ABSTRACT
A spreadsheet based mechanism analysis and animation system is presented in this article. The system is developed in Excel using Ch, an embeddable C/C++ interpreter, and Ch Mechanism Toolkit. It allows users to develop and solve complicated planar mechanism problems conveniently using an easy to use Excel spreadsheets as the front end. Users can input the required data to define a mechanism in an Excel spreadsheet and execute script files associated with the work cells. The spreadsheet based system can be used for kinematic and dynamic analysis, graphical plotting, and animation for four-bar, crank-slider, fivebar, and sixbar linkages. Examples are provided to illustrate its ease of use. The source code and demos are available for downloading from the internet.

Keyword: Mechanism design, analysis, Excel, spreadsheet, Ch, C/C++ interpreter.

1 Introduction
General purpose software packages such as Automated Dynamic Analysis of Mechanical Systems (ADAMS), Dynamic Analysis and Design System (DADS), Working Model [1] were developed to solve complicated engineering design and analysis problems. Although these software programs are powerful, learning curves for these packages are quite steep because users are assumed to have full knowledge of the subject. Special purpose software packages such as the Linkage I Nteractive Computer Analysis and Graphical Enhanced Synthesis Package (LINCAGES) [2, 3], WATT by Heron [4], and Simulation and Analysis of Mechanisms (SAM) by Artas [5] are available for the synthesis and analysis of planar mechanisms. The software package SYNTHETICA [6] can be used for synthesis of spatial mechanisms. Compared with general purpose software packages, these specialized packages are easier to use for mechanism design and analysis.

Spreadsheet was originally created to help accountants to solve their accounting problems. Due to its friendly user interface and easy of use, it has been extensively used in many different fields. It is also widely used as an educational tool in engineering. For example, Excel spreadsheet has been used in simulating engineering systems, such as logic networks, control systems and antenna array design [7]. Spreadsheet has also been used in the field of machine design [8], such as design of worm gear geometry [9], determination of spur gear form factors [10], aircraft structure analysis [11].

To customize spreadsheets for solving complicated engineering problems, Excel spreadsheet is typically programmed in Visual Basic. Excel can also be extended by means of C code or other languages that support either OLE automation or Windows dynamic link libraries (DLL) [12]. An add-in can be created from a program making it a seamless part of the Excel Interface and can also add customized functions to automate complex pro-
Ch [13–15] is a C/C++ interpreter. It conforms to the ISO C standard with extensions and contains salient features of MATLAB for numerical computing, graphical plotting, and script computing. The Ch Mechanism Toolkit [16], [17] developed in Ch is significantly different from other software packages. The users are able to examine the available source code and demo programs provided within the toolkit to solve practical engineering problems. Open source programs based on Ch and Mechanism Toolkit for kinematic synthesis of mechanisms are also available [18]. Based on the Ch Mechanism Toolkit, a Web-based mechanism design and analysis module has been developed [19–21]. The user can design and analyze mechanisms interactively through a Web browser without computer programming. The Web-based system provides the user with a convenient means to quickly obtain solution to many mechanism design problems. It is especially suitable for teaching and learning mechanism design and analysis. The Ch Mechanism Toolkit has been used as a teaching tool in an undergraduate course “Computer-Aided Mechanism Design” at the University of California, Davis [22]. Students can use the toolkit’s high-level building blocks to develop their own software programs for solving complicated engineering analysis and design problems. For example students were assigned to develop a complete program for analysis and animation of a Whitworth quick return mechanism [23]. It allows the user to verify their solutions quickly with numerical and graphical output as well as animation. The toolkit provides students with the opportunity to study and understand the algorithms and their software implementation for mechanism design and analysis.

Although Web-based mechanism design and analysis is easy to use, it typically needs network connection. To take advantage of the power of Excel spreadsheet and Ch Mechanism Toolkit, we have developed a software package for spreadsheet-based interactive mechanism design and analysis [24]. In this software package, Excel spreadsheet is used as the front end for friendly user interface, numerical computations in the back end are performed in Ch Mechanism Toolkit through ChExcel [25]. ChExcel is an Excel add-in that integrates Microsoft Excel’s spreadsheet interface with Ch’s computational power. ChExcel allows users to access the computation power of Ch from Excel work-sheets and Visual Basic for Applications (VBA). ChExcel lets user to communicate and exchange data between the Ch and Excel workspace. The communication between Excel and Ch is controlled by a few simple function, keeping the interface simple and easy to use.

The spreadsheet-based interactive mechanism design and analysis package contains spreadsheets and C/C++ script files for various planar mechanisms including four bar, five bar, and multi-loop sixbar linkages. It allows users to develop and solve complicated planar mechanism problems conveniently using an easy to use Excel spreadsheets as the user interface. Users can input the required data to define a mechanism in an Excel spreadsheet and execute C/C++ script files associated with the work cells. The software package is ideal for rapid prototyping and teaching.

In this paper, design, implementation and application of spreadsheet-based interactive mechanism analysis and animation are described. The presentation of this paper will focus primarily on fourbar linkages. However, the ideas and concepts are applicable to other mechanisms as well.

2 Features of Spreadsheet Based Mechanism Analysis and Animation

The spreadsheet-based mechanism design and analysis system incorporates all features available in the Ch Mechanism Toolkit through spreadsheets, users can solve complicated mechanism problems without programming. Some tedious and often complicated processes, such as kinematic and dynamic analysis of a fourbar linkage, can be performed with ease. A user only needs to enter parameters and execute ChExcel functions in appropriate cells in a spreadsheet.

A Excel sheet for mechanism analysis is typically separated into two sections, namely input and output sections. The input section gets the required data, which are then processed in the back end using the Ch computational engine. The result is displayed in appropriate cells in the output section of the same Excel sheet. Excel sheets were developed for mechanism design and analysis, such as animation of a fourbar mechanism and plotting of coupler curve. They can be downloaded from internet [24]. The first section of a spreadsheet is typically for input data that are required for the analysis such as link lengths, phase angles, velocity and acceleration of the driving link. Different options can be selected for handling different situations such as solving for output angles for a given input angle or plotting for a coupler curve. The Excel sheets are also available for dealing with the fourbar linkage synthesis and dynamic analysis. Excel sheets can also be developed for special fourbar linkages such as Grashof, non-Grashof, straight line, quick return and symmetrical linkages as well as singular configurations.

Aside from fourbar linkage shown in Figure 1, the software package contains Excel sheets for analysis of crank-slider, geared fivebar and various sixbar linkages. The available sixbar linkage include the fourbar slider, Stephenson I and III, as well as Watt I and II sixbar linkages. Operations such as kinematic and dynamic analysis are performed on the planar linkages using C/C++ scripts in the back end that are accessed from the Excel sheet. For example, given a linkage configuration, the angular position, velocity and acceleration of various links can be determined. The Excel sheet also has the ability to invoke the graphical and animation capabilities of the Ch Mechanism Toolkit. This animation feature allows the user to visually study the behavior of the mechanism as it rotates through its entire range of motion.
3 Interface with Excel Spreadsheet for Mechanism Analysis and Animation

The Excel sheets were designed to be simple and easy to use. ChExcel functions placed in certain cells can be evaluated on accessing those cells. These functions can execute C/C++ script files using the data from the spreadsheet. An adjacent cell contains the same value with an additional symbol <=, to indicate a comment cell. For example, the comment <=ChPutMatrix(C4,"r1") in cell F4 in Figure 2 indicates that E4 contains the ChExcel function ChPutMatrix(C4,"r1"). When this ChExcel function is executed, the value in the cell C4 would be stored in a variable named r1 in the Ch space, which is equivalent to declaring a double variable r1 and initializing it with the value 12 in cell C4. This variable r1 can also be accessed from Ch. To execute a ChExcel function in a cell, press function key F2. The script file "fb_pos_initialization.ch" listed in Program 1 placed in the cell E2 is first executed, which includes the required header files and initializes the variables that would be used in the analysis. For example,

```c
CFourbar fourbar;
```

in Program 1 declares an object fourbar for the class CFourbar which will be used in accessing the member function of this class from other script files. Similarly, plot is an object declared for the class CPlot to handle the plotting feature available in Ch. The variables inputlimitmin, inputlimitmax, outputlimitmin and outputlimitmax were used in processing the limits of the fourbar linkage. The snapshot of the excel sheet that was developed in analysing the limits of the fourbar linkage is shown in figure 4. The array's theta_1deg and theta_2deg are the variables that store the possible two solutions generated for the given conditions of a fourbar linkage. Figure 2 is a snap shot of the Excel sheet used in calculating the angles θ3 and θ4 at a given value for θ2. The data given are lengths for each link and the phase angle β and rp for a coupler point shown in Figure 1. The software module calculating the angles θ3 and θ4 can be found in the Mechanism Toolkit. The zeros in column E shown in Figures 2 and 3 indicate the successful execution of the Ch add-in functions from Excel.

```c
#include <math.h>
#include <fourbar.h>
#include <chplot.h>
#include <stdio.h>

double inputlimitmin[2], inputlimitmax[2],
       outputlimitmin[2], outputlimitmax[2];
double theta_1deg[1:4], theta_2deg[1:4];
int fourbartype;
int i;

CFourbar fourbar;
class CPlot plot;
```

Program 1. The script for initialization of variables used in position analysis of fourbar linkage (fb_pos_initialization.ch).
Angular positions of the links are calculated in the script file fb_angularPos.ch listed in Program 2. The program first converts the input angles from degrees to radians, then determines the angular positions of links 3 and 4 using the member function CFourbar::angularPos(). The member functions setLinks() and setCouplerPoint() build a four-bar linkage, which would be used for future reference during the analysis. The function parameter for an angle for these member functions has to be in radian. Therefore, the input data in degree needs to be converted to radian. For example, statement

\[ \text{theta1} = \text{theta1} \times \text{M_PI} / 180; \]

converts \text{theta1} from degree to radian. Similarly, the known angles with the angle number given by the user and angle \( \beta \) are converted to radian. The developer can also create an option by which the user can choose U.S. customary or SI units to specify the link lengths. Furthermore, the relevant angles may be specified in degrees or radians by giving the user a choice. The ChExcel function ChGetMatrix(C21:C24, "theta1") allows the user to display the values of array \text{theta1} for joint angles of different links in the Ch script in cells from C21 through C24 as shown in Figure 3. These cell values can be changed according to the user input. Member functions setLinks() and setCouplerPoint() of class CFourbar initialize the variables for further process, whereas member function
angularPos() determines the angular position of every link from the given input. The configuration of a fourbar linkage can also be displayed graphically using functions in the Ch Mechanism Toolkit.

Figure 3 is a snapshot of the Excel sheet after executing the script file "fb_angularpos.ch" in cell E20 and ChExcel functions in cells E21 and E22. The fourbar linkage forms two circuits for the given $\theta_1$. The angles formed by the links in each circuit is given separately in an array in Ch. These two arrays can be displayed in proper cells as shown in Figure 3. The angles displayed in cells "C21 to C24" are for the first circuit and "C28 to C31" for the second circuit. These script files are reusable and can be executed from different Excel sheets.

```
thetal = thetal * M_PI/180;
theta_1deg[1] = thetal;
theta_1deg[angleno] = angle * M_PI/180;
theta_2deg[1] = thetal;
theta_2deg[angleno] = angle * M_PI/180;
beta = beta * M_PI/180;
fourbar.setLinks(r1, r2, r3, r4, thetal);
fourbar.setCouplerPoint(rp, beta);
fourbar.angularPos(theta_1deg, theta_2deg, angleno);
```

Program 2. The script for angular position analysis of fourbar linkage (fb_angularPos.ch).
The Ch Mechanism Toolkit contains classes for design and analysis of Geared fivebar mechanism, Crank slider, Fourbar slider mechanism, stephenson I and III mechanism and Watt I and II mechanism. The scripts associated with Excel spreadsheets executed in the back end use these classes to solve the design and analysis problems of different mechanisms.

4 An Example of Spreadsheet Based Mechanism Design and Analysis

In this section, an example will be used to further illustrate some of the features and application of the spreadsheet-based mechanism design and analysis.

Problem Statement: The link lengths of a four bar linkage shown in Figure 1 are given as follows: \( r_1 = 12\text{cm}, \ r_2 = 4\text{cm}, \ r_3 = 12\text{cm}, \) and \( r_4 = 7\text{cm}. \) The phase angle for the ground link is \( \theta_1 = 10^\circ. \) The coupler point \( P \) is defined by the distance \( r_P = 5\text{cm} \) and constant angle \( \beta = 20^\circ. \) Determine the angular positions \( \theta_3 \) and \( \theta_4 \) as well as the position for coupler point \( P \) when the input angle \( \theta_2 = 70^\circ. \) Simulate the motion of the fourbar linkage for its entire range of motion.

This problem can be quickly solved using the spreadsheet shown in Figure 2 and its continuation in Figure 4, which can be used as a template to solve other problems related to fourbar linkage. After inputting the required parameters, the analysis is performed by executing the ChExcel functions in the cells of column E. The possible formation of two circuits is also considered and two sets of results are determined. These sets of two values are stored in two separate variables which are then displayed in Excel.

The numerical results of the analysis are shown in Figure 5 which is the continuation of the spreadsheet shown in Figure 3. The angular ranges of the input and output links are determined using script file fb\_getJointLimits.ch shown in Program 3. The member function CFourbar::getJointLimits() determines the range of the input link and the output link in unit of radian which are later converted to degree explicitly.

The four bar linkage analyzed is a crank-rocker with two circuits. The input range for \( \theta_2 \) is a complete rotation with 360 degrees. The output range for \( \theta_4 \) is \( 77.98^\circ \leq \theta_4 \leq 150.15^\circ \) and \( -130.16^\circ \leq \theta_4 \leq -57.98^\circ \) for the first and second circuits, respectively. When \( \theta_2 = 70^\circ, \theta_3 \) and \( \theta_4 \) are 0.2713 and 1.4376 radian respectively for the first circuit and -0.9948 and -2.1111 radian for the second circuit, respectively.

![Figure 4](image-url)
Figure 5. A spreadsheet for analysis of a fourbar mechanism with output.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Range</td>
<td>Circuit (in deg)</td>
<td>Circuit 1</td>
<td>Circuit 2</td>
<td>0:ChAppendScript(&quot;tb_getjoints limits.ch&quot;)</td>
</tr>
<tr>
<td>Circuit (in deg)</td>
<td>Lower limit</td>
<td>0</td>
<td>0</td>
<td>0:ChGetMatrix(C3: D3, &quot;inputlimmin&quot;)</td>
</tr>
<tr>
<td>Lower limit</td>
<td>Upper limit</td>
<td>360</td>
<td>360</td>
<td>0:ChGetMatrix(C4: D4, &quot;outputlimmax&quot;)</td>
</tr>
<tr>
<td>Upper limit</td>
<td>360</td>
<td>360</td>
<td>0:ChGetMatrix(C5: D5, &quot;outputlimmax&quot;)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. A coupler curve generated from the script in Program 4.
Figure 7. A single frame of fourbar animation for the first branch.

Figure 8. All frames of fourbar animation for the first branch.
fourbar.setLinks(r1, r2, r3, r4, theta1);
fourbarType = fourbar.getJointLimits(
    inputLimitMin, inputLimitMax,
    outputLimitMin, outputLimitMax);

for (i = 0; i <= 1; i++) {
    inputLimitMin[i] *= 180./M_PI;
    outputLimitMin[i] *= 180./M_PI;
    inputLimitMax[i] *= 180./M_PI;
    outputLimitMax[i] *= 180./M_PI;
}

Program 3. The script for calculating the range of motion of fourbar linkage (fb_getJointLimits.ch).

In addition to the values for $\theta_3$ and $\theta_4$, the numerical output also displays the range of motion for the input/output links as well as the curve traced by the coupler point P for the first circuit of the four bar indicated by the value branchno in Program 4. On execution of script file fb_curve.ch, it displays a coupler curve shown in Figure 6, which is created by the member function fourbar.plotCouplerCurve().

CPlot plot1;

fourbar.setLinks(r1, r2, r3, r4, theta1);
fourbar.setCouplerPoint(rp, beta);
fourbar.setNumPoints(50);
fourbar.plotCouplerCurve(plot1, branchno);

Program 4. The script for plotting a coupler curve for fourbar linkage (fb_curve.ch).

The animation for the four bar linkage is generated for the specified circuit using script file fb_animation.ch. The member function call fourbar.setNumPoints(50) specifies 50 frames for the animation. The function call fourbar.animation(branchno) launches the animation for the specified branch by the integral value branchno. Figure 7 shows one frame of animation for the first branch of the fourbar mechanism, whereas Figure 8 is an overlay of all the frames of animation.

Program 5. The script for animation of a fourbar linkage (fb_animation.ch).

Note that the four bar specified in the above problem statement is a crank-rocker, which means that there are two distinct circuits for each input link position. Thus, the user can display either first or second circuit by selecting the branch number from the cell C44 and C49 in Figure 4.

5 Conclusion

The Excel spreadsheet-based mechanism analysis and animation system, has been presented in this paper. The software package for spreadsheet-based mechanism analysis and animation system is developed using ChExcel and Ch Mechanism Toolkit. The package uses spreadsheets as the graphical user interface (GUI) and Ch scripts to to perform computations. It can be readily used to perform relevant analysis and animation of various mechanisms including four bar linkage, crank-slider mechanism, five bar linkage, multi-loop sixbar linkages such as Fourbar slider, Stephenson I and III, as well as Watt I and III. The system contains salient features of both friendly familiar Excel interface and powerful computational capabilities of Ch, a C/C++ interpreter with high-level extensions for applications in engineering and science. A large number of software modules in Ch are readily available for spreadsheet-based applications. The ideas and concepts presented for spreadsheet-based mechanism design and analysis can be applied to solve many other engineering problems which can take advantage of the friendly user interface of Excel and computational power of C/C++.

ACKNOWLEDGMENT

This work was supported in part by the University of California, Davis through an undergraduate instructional improvement grant and an educational grant from Intel Corporation.

REFERENCES


