USyCaMs: A Windows® software package for the synthesis of cam mechanisms

Max Antonio González-Palacios, Ricardo García-García.

Departamento de Ingeniería Mecánica, Universidad de Guanajuato,
División de Ingenierías Campus Irapuato-Salamanca DICIS
Carretera Salamanca–Valle de Santiago km. 3.5 + 1.8 km, Comunidad de Palo Blanco, Salamanca, Gto., MEXICO

maxg@ugto.mx, r.garcia.garcia@ugto.mx

RESUMEN

Hoy en día existen una variedad de programas CAD que pueden realizar el diseño y simulación de mecanismos de leva fácilmente. La mayoría de ellos sólo lo hacen para mecanismos de leva planos, y algunos otros se extienden a mecanismos de levas cilíndricas. En este artículo proponemos una versión Windows® del paquete de software USyCaMs para la síntesis de mecanismos de leva planos, esféricos y espaciales. El software propuesto está basado en una teoría novedosa y unificada, lo que permite al usuario sintetizar, en un mismo marco teórico, alguno de estos tres tipos de mecanismos de leva. Además, ha sido desarrollado de tal manera que el usuario puede sintetizar y simular en tiempo real el comportamiento cinemático de alguno de estos mecanismos. Con esto, se dispone a los investigadores y diseñadores mecánicos de una herramienta poderosa, interactiva y actualizada para la síntesis de los diferentes tipos de mecanismos de leva ya mencionados.

ABSTRACT

Nowadays there are a variety of CAD programs that can make the design and simulation of cam mechanisms easily. Most of them just do it for flat cam mechanisms, and some others extend to cylindrical cam mechanisms. In this paper we propose a Windows® version of the software package USyCaMs for the synthesis of flat, spherical and spatial cam mechanisms. The proposed software is based on a novel and unified theory, allowing the user to synthesize, in a single framework, any of these three types of cam mechanisms. Also, has been developed in such a way that the user can synthesize and simulate real-time kinematic behavior of any of these mechanisms. With this, we provide to researchers and mechanical designers of a powerful, interactive and updated tool for the synthesis of the different types of cam mechanisms mentioned above.

INTRODUCTION

The proposed software is user friendly software to plot different types of cam mechanisms on the screen, having parallel axes, intersecting axes and spatial axes, with a powerful graphics support in order to perform real-time animation. To build our perspective of software packages that can synthesize cam mechanisms, we present a brief description of commercial CAD packages having similar capabilities compared to USyCaMs.

SolidWorks® [1], is a parametric 3D modeling program focused on mechanical design. Within their capabilities exists some tools that assist the user in creating flat, cylindrical and linear cams.

CATIA® [2], is a CAD/CAM/CAE computer program with parametric approach, and is intended to manage the lifecycle of a product. The software tools help the user to develop flat, linear and barrel cams.

AutoCAD Mechanical® [3], is a software application for both 2D and 3D design, includes several tools for mechanical design, and has support to design flat cams.

THEORETICAL BASIS

A. Problem Definition

USyCaMs (Unified Synthesis of Cam Mechanisms) arose from the need to interpret the results obtained of a kinematic mechanism...
problem, which consisted of determining the mathematical formulation that describes the surfaces of two rigid bodies, being capable of relative motion transmitted by direct contact when these two are connected to a third party by 'axes' whose position in space is arbitrary, with the only constraint that the transmission is performed with minimum slip.

The cam mechanisms study presented in USyCaMs, is based on minimizing friction losses, which results in that the contact between the surfaces that transmit the movement is a line, which is called instantaneous screw axis ISA. This line generates in turn a ruled surface, which when formed around the ISA which relates the input link determine the cam, whereas when is formed around the ISA that relates the output link determines the follower.

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. Here, due of the extension of the theory, only are presented the vectors to create the ruled surfaces of the cam ($r_2(t)$) and the follower ($r_3(t)$) of three-link cam mechanisms. The complete formulation to determine the ruled surfaces of three-link cam mechanisms and four-link cam mechanisms is found in [4].

$$
\begin{align*}
\mathbf{r}_2(t, \lambda) &= \begin{bmatrix}
    b_2 c \psi(t) - z_2(t) s \psi(t) s \theta_2 c \theta_2 \\
    -b_2 s \psi(t) - z_2(t) c \psi(t) s \theta_2 c \theta_2 \\
    -z_2(t) s^2 \theta_2 \\
    -s \psi(t) s \theta_2 \\
    -c \psi(t) s \theta_2 \\
    c \theta_2
\end{bmatrix} \\
\mathbf{r}_3(t, \lambda) &= \begin{bmatrix}
    d c \phi(t) - z_3(t) s \phi(t) s \beta c \beta \\
    -d s \phi(t) - z_3(t) c \phi(t) s \beta c \beta \\
    -z_3(t) s^2 \beta \\
    s \phi(t) s \beta \\
    -c \phi(t) s \beta \\
    c \beta
\end{bmatrix}
\end{align*}
$$

where $d = b_2 - a_1$, $\beta = \theta_2 - \alpha_1$, $c$ stands for cosine and $s$ stands for sine.

USyCaMs is based on (1, 2 and the analogous equations for the four-link mechanisms, in order to generate the ruled surfaces and display them on the screen.

C. Generalized Input-Output Function

The dimensions and symbols of the variables of the input-output functions change according to the type of kinematic pair of the mechanism to be considered. Two kinds of pairs have been assumed either for the input or the output motions, namely, revolute and prismatic. Therefore, a total of four combinations are achieved, RR, RP, PR and PP, which are applicable to both three and four-link cam mechanisms. However, all of them can be considered as one generalized input-output function, showed in (3).
\[ \varphi(x) = h \tau(x) \quad 0 \leq \tau \leq 1, \quad 0 \leq x \leq 1 \] (3)

where \( x \) and \( \tau \) denote the normalized variables of input and output, respectively, \( h \) represents the rise of the follower. In this way are defined in Table 1 \( \varphi \), \( h \), and \( x \) for each of the four types mentioned above.

<table>
<thead>
<tr>
<th>Type</th>
<th>( \varphi )</th>
<th>( h )</th>
<th>( x )</th>
<th>( x' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR</td>
<td>( \phi(\psi) )</td>
<td>( \Delta \phi )</td>
<td>( \psi/\Delta \psi )</td>
<td>( dx/d\psi = 1/\Delta \phi )</td>
</tr>
<tr>
<td>RP</td>
<td>( z_1(\psi) )</td>
<td>( \Delta z_2 )</td>
<td>( \psi/\Delta \psi )</td>
<td>( dx/d\psi = 1/\Delta \phi )</td>
</tr>
<tr>
<td>PR</td>
<td>( \phi(z_3) )</td>
<td>( \Delta \phi )</td>
<td>( z_2/\Delta z_2 )</td>
<td>( dx/dz_2 = 1/\Delta \phi )</td>
</tr>
<tr>
<td>PP</td>
<td>( z_3(z_2) )</td>
<td>( \Delta z_3 )</td>
<td>( z_2/\Delta z_2 )</td>
<td>( dx/dz_2 = 1/\Delta \phi )</td>
</tr>
</tbody>
</table>

Table 1: Generalized Input-Output function

According to the type of movement desired on the output, the follower can be oscillatory or intermittent. Furthermore it is considered, for all cases, that the cam has a constant speed, either linear or angular.

The normalized functions \( \tau \) that \( USyCaMs \) can perform are extensively studied in the literature, and are listed below:

- Harmonic
- Cycloidal
- Modified Cycloidal
- 3-4-5 Polynomial
- 4-5-6-7 Polynomial
- Modified Trapezoidal
- Modified Harmonic

D. Types of movement
There are two types of movement that \( USyCaMs \) can perform, oscillating movement and intermittent movement. The former is divided into four intervals, dwell, rise, dwell, and return. For the intermittent movement there are two intervals, rise and dwell.

E. Configurations and nomenclature of cam mechanisms
The number of possible configurations of cam mechanisms is sixteen, which is given by the number of links, type of follower motion, type of joint at the input, and type of joint in the output. The number of combinations of all of them results in that sixteen possibilities. With this, we can identify each type of mechanism by setting the following nomenclature:

\[ x_{xxx}xx \]

x: The number of links of the mechanism, 3 or 4.
xxx: Type of movement of the follower, indexing (ind) or oscillating (osc).
xx: The type of kinematic pair, prismatic (p) or revolute (r), for the input or the output axes, respectively.

THE SOFTWARE

The software, \( USyCaMs \), is user friendly software to synthesize one of fifteen of mechanisms choices, seven mechanisms of three links, and eight mechanisms of four links, having different arrangements depending on the motion program of the follower and the type of transmission of the kinematic pair in the entry and exit of the cam mechanism. And was created from a development kit ADEFID\textsuperscript{®} [5] (ADvanced Engineering platForm for Industrial Development), that is a set of libraries designed to study mechanical and mechatronic systems with OpenGL\textsuperscript{®} graphics [6,7].

A. Main Characteristics
The program allows the user to select one of fifteen cam mechanisms by the mouse, besides being able to do zoom in and out, translations and rotations. Also the user can adjust the settings of the scene, such as scale, projection type, background color, and so on.

The program focuses on the calculation and visual representation of the synthetized cam mechanism, which can be flat, spherical and spatial mechanisms. Furthermore, the animation of the mechanism can be performed by clicking on some buttons.

The user can enter data interactively through controls and see real-time changes on the mechanism. Moreover, it possible to save the vertices that defines the cam in text format and AutoCAD script format.

B. Synthesis of a cam mechanism
To synthesize a cam mechanism, simply select one of the fifteen types available, and change the properties of the mechanism as required.

Mechanisms can be selected through menus, or through a toolbar. And the parameters can be modified by accessing dialogs by means of menus or the toolbar.
MENU OPTIONS

USyCaMs has three basic menu options, namely, *Type of Mechanism* menu, *IO Function* menu and *GUI* menu.

A. *Type of Mechanism*

The menu contains five submenus and one entry called *Parameters* which executes the dialog with the same name. The menu is showed in Figure 2.

![Figure 2 Type of Mechanism Menu](image)

With the first four submenus the user is able to select the number of links, type of motion, and the type of kinematic joint at entry/exit. Also, with the fifth submenu the user can select the type of mechanism with its identification name.

B. *IO Function*

This menu manages the functions for the rise and return of the cam mechanism, Figure 3 shows the design of the menu. It contains two submenus and one entry to the Programming dialog.

![Figure 3 IO Function Menu](image)

The first submenu contains other submenus to specify the *Displacement Programs* and the *Rise/Return* function. The second submenu has only a *Rise* item.

For the *Rise/Return* submenu, the type of function is specified; these types include the principal function harmonic, cycloidal, trapezoidal and polynomial, as well as its modified function. For a detailed review see reference [8].

C. *GUI*

Four entries to manage the animation process are defined in the *GUI* menu, as indicated in Figure 4. The first five entries are used to manage the graphics scene. And the last four entries have the purpose of carrying out the animation of the cam mechanism in question. The user can start the animation, stop animation, and increase or decrease the speed of the animation.

![Figure 4 GUI Menu](image)

TOOLBAR OPTIONS

The first toolbar is devoted to select the mechanism, and to activate the entries of the *Type of Mechanism* menu and the *Programming* entry of the *IO Function* menu.

![Figure 5 Toolbar to select the mechanism and trigger dialogs](image)

The first four pairs of buttons work just like first four submenus of the *Type of Mechanism* menu. The last two buttons activates the *IO Function* dialog and the *Parameters* dialog, respectively.
The second toolbar works in the same manner as the four animation entries of the GUI menu. Figure 6 shows the toolbar.

![Figure 6 Toolbar to animate the mechanism]

**DIALOG OPTIONS**

**USyCaM** uses two dialog boxes to exchange data with the user. These resources are modeless dialogs to allow interaction with the main window where the mechanism is displayed, and are programmed in such a way to perform the instant actualization of the mechanism when a parameter is changing.

**A. Parameters of Mechanism**

This dialog has the purpose to manage geometrical parameters of mechanisms, and consists of several controls, such as static texts, sliders, edit controls, group boxes and buttons, as shown in Figure 7.

![Figure 7 Parameters of Mechanism Dialog]

In the *Pairs of axis* group box the user can specify geometrical parameters like distances and angles. While in the *Thickness & Steps* group box we can change the width of the cam, and we can change the number of steps for intermittent cam mechanisms.

**B. IO Function**

This dialog has the intention of managing properties and numerical values of the rise/return functions. The layout of this dialog is shown in Figure 8.

![Figure 8 IO Function Dialog]

The user can choose the type of function at rise/return selecting an entry in the indicated combo boxes, and in the Displacement Program combo box the user can specify the type of displacement program for the oscillating mechanisms.

*Kappa* variables are controlled with two spin controls, and they are only available when the *Modified Cycloidal* function is selected, since this function uses them. The controls inside of the *Variables* group box manages numeric values, except for the return edit box, because its value is calculated internally and displayed as information only.

With the profile button the user can save the vertices of the cam as a text file (.txt) or as an AutoCAD script (.scr). With the first extension the coordinates are saved as point per line separated by commas, and for the second, the coordinates are saved as an AutoCAD® script format in order to regenerate the solid in that package.
ON THE SOFTWARE DEVELOPMENT

All USyCaMs algorithms were created in Visual Studio 2010® (Visual C++ [9]) integrating a graphic environment based on OpenGL [7]. The program uses the standard libraries of MFC® [9] (Microsoft Foundation Classes), making it compatible with Microsoft Windows® operating system.

A. Methodology
The description of the entire development process of the program is large and complex, and strong programming skills were required, therefore this article outlines the main phases for its development, showed in Figure 9. Reference [10,11] shows the entire development.

B. Class USyCaMs and Classes for the cam mechanisms
There are as many as fifteen types of mechanisms that can be designed in USyCaMs and one class for each type was developed. Each class, encapsulates properties and functionality, and contains the algorithms for constructing contact surfaces of the cam and follower, as well as methods to render the solids on the screen.

Each mechanism class are derived from the base class CUSyCaMs, which aims to manage the main program algorithm and the performance of polymorphic mechanism between instances of each type of mechanism. CUSyCaMs inherits, in turn, properties from ADEFID® library. Figure 10 shows a hierarchical tree of the classes mentioned.

<table>
<thead>
<tr>
<th>Vp3_var</th>
<th>Param.</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>α₁</td>
<td>Distance between the axes of input and output</td>
</tr>
<tr>
<td>1</td>
<td>α₂</td>
<td>Angle between the axes of input and output (radians)</td>
</tr>
<tr>
<td>2</td>
<td>α₃</td>
<td>Distance between output shafts and roller</td>
</tr>
<tr>
<td>3</td>
<td>α₄</td>
<td>Angle between the output shafts and roller (radians)</td>
</tr>
<tr>
<td>4</td>
<td>α₅</td>
<td>Distance between the axes of roll and of its generatrix</td>
</tr>
<tr>
<td>5</td>
<td>α₆</td>
<td>Angle between the axes of roll and of its generatrix (radians)</td>
</tr>
<tr>
<td>6</td>
<td>λ₁</td>
<td>Distance to the origin of the first cam face</td>
</tr>
<tr>
<td>7</td>
<td>λ₂</td>
<td>Distance to the origin of the second cam face</td>
</tr>
</tbody>
</table>

Table 2 Meaning of vp3_var used on dialog Parameters of Mechanisms
D. Variables used in ‘IO Function’ dialog

Table 3 shows the required variables for the IO Function dialog, except the fifth element of vp4 var array which is used in the Parameters of Mechanism dialog.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>rise[0]</td>
<td>Stores the number of the rise function</td>
</tr>
<tr>
<td>retu[0]</td>
<td>Stores the number of the return function</td>
</tr>
<tr>
<td>disp_prog</td>
<td>Stores the number of the displacement program</td>
</tr>
<tr>
<td>vp4 var[0]</td>
<td>Interval oscillation of the follower (Amplitude)</td>
</tr>
<tr>
<td>[1]</td>
<td>Start value of input-output function (Elevation)</td>
</tr>
<tr>
<td>[2]</td>
<td>Rise, as a fraction of full turn</td>
</tr>
<tr>
<td>[3]</td>
<td>High dwell, as a fraction of full turn</td>
</tr>
<tr>
<td>[4]</td>
<td>Number of steps of intermittent mechanisms</td>
</tr>
<tr>
<td>[5]</td>
<td>Rise in radians (=vp4 var[3]<em>2</em>PI)</td>
</tr>
<tr>
<td>[6]</td>
<td>Kappa (\kappa) for modified cycloidal rise function</td>
</tr>
<tr>
<td>[7]</td>
<td>Kappa (\kappa) for modified cycloidal return function</td>
</tr>
</tbody>
</table>

Table 3 USyCaMs variables used for the dialog IO Function

RESULTS

To show the USyCaMs perform, several examples are presented. And the Figure 11 shows other examples from commercial packages. Figure 14 shows the three possible solutions of three-link cam mechanisms with indexing follower. Figure 15 shows the four solutions of three-link cam mechanisms with oscillating follower. Also, in Figure 16 and Figure 17 are presented all possible cases regarding four-link cam mechanisms. In Figure 16 the follower is indexing, while in Figure 17, the follower is oscillating. In all figures, the cam and the follower, are rendered in orange and purple (darker), respectively.

The functions applied for the synthesis of the fifteen cases, are presented in Table 4. In the same table, the numerical values and the displacement programs are listed as well. Furthermore, Table 5 shows the considered values to define the geometrical parameters.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dipp</td>
<td>MCyc0.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>360</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>345 Poly</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>140</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>Cycloidal</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>196</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>MCyc0.0</td>
<td>-</td>
<td>-</td>
<td>2.50</td>
<td>0.00</td>
<td>-</td>
<td>360</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>MCyc0.5</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
<td>0.60</td>
<td>-</td>
<td>360</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>Harmonic</td>
<td>Harmonic</td>
<td>DRDR</td>
<td>1.22</td>
<td>1.05</td>
<td>41</td>
<td>117</td>
<td>97</td>
<td>105</td>
</tr>
<tr>
<td>Dipp</td>
<td>Harmonic</td>
<td>Harmonic</td>
<td>DRDR</td>
<td>0.50</td>
<td>2.00</td>
<td>30</td>
<td>160</td>
<td>0</td>
<td>170</td>
</tr>
<tr>
<td>Dipp</td>
<td>MCyc1.0</td>
<td>-</td>
<td>-</td>
<td>2.09</td>
<td>-</td>
<td>-</td>
<td>270</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>MCyc1.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>360</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>MCyc1.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>240</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dipp</td>
<td>Harmonic</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>170</td>
<td>-</td>
<td>190</td>
</tr>
</tbody>
</table>

Table 4 Used values for the functions rise/return of the mechanisms presented

![Figure 11 SolidWorks® example](image1)

![Figure 12 CATIA® example](image2)
Figure 13 AutoCAD Mechanical® example

Figure 14 Three link mechanisms with indexing follower
(3ipr, 3irp and 3irr).

Figure 15 Three link mechanisms with oscillating follower
(3opp, 3opr, 3orp and 3orr)
Figure 16 Four link mechanisms with indexing follower (4ipp, 4ipr, 4irp and 4irr)

Figure 17 Four link mechanisms with oscillating follower (4opp, 4opr, 4orp and 4orr)
CONCLUSIONS

With the procedures and programming tools described above, USyCaMs was implemented in the Windows® operating system. With this, a contribution is provided to the Windows® community with a powerful, interactive and friendly design tool focused on cam mechanisms.

Moreover, due to the strong theoretical framework, the software package not only deals with planar, but also with spherical and spatial cam mechanisms.

USyCaMs is a powerful tool to represent on the screen a variety of cam mechanisms, including uncommon cam mechanism, what gives a particular advantage, since no commercial package can make the design of these mechanisms, or at least do not have specialized tools to synthesize them, as we see in Figure 11 to Figure 13.

But this tool do not have CAD/CAM characteristics, is limited to simulate a series of preset cam mechanisms. The user cannot draw on the screen nor can generate code for machining.

Therefore, the only advantage over other software packages that can synthesize cam mechanisms is that USyCaMs can simulate, efficiently, the spherical and the spatial cam mechanisms.

ACKNOWLEDGEMENTS

The first author acknowledges the support from SNI, (Sistema Nacional de Investigadores), México. The second author acknowledges the support from CONACYT (Consejo Nacional de Ciencia y Tecnología), México.

REFERENCES