



IMPLEMENTATION OF SHAPE CHANGER AS PRELIMINARY PROCESS TO SYNTHESIZE A MECHANISM IN CATIA SOFTWARE

M. H. Ismail¹, S. A. Shamsudin^{1,2} and M. N. Sudin^{1,2}

¹Centre for Advanced Research on Energy, Universiti Teknikal Malaysia Melaka, Malacca, Malaysia

²Faculty of Mechanical Engineering, Universiti Teknikal Malaysia Melaka, Malacca, Malaysia

E-Mail: mohammadhazrin858@gmail.com

ABSTRACT

This paper implements the 'ShapeChanger' code with MATLAB to approximate a set of planar profiles that have significant differences in arc length. The first step is to specify the types of segments; the mean segments (M-segments) or/and the constant curvature segments (C-segments). Then, a chain of rigid body segments from a set of points is created. Additionally, compound segments combine the M- and C-segments where possible. This work is extended to prepare the parametric data of the segments and profiles to be transferred into the Catia CAE software via macros commands. Next, Geometric Constraint Programming (GCP) techniques are applied to synthesize a mechanism.

Keywords: rigid-body mechanism, shape-changing, segmentation process, GCP.

INTRODUCTION

The use of computer aided software to solve mechanism problems began in late 1950's notably when Freudenstein and Sandor published the first paper on this topic (1959). Later, (Freudenstein, 1973) discussed and provided some examples of how the synthesis program can be utilized using a "digital computer" to solve several related synthesis problems. Developed by Kaufman using interactive computer systems, KINSYN (Kaufman and Maurer, 1971) was capable of designing linkages with slider joints in addition to revolute joints. Other programs were introduced later such as LINCAGES and RECSYN, but some have ceased to exist with changes in computer platforms (Waldron and Kinzel, 2003). Two and three position syntheses in RECSYN were successfully reprogrammed into MATLAB (Chase *et al.*, 2013).

ShapeChanger was aimed to solve the mechanism problems from the early stage involving the definition of segmentation of the design profiles. This paper begins with introducing the function of some of its interfaces that will help the user to create the design profiles. The next section shows how the design profiles are converted into target profiles. The last section of this paper details the steps to transfer the data from MATLAB software into Catia software – a computer-aided engineering (CAE) tool – using Visual Basic for Application (VBA) from Excel application.

MATLAB IMPLEMENTATION FOR THE SEGMENTATION PROCESS

This paper introduces ShapeChanger, a MATLAB-based development code to perform all segmentations shown in this work including the process to create a chain of rigid body segments from a set of open, closed, or fixed-end profiles. This work is based on matrices where a design vector dictates the creation of an initial segment matrix and specifies the segment types (M- and C-segments) in the chain. A mean segment (M-segment) is a segment that contains the same number of points in all instances on each target profile while a

constant curvature segment (C-segment) is a segment that consists of different number of points in all instances on each profile (Shamsudin *et al.*, 2013). The segment distance errors are calculated to improve the unconnected segments. The process can be repeated with other design vectors and tried with many starting guesses. The set of segments can be connected and each instance of segment location optimized versus the set of target profiles.

Design profiles

The first step is to generate design profiles. Design profiles may be entered into ShapeChanger in two ways. First, design profiles may be entered as an array of points that describe a piecewise-linear curve. This would be typical of profiles generated from another application. MATLAB is then used to transform those profiles into splines. Second, design profiles may be generated directly within ShapeChanger utilizing features of MATLAB's Spline Toolbox. Launching ShapeChanger at the MATLAB command prompt opens the window shown in Figure-1(a).

Generating design profiles via ShapeChanger

Before generating design profiles from ShapeChanger, the user needs to identify the type of design profiles; open curves, closed curves, or fixed-end curves. It is important for ShapeChanger to know the type of design profiles, as it will be used for later operations. The steps to create these three profiles are mostly the same, but ShapeChanger enforces additional constraints for closed or fixed-end curves. For closed profiles, the profile is automatically closed when the user clicks the right button of the mouse, meaning that the first and last points on each profile are constrained to be the same. For fixed-end profiles, the locations of the fixed-ends are determined from the first and last points of the initial profile created.



Uploading files

Selecting the “Upload File” button gives the user to upload the design profiles generated from the previous work done via ShapeChanger or another application. For the design profiles generated apart from ShapeChanger, the coordinates identifying the points on the design curves needs to be saved in a format that ShapeChanger can manipulate.

Editing design profiles

The user is able to edit the design profiles from the “Edit” windows as shown in Figure-1(b). The design profiles may be individually edited via the addition, deletion, or movement of the spline’s control points. The design profiles may also be translated, rotated, and scaled. The placement (including orientation) of each design profile does not affect the segmentation results. The relative locations of the design profiles are critical, however, to the mechanization step of designing a rigid-body shape-changing device. Thus, translation and rotation are presented with the long-term goal of integrating ShapeChanger with a complete set of mechanization algorithms. The scaling option, when selected, scales all design profiles to the same arc length as the shortest profile in the set. The scaling option is useful when the design profiles need to be the same arc length, as in the case of synthesizing a chain that includes only M-segments.

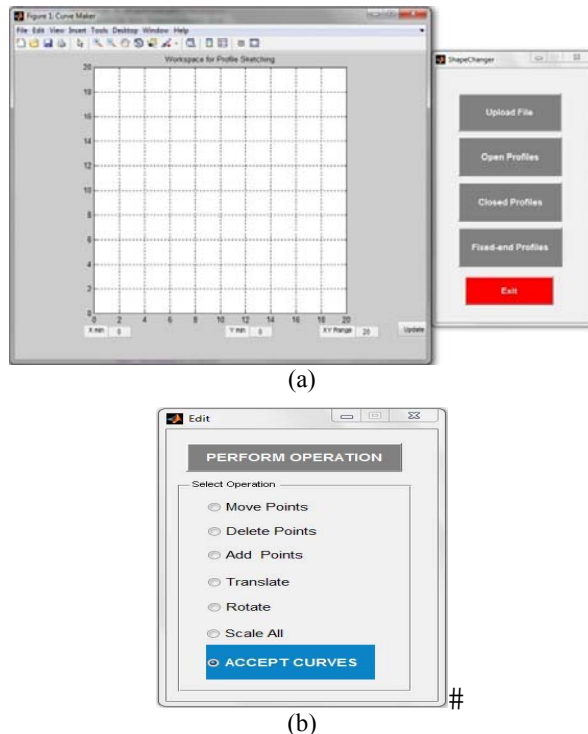


Figure-1. Upon ShapeChanger application is launched, (a) the window for selecting the types of profiles and generating the design profiles are opened, and (b) the window for editing the design profiles is appeared once the design profiles are generated.

GENERATING TARGET PROFILES

Given the arc lengths of the design profiles, target profiles are now can be generated. The target profiles are defined as a set of curves that represent the set of design profiles. ShapeChanger has common features to compare the groups of contiguous points with all profiles in order to form a suitable chain of rigid bodies that when repositioned will approximate all design profiles.

In generating target profiles, ShapeChanger faces difficulty to specify a set of p design profiles that represent the different shapes to be attained by the mechanism. (Murray *et al.*, 2008) describe a design profile j as an ordered set of n_j points for which the arc length between any two can be determined.

Design profiles can be viewed as being piecewise linear. A piece is the line segment connecting two contiguous points on the profile. The i^{th} point on the j^{th} design profile is designated $\{a_{j_i}, b_{j_i}\}^T$. The length of the i^{th} piece on the j^{th} design profile is

$$c_{j_i} = \sqrt{(a_{j_{i+1}} - a_{j_i})^2 + (b_{j_{i+1}} - b_{j_i})^2}, \quad (1)$$

and the arc length of the j^{th} design profile is

$$C_j = \sum_{i=1}^{n_j-1} c_{j_i}. \quad (2)$$

The design profiles may be defined by any number of points spaced at various intervals, producing a wide range of c_{j_i} .

The general profiles may contain substantial differences in arc length. Using the same number of points on different length profiles would result in different piece lengths and contaminate the shape comparisons among groupings of contiguous points. In order to produce a constant piece length, the conversion scheme from design to target profiles must be modified from the established method to allow for a different number of points on each target profile. By specifying a desired piece length s_d , the number of pieces m_j on profile j can be determined. Smaller values of s_d will produce more pieces and typically result in smaller variations between the design and target profiles.

The number of pieces must be an integer, and initial value is calculated as

$$m_j = \left\lceil \frac{C_j}{s_d} \right\rceil \quad (3)$$

where $\lceil \zeta \rceil$ represents the ceiling function, the smallest integer when ζ is rounded up. The n_j points are placed at



increments of C_j/m_j along the j^{th} design profile to generate provisional target profiles. The j^{th} target profile becomes a piecewise linear curve connecting the ordered set of points $z_{j_i} \{x_{j_i}, y_{j_i}\}^T$, $i=1, \dots, n_j$. The length of the i^{th} linear piece on the j^{th} target profile is

$$s_{j_i} = \|z_{j_{i+1}} - z_{j_i}\| = \sqrt{(x_{j_{i+1}} - x_{j_i})^2 + (y_{j_{i+1}} - y_{j_i})^2}. \quad (4)$$

The corresponding number of points on the target profile j is $n_j = m_j + 1$.

For a provisional target profile, the piece length $s_{j_i} \leq s_d$, as in Equation 3 generated slightly more segments than would exactly match the profile using the piece length s_d . In addition, any curvature of the design profile results in piece lengths shorter than s_d . In this case, some piece $s_{j_i} = s_d$ only when s_d divides exactly into C_j and the design profiles has a zero curvature portion long enough to include the entire piece. The average piece length for the j^{th} profile is

$$\bar{s}_j = \frac{1}{m_j} \left(\sum_{i=1}^{n_j-1} s_{j_i} \right). \quad (5)$$

As the provisional target profile is constructed to (potentially) have more pieces to accurately achieve s_d , the probable scenario is that fewer pieces will produce a value of \bar{s}_j closer to s_d . An error representing the difference between the average segment length and desired piece length is calculated as $E_{s_j} = |s_d - \bar{s}_j|$. Decreasing n_j by 1 and redistributing points along the design profile creates a new target profile. Points are removed until $n_j = n_j^*$ (and correspondingly, $m_j = m_j^*$) that minimizes E_{s_j} . The end result is the fewest n_j^* points are used to

construct the j^{th} target profile such that all linear piece lengths are approximately equal to the desired piece length. Desirable target profiles are those with the fewest pieces that achieve the accuracy needed to satisfactorily represent the original design profiles the more pieces used in a set of target profiles, the approximation of the design profiles.

After each m_j^* is established, the total length of the j^{th} target profile is calculated as

$$S_j = \sum_{i=1}^{m_j^*} s_{j_i}. \quad (6)$$

Applying this to all design profiles, p target profiles are constructed such that all linear pieces possess lengths that are about equal to s_d . The average length of all m_j^* linear pieces on all p profiles is

$$\bar{s}_m = \frac{\sum_{j=1}^p S_j}{\sum_{j=1}^p m_j^*}. \quad (7)$$

The accuracy of the target profiles gauged by their arc lengths versus those of the original design profiles. In general, smaller desired piece length s_d results in a more accurate representation of the design profile.

ShapeChanger does not allow a value of s_d larger than one-third of the arc length of the shortest profile as too few pieces will result, and the likelihood of acceptable target profiles is small. Although s_d is never exactly achieved, with a small desired piece length compared to the arc length, or $s_d \ll C$, the target profiles will represent the design profiles very closely.

The values of \bar{s}_j and m_j^* can be determined by giving the set of target profiles from a selected s_d . There is now an important interplay between variables that must be identified before segmentation may occur. The user must identify the preferred number of segments q for the segmentation process as well as the minimum number of pieces per segment, α . Note that, at a minimum, the number of pieces on the shortest profile has to be greater than αq . The optimization algorithm adjusts the segment matrix, such that pieces can be added or removed from each segment. Thus, the heuristic that relates q to the number of pieces on the shortest curve to α is

$$q = \left\lceil \frac{S_j}{2\alpha} \right\rceil, \quad j=1, \dots, p. \quad (8)$$

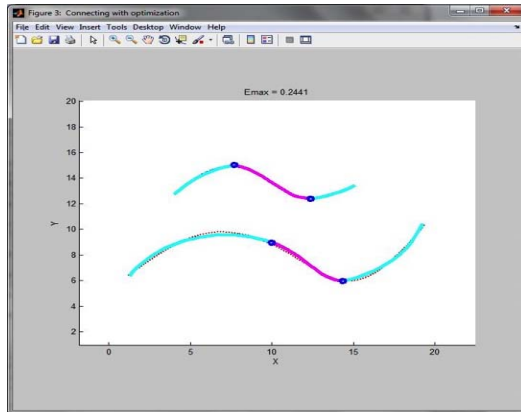
TRANSFERRING MATLAB DATA TO CATIA SOFTWARE

MATLAB can be integrated with other design and analysis software such as SolidWorks and ADAMS, but there is still no function or plug-in can be used to directly connect data or results from MATLAB to Catia or vice versa. In this case, the user needs a platform to connect the MATLAB data with Catia software. Before the data can be imported into Catia, a preliminary process is essential so that the data could be integrated with Catia.

Back to ShapeChanger, after the segmentation process is completed, the segmented profiles are presented in MATLAB figure format where the user can view the result as two-dimensional figure as shown in Figure-2(a). The data of xy coordinates of the segments and profiles defined by matrix form can be extracted from earlier MATLAB figure file and must be saved into binary file



format called Excel Binary File Format (XLS). Macros command is implemented into Microsoft Excel application via Visual Basic for Application (VBA). Figure-2 shows how the flow of segmentation process is done from ShapeChanger software to Catia software.



(a)

x-axis	y-axis	z-axis
Start/End	Start/End	Start/End
3.984744255	12.74790927	0
4.363391149	13.1564946	0
4.773364459	13.53363825	0
5.212361619	13.87694605	0
5.676697721	14.18423865	0
6.164323193	14.4535653	0
6.671842524	14.68321629	0
7.196033911	14.87173375	0
7.73356972	15.01792097	0
EndCurve	EndCurve	EndCurve
Start/End	Start/End	Start/End
7.73356972	15.01792097	0
8.256977621	14.83282889	0
8.759463384	14.60016848	0
9.236695211	14.31272727	0
9.687026322	13.98499259	0
10.11849259	13.63396964	0
10.54658297	13.27928161	0
10.96928164	12.94197692	0
11.48252453	12.65316698	0
11.98226338	12.45810724	0
12.52301303	12.37272154	0
EndCurve	EndCurve	EndCurve
Start/End	Start/End	Start/End
12.52301303	12.37272154	0
13.06422758	12.50463914	0
13.593123	12.67962504	0
14.10629389	12.89625318	0
14.60043607	13.15342813	0
15.0723679	13.44939399	0
EndCurve	EndCurve	EndCurve

(b)

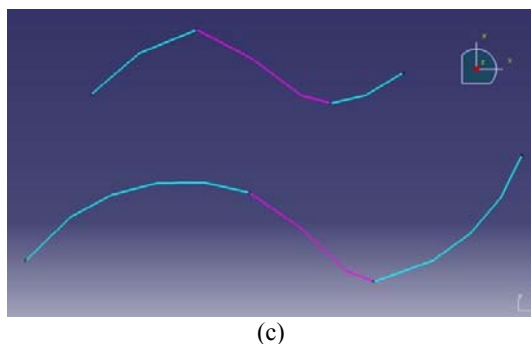


Figure-2. (a) ShapeChanger does the segmentation. (b) The data of xy coordinates can be transferred into Excel application. (c) Visual Basic for Application (VBA) is applied to convert data from Excel application into Catia.

The segmented profiles are ready to be added with some mechanisms in order to show how the design

profile is morphing between two or more shapes. In this work, Geometric Constraint Programming (GCP) is used to do the mechanization process with the help of Catia software. The selecting of Catia as design software is based on the some factors. As this work will be extended to build solid model for this design profiles, Catia software offers the advanced tools to convert two-dimensional profiles into three-dimensional model. DMU Kinematic Simulator, one of environments available in Catia software simulates the mechanism into real time motion and allows the user to determine the number of degree of freedom (DOF) needed to mechanize the mechanism (Zamani and Weaver, 2012). This function eases the user to make any improvement on the mechanism to achieve the desired DOF.

CONCLUSIONS

This paper implements ShapeChanger application to synthesize planar profiles that are capable of approximating a shape-change defined by a general set of different-length curves. The result of this work is the creation of the chain of segments joined together with revolute and/or prismatic joints. One example of segmentation process for open design profiles is successfully done in order to show the steps of segmentation process started with the use of ShapeChanger application, into VBA from Microsoft Excel software and ended on Catia software.

ACKNOWLEDGEMENTS

This research is supported by the grants from the Department of Higher Education of Malaysia with grant number RAGS/2013/TK01/FKM/04/B0043. Thanks are given to the Universiti Teknikal Malaysia Melaka (UTeM) for providing support and facilities for research.

REFERENCES

- [1] Chase, T.R., Kinzel, G.L. and Erdman, A.G. 2013. Computer Aided Mechanism Synthesis: A Historical Perspective. Appears in Adv. in Mech., Rob. & Des. Educ. & Res., Kumar, V., Schmiedeler, J., and Sreenivasan, S. V. (eds.), Springer International Publishing, Switzerland, pp. 17-33.
- [2] Freudenstein, F. 1973. Kinematics: Past, Present and Future. Mechanism and Machine Theory, 8, pp. 151-160.
- [3] Freudenstein, F. and Sandor, G. N. 1959. Synthesis of Path Generating Mechanisms by Means of a Programmed Digital Computer. ASME J. Eng., 81B(2), pp. 159-168.
- [4] Kaufman, R.E. and Maurer, W.G. 1971. Interactive Linkage Synthesis on a Small Computer. ACM National Conference.



- [5] Murray, A.P., Schmiedeler, J.P. and Korte, B.M. 2008. Kinematic Synthesis of Planar, Shape-Changing Rigid-Body Mechanisms. *Journal of Mechanical Design*, 130(3), pp. 1–10.
- [6] Shamsudin, S.A., Murray, A.P., Myszka, D.H. and Schmiedeler, J.P. 2013. Kinematic Synthesis of Planar, Shape-Changing, Rigid Body Mechanisms for Design Profiles with Significant Differences in Arc Length. *Mechanism and Machine Theory*, 70, pp. 425–440.
- [7] Waldron, K. J. and Kinzel, G. L. 2003. *Kinematics, Dynamics, and Design of Machinery*. 2nd Edition, Wiley Publisher, New York.
- [8] Zamani, N.G. and Weaver, J.M. 2012. *CATIA V5 Tutorials Mechanism Design and Animation Release 21*, SDC Publications, Kansas.