeated for every two revolutions of the crank  $AB$ .

## Concluding Remarks

Textbooks and instructors are indispensable in the education of engineering students. However, appropriate animation of bodies and mechanisms can greatly enhance engineering students' understanding in their learning of technical subjects, such as dynamics that deals with motions of bodies and mechanisms. This paper is intended to share some programming techniques and ideas with instructors who may contemplate on writing their own programs to complement available canned animation software that does not lend itself to animating a desired specific task. QuickBASIC has the built-in strength in graphics and is easy to program, very flexible, low overhead, and widely available; it is naturally an attractive choice to complement available canned animation software.

Surely, the animation of mechanisms can be accomplished using higher-level programming languages and more automatic features provided in other more advanced software packages, such as MATLAB and MATHEMATICA.<sup>5,6</sup> Nevertheless, QuickBASIC requires less execution time and has fewer constraints imposed by the built-in automatic features in the more advanced software packages. As a result, QuickBASIC allows some advantages, such as less total memory required, less throughput time in the computation and animation, and more flexibility in designing the layout (e.g., positioning titles on graphics screens or inserting special characters).

## References

1. Chapra, S. C. and Canale, R. P., *Numerical Methods for Engineers*, Second Edition, McGraw-Hill Book Company, 1988, pp. 167-168.
2. Jong, I. C., Reynolds, R. R., and Adams, G. P., "Determination of Space Centrode of a Coupler Link," Session 2668, *1996 ASEE Annual Conference Proceedings*, Washington, D.C.
3. *Working Model*, Version 2.0.2, Knowledge Revolution, Inc., Tel. (800) 766-6615, San Mateo, CA, 1989-1994.
4. The Waite Group, *Microsoft QuickBASIC Bible*, Microsoft Press, One Microsoft Way, Richmond, WA, 1990.
5. Adams, G. P. and Jong, I. C., "Using MATLAB to Animate the Generation of a Space Centrode in Kinematics," Session 2668, *1997 ASEE Annual Conference Proceedings*, Milwaukee, WI.
6. Reynolds, R. R. and Jong, I. C., "Using MATHEMATICA to Animate the Generation of a Space Centrode in Kinematics," Session 1668, *1997 ASEE Annual Conference Proceedings*, Milwaukee, WI.

### I. C. Jong

Ing-Chang Jong received a BSCE from the National Taiwan University in 1961, an MSCE from the SDSM&T in 1963, and a Ph.D. from Northwestern University in 1965. He is a Professor of Mechanical Engineering at the University of Arkansas. He and Dr. Bruce G. Rogers published an engineering mechanics textbook in 1991. He is serving as the Chair of the Mechanics Division, ASEE, 1996-97

### S. N. Onggowijaya

Stanley N. Onggowijaya received a BSME *cum laude* from the University of Arkansas in 1996. He is currently a graduate student studying toward an MBA degree and is a technology support staff in the College of Business Administration, University of Arkansas. His areas of interest include computer software applications and analyses for PC's and Macintoshes.

Tutorials teaching QBasic / QuickBasic graphics techniques, as well as the theory and principles of graphic design. Categories: Beginners. Sprites. Palette and Colors. Animation. 3D Graphics. 2D Graphical Manipulation / Effects. An overview of graphics programming in QB that covers some lesser-known techniques like using multiple pages in SCREEN 7, 8 and 9. However, this tutorial is hardly "advanced". Jonathon. Screen 13 Graphics. QBasic as well as QuickBasic is an easy-to-learn programming language (and therefore ideal for beginners), based on DOS operating system, but also executable on Windows. Differences between QBasic and QuickBasic: QBasic is the slimmed-down version of QuickBasic. Compared to QuickBasic, QBasic is limited as it lacks a compiler. Therefore QBasic cannot be used to produce executables (.exe files). The source code (usually files with .bas extension) can only be executed immediately by the built-in QBasic interpreter. Furthermore, QuickBasic has a more extensive command set than QBasic. Offers on Q... Teaching aids which are at teachers' disposal in contemporary schools may be grouped into (1) non-mechanical aids and (2) mechanical aids. Programming allows the teacher to improve the effectiveness of teaching by constructing materials which will guide the pupil through a series of steps towards the mastery of a learning problem. These steps should be of appropriate size and require the pupil's active cooperation; he may be asked to answer a question, to fill in a blank, to read, etc. It is very important to grade progress of steps throughout the programme so carefully that each pupil get every step right. Media of programmed instruction are programmed lessons or textbooks and teaching machines. There are at least t