USyCaMs: A Software Package for the Interactive Synthesis of Cam Mechanisms

Max A. González-Palacios
Universidad Iberoamericana-León, Departamento de Ingeniería
Lib. Nte. Km. 3, Apdo. Postal 1-26, León, Gto. México 37000
email: maxg@bach.ing.leon.uia.mx

and

Jorge Angeles
McGill Research Centre for Intelligent Machines &
Department of Mechanical Engineering, McGill University
817 Sherbrooke Street West, Montreal, Canada H3A 2K6
email: angeles@cim.mcgill.ca

Résumé. Les auteurs donnent une description complète du logiciel de conception mécanique *USyCaMs* (Unified Synthesis of Cam Mechanisms), mis au point sur matériel infographique très puissant. Cet ensemble logiciel-matériel permet de visualiser, par animation, les mouvements du modèle de corps solide du mécanisme en question. Il permet, en outre, d'observer, en temps réel, les transformations des composants, à savoir, la came, son maillon entraîné et, le cas échéant, le roulement, lorsque les paramètres réels ou discrets, définissant géométriquement le mécanisme changent de façon continue ou discrète, sans interrompre le mouvement du mécanisme.

Abstract. Introduced here is a detailed description of *USyCaMs* (Unified Synthesis of Cam Mechanisms), with a powerful graphics support, which allows the user to visualize the animated solid model of a mechanism undergoing the required motion. Moreover, it is possible to observe in real time the transformation of the components, i.e., cam, follower, and roller-follower, upon changing *continuously* the *real-valued* parameters or *discretely* the integer-valued parameters defining the mechanism geometry, *without interrupting the motion of the mechanism.*

v

1 Introduction

In the last twenty years or so, computer graphics has become a fundamental tool for the study of mechanisms, as made apparent by the software available in this area. We can cite IMP (Integrated Mechanisms Program) as one of the earliest software packages developed for the analysis of arbitrary linkages [ShU72]. A few years later KINSIN III, a package meant for the synthesis of mechanisms, was introduced [RuK77]. The graphical part of these early packages consisted essentially of 2-D lines. Further mechanism software packages, with improvements in their graphics, have been produced, such as LINCAGES [ErG77], MINN-DWELL [KER87] and SIXPAQ [GoA91].

Given that modern workstations provide a highly integrated environment for computation and interaction, our goal in the development of USyCaMs was that it should be intuitively easy to use and as general as possible. Following these criteria, the user of USyCaMs need not be a CAD expert, although a familiarity with the theory of cam mechanisms is expected, in order to better exploit the capabilities of USyCaMs. Solving complex problems, like undercutting, with visualization aids is reduced to moving the mouse around the appropriate menus.

USyCaMs has many applications. It can be used to give an introduction to cam mechanisms in an undergraduate course, or to solve complex design problems involving, e.g., undercutting of spatial cam mechanisms, in a graduate design course. Furthermore, USyCaMs can be regarded as a sophisticated design tool for industrial applications involving dimensioning, balancing, dynamic analysis, simulation and finite-element analysis (FEA) for stress, strain and thermal calculations. Note that USyCaMs provides a database for useful mesh generation in FEA, but is limited to the handling and production of geometric and kinematic information.

2 Kinematics of Cam Mechanisms

The synthesis procedure is based on the minimization of power losses, which is achieved, in turn, by minimizing the magnitude of the sliding velocity along the contact surfaces. To this end, the contact surfaces are designed as ruled surfaces, and motion is transmitted along a common line, the contact generatrix, which gives rise to a higher kinematic pair. Two more kinematic pairs arise, namely, the cam-frame and the follower-frame pairs, which belong to the class of lower kinematic pairs, and can be either revolute or prismatic [Ang82]. USyCaMs thus allows the synthesis of cam mechanisms not only with rotating, but also with translating cams or followers.

One objective of USyCaMs is to synthesize the contact surfaces of all the elements involved for two cases: (a) mechanisms comprising the frame, the cam, and the follower, henceforth termed *three-link mechanisms*; and (b) mechanisms similar to the former, but with an intermediate fourth element, the roller, henceforth termed *four-link mechanisms*. As a matter of fact, we are here following the established terminology in the realm of cam mechanisms, but, properly speaking, the intermediate element between cam and follower is not always a 'roller'. Indeed, when minimizing the magnitude of the sliding velocity between cam and 'roller' and between 'roller' and follower, the 'roller' turns out to be a hyperboloid of revolution that both slides and rotates with respect to the cam and the follower, in the most general case in which the axes of rotation of the cam and the follower are skew. In this case, we have a *spatial mechanism*. If the same axes intersect, we have a *spherical mechanism* and the roller takes the form of a cone of revolution and rotates without slipping about both the cam and the follower. Therefore, in the case of three-link mechanisms, two ruled surfaces are synthesized, while three are synthesized in the case of four-link mechanisms. The shape of the foregoing surfaces is determined so as to produce a given *input-output function* between cam and follower.

3 Software Description

We give an outline of USyCaMs for a *Silicon Graphics Inc. IRIS* workstation. It can run on other *UNIX* workstations with suitable graphics software and hardware, but then, obviously, all device-dependent features must be modified.

At the outset, we divide the window into five *viewports*, VP1, VP2, VP3, VP4 and VP5, as shown in Fig. 1. Each of these viewports serves a specific function:

VP1: This part of the screen is devoted to the rendering of both the still solid models and the animation of the motion of the mechanism under design.

VP2: This viewport displays the main menu in two modes, namely, passive and active. In passive mode, VP2 tells the user what kind of mechanism is in the process of synthesis; in active mode, the user can interact with the program and choose the type of mechanism desired.

VP3: This viewport shows the design parameters and interacts simultaneously with VP1, so that a change in any of the mechanism parameters is reflected in the solid models of the mechanism.

VP4: As VP3, VP4 interacts with VP1; this viewport shows the parameters pertaining to the input-output function.

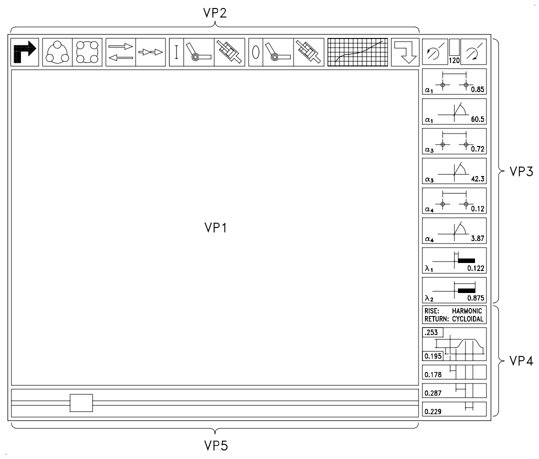


Figure 1: *USyCaMs interface window.*

VP5: If at least one of the parameters of VP3 or VP4 is active, a sliding bar appears in VP5, so that the user can modify continuously the parameter values by moving the cursor, with the aid of the mouse, along this bar.

3.1 Iconography

The main menu is iconized and displayed in VP2 as shown in Fig. 2; the user can choose the type of mechanism to be synthesized by clicking these icons with the mouse. Thus, VP2 is divided into seven sections called POSi, for $i = 0, 1, 2, \cdots, 6$, as described below:

POS0: When this area is selected, VP2 changes from passive mode to active mode.

POS1: The mechanism can comprise either three links or four links. The option selected is highlighted to let the user know which one is active.

POS2: Similar to POS1, except that, in this case, the user chooses either an oscillating or an indexing mechanism.

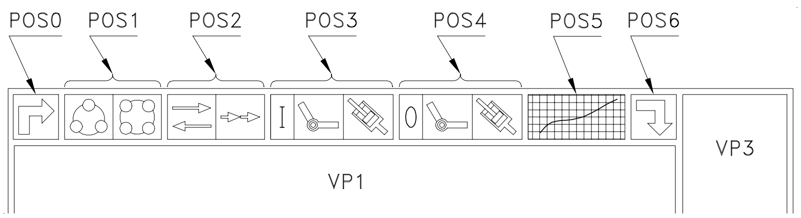


Figure 2: *Icons of the main menu.*

POS3: The input kinematic pair can be revolute or prismatic; the user selects the type here.

POS4: Similar to POS3, but, in this case, the kinematic pair is chosen for the output.

POS5: This icon is used to define the input-output function with the aid of a submenu displayed in VP4.

POS6: The user can exit VP2 any time by selecting this icon, VP2 thus switching into passive mode.

In order to identify the type of mechanism under synthesis, we introduce a suitable labelling, namely, `x_xxx_xx`, the meaning of each of the three fields being

`x`: The number of links of the mechanism, three (`3`) or four (`4`).

`xxx`: The type of the follower motion, indexing (`ind`) or oscillating (`osc`).

`xx`: Two characters are reserved for this field, `p` and `r`, which stand for prismatic and revolute, respectively. Thus, if we read from left to right, the first character tells us the input pair, while the second, the output pair.

For example, `4_osc_rr` indicates a four-link mechanism with oscillating follower, revolute input and revolute output.

The two square icons at the top of **VP3** allow the user to animate the motion of the mechanism. Here, the user can choose the sense of the input motion. With the six icons below these two, it

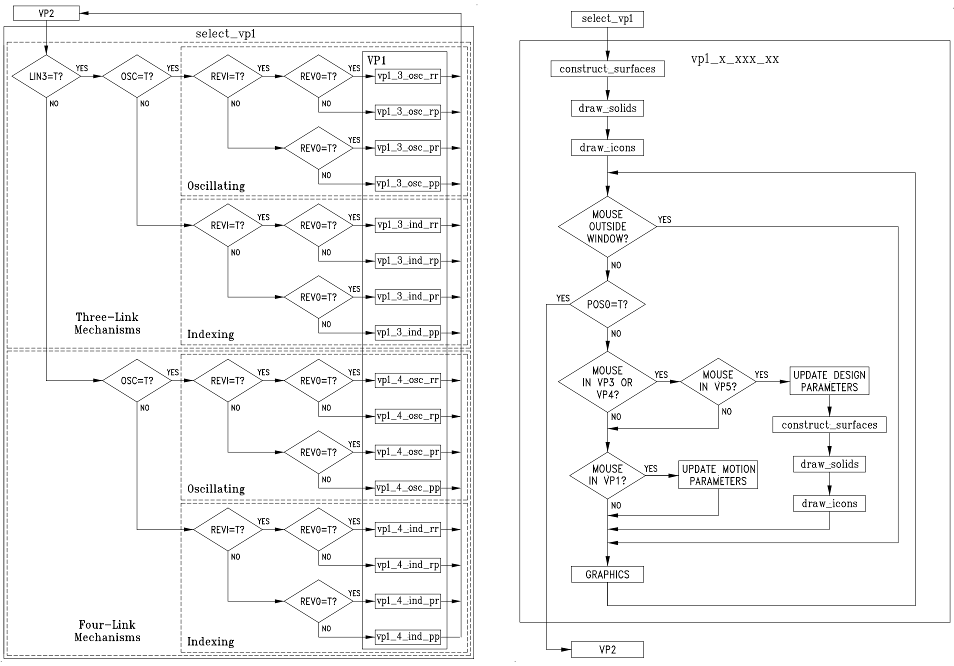
Type of Mechanism v	VP3							
	a_1	α_1	a_3	α_3	a_4	α_4	λ_1	λ_2
`3_ind_pp`								
`3_ind_pr`	×	×					×	×
`3_ind_rp`	×	×					×	×
`3_ind_rr`	×	×					×	×
`4_ind_pp` and `4_osc_pp`		×	×		×		×	×
`4_ind_pr` and `4_osc_pr`	×	×	×		×		×	×
`4_ind_rp` and `4_osc_rp`	×	×	×		×		×	×
`4_ind_rr` and `4_osc_rr`	×	×	×	×	×	×	×	×
`3_osc_pp`	×				×		×	×
`3_osc_pr`	×	×			×		×	×
`3_osc_rp`	×	×			×		×	×
`3_osc_rr`	×	×			×	×	×	×

Table 1: *Design Parameters of VP3*

is possible to modify the distance and the angle of three pairs of axes, namely,

1. The distance a_1 between the input and output axes;
2. the angle α_1 between the above two axes;
3. the distance a_3 between the output and roller axes;
4. the angle α_3 between the above two axes;
5. the distance a_4 between the roller axis and its generatrix; and
6. the angle α_4 between the above two axes.

With the eighth and ninth icons of **VP3**, the user can modify the thickness of the contact surfaces, λ_1, λ_2, which do not affect the kinematics of the mechanism, but have to be specified for manufacturing purposes. The design parameters vary depending on the type of mechanism selected. The parameters pertaining to a given mechanism type are marked with $\times$ in Table 1.



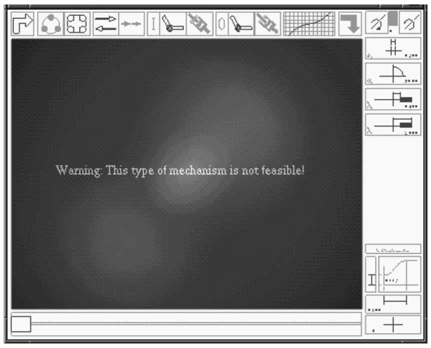
(a) (b)

Figure 3: *Flowchart of* a) *the* `select_vp1` *procedure;* b) *VP1 procedures.*

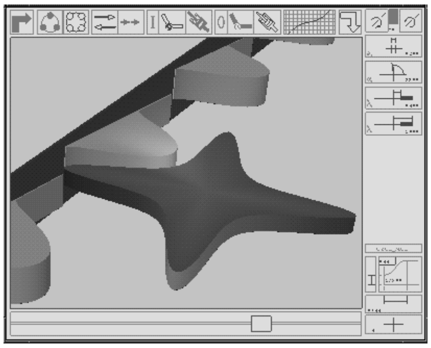
3.2 Main Loop

As mentioned above, viewport VP1 is used to display the solid model representation of the mechanism selected in VP2. If we look at Fig. 3*a*, we notice that there are up to sixteen different types of mechanisms with independent *synthesis procedures*, which are identified with the labels shown at the right-hand side of Fig. 3*a*. The flowchart sample of these procedures is shown in Fig. 3*b*. In the synthesis of one of the sixteen mechanisms the user will be working most of the time in its corresponding procedure, which is the reason why we call it the main loop.

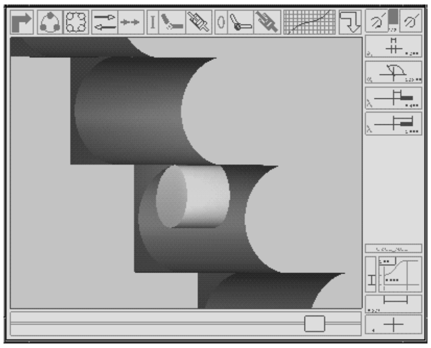
All sixteen procedures have similar structures; what changes in each case is only the synthesis algorithm, `construct_surfaces`, the procedures to generate the solid models of cam and follower, `draw_solids`, and the procedures to draw the icons of VP3 and VP4, namely, `draw_icons`. At the main-loop level, USyCaMs performs the algorithm of the flowchart of Fig. 3*b*.



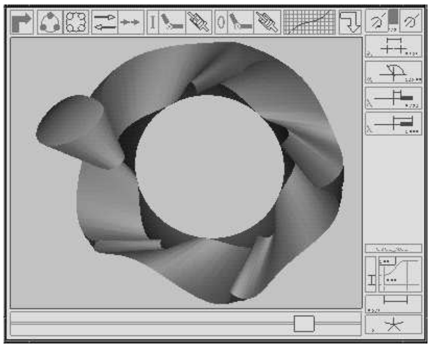
(a) `3_ind_pp`



(b) `3_ind_pr`



(c) `3_ind_rp`



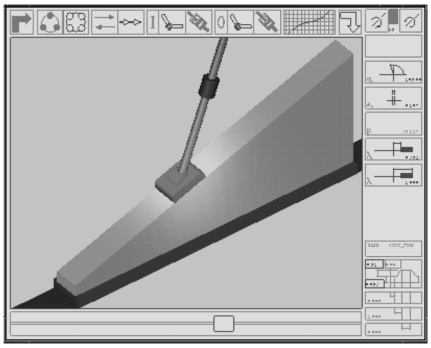
(d) `3_ind_rr`

Figure 4: *Three-link indexing cam mechanisms.*

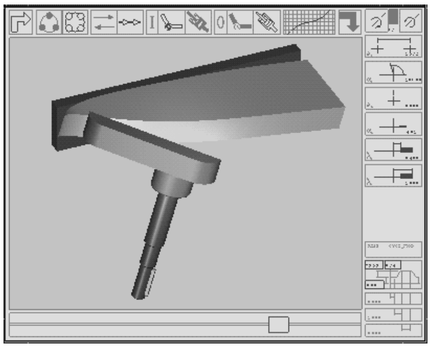
4 Rendered Examples

In this section we display still frames for each one of the cases mentioned above. Thus, in Fig 4 we include the possible solutions of three-link indexing mechanisms, one of them being unfeasible [GoA93].

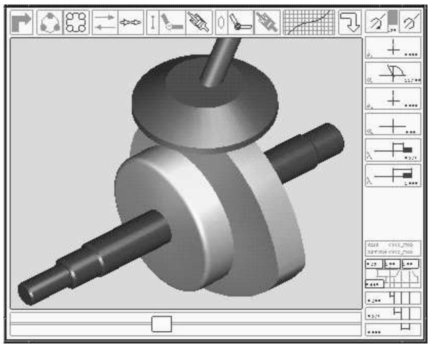
Three-link oscillating mechanisms with constant pressure angle are shown in Fig. 5. If we set $\alpha_1 = \pi/2$ in the mechanism shown in Fig. 5*c*, the solution is well known as the *cam mechanism with translating flat-face follower*.



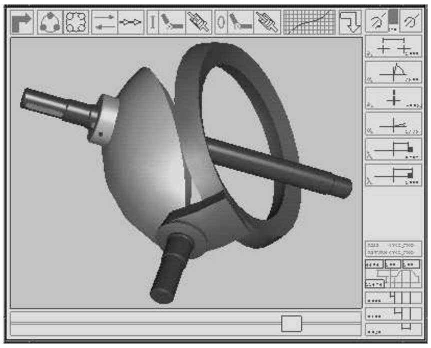
(a) `3_osc_pp`



(b) `3_osc_pr`



(c) `3_osc_rp`



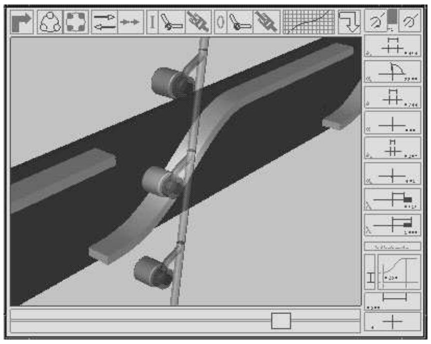
(d) `3_osc_rr`

Figure 5: *Three-link oscillating cam mechanisms.*

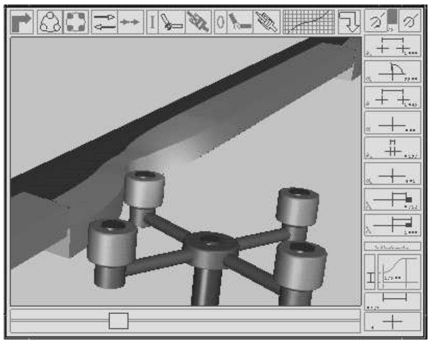
Shown in Fig. 6 are the solutions of four-link indexing mechanisms, while, in Fig. 7, the solutions of four-link oscillating mechanisms. If we set $\alpha_1 = \pi/2$ in the mechanism shown in Fig. 7*c*, the resulting mechanism is known as the *cam mechanism with translating roller-follower*. The spherical counterpart of the planar cam mechanism with oscillating roller-follower is shown in Fig. 7*d*. The latter can be obtained by changing the design parameters.

5 Conclusions

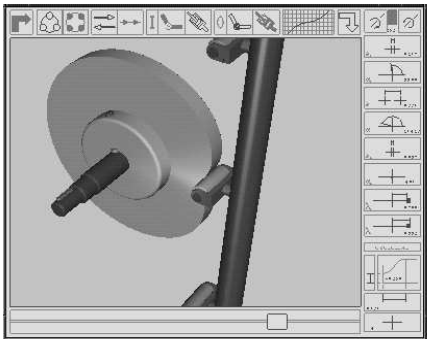
The first version of USyCaMs has been completed according to the guidelines and objectives outlined in [GoA93]. This version, called USyCaMs 1.0, was coded in C and supported by the graphics library



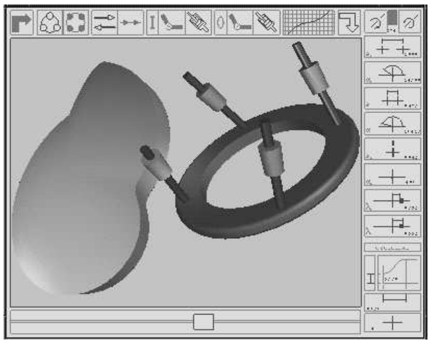
(a) `4_ind_pp`



(b) `4_ind_pr`



(c) `4_ind_rp`



(d) `4_ind_rr`

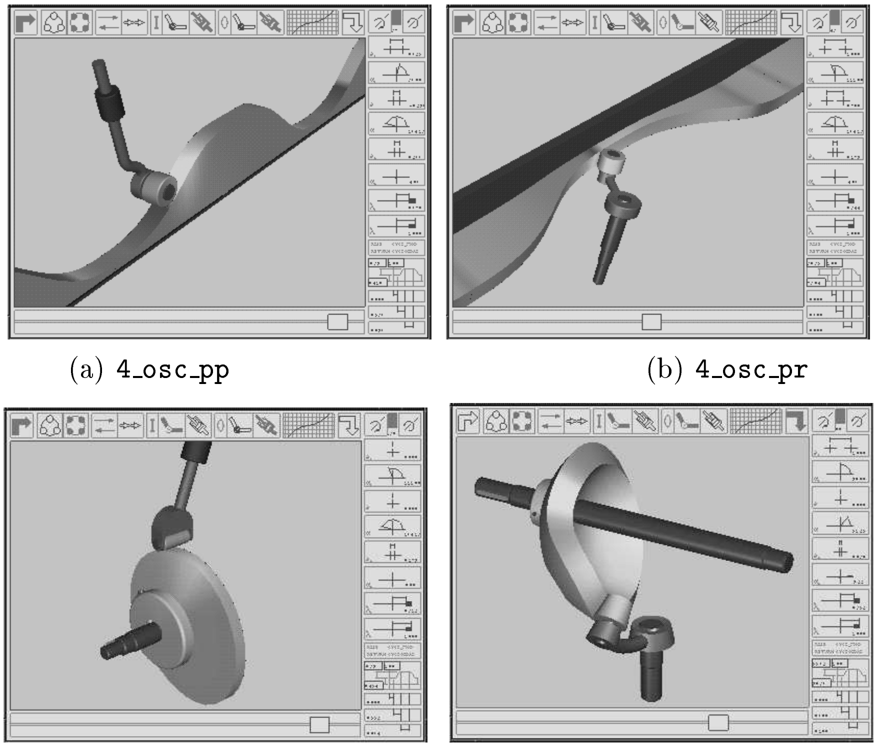
Figure 6: *Four-link indexing cam mechanisms.*

GL.

Thanks to the graphical potential of USyCaMs, it has been possible to design mechanisms never conceptualized before. In this way, we have been able to develop innovative transmission systems; such is the case of PRICAM (Pure Rolling Indexing Cam Mechanism), whose prototypes were fully designed on the monitor, and then, built in two versions, planar and spherical.

6 Acknowledgements

The research of the first author was supported by Universidad Iberoamericana at León, by Mexico's *Sistema Nacional de Investigadores* Program, and by the mexican Company *Manufacturas Di-*



(a) `4_osc_pp` (b) `4_osc_pr`

(c) `4_osc_rp` (d) `4_osc_rr`

Figure 7: *Four-link oscillating cam mechanisms.*

versas S.A. The second author acknowledges support from Canada's Natural Sciences and Engineering Research Council.

7 References

[Ang82] J. Angeles, 1982, *Spatial Kinematic Chains*, Springer-Verlag, Berlin.

[ErG77] A. G. Erdman and J. E. Gustafson, 1977, "LINCAGES: Linkage INteractive Computer Analysis and Graphically Enhanced Synthesis Package," *ASME Design Engineering Technical Conference*, Chicago, Illinois, Paper No. 77–DET–5.

[GoA91] M. A. González-Palacios and J. Angeles, 1991, “SIX-PAQ: A Comprehensive Software Package for Analysis and Synthesis of Six-Bar Dwell Linkages”, *Proc. 1991 ASME Computers in Engineering*, Santa Clara, Cal., Vol. 1, pp. 309–314.

[GoA93] M. A. González-Palacios and J. Angeles, 1993, *Cam Synthesis*, Kluwer Academic Publishers, Dordrecht.

[KER87] S. Kota, A. G. Erdman and D. R. Riley, 1987, “MINN-DWELL – Computer Aided Design and Analysis of Linkage-Type Dwell Mechanisms”, *Proc. ASME International Computers in Engineering Conference*, Vol. 2, New York, pp. 351–358.

[RuK77] A. J. Rubel and R. E. Kaufman, 1977, “KINSYN III: A New Human-Engineered System for Interactive Computer-Aided Design of Planar Linkages”, *Journal of Engineering for Industry, Trans. ASME*, Vol. 99, pp. 440–448.

[ShU72] P. N. Sheth and J. J. Uicker, 1972, “IMP (Integrated Mechanism Program), A Computer-Aided Design Analysis System for Mechanisms and Linkages”, *Journal of Engineering for Industry, Trans. ASME*, Vol. 94, pp. 454–464.