KINEMATIC SYNTHESIS (FUNCTION GENERATION) IN ADAMS/VIEW

Nitin B. Talekar
Application Engineer, MDI

Todd DePauw
MDI

Abstract

Function generation is a branch of kinematic synthesis where a given relationship between the input and the output angular motion is required. Various techniques have been developed to solve the function generation problems. The usual graphical method of getting the functional relationship between the input and the output angle is by graphically constructing the circle point and center curves. In any given position of the mechanism, the circle point curve is the locus of all those points on the body in several positions of the body. The locus of all the fixed pivots is called the center curve. The selection of a pair of points, one on either curve, determines a binary link, two of which will guide the output link through the desired angles.

These curves can also be obtained analytically for the solution of the planar mechanisms but must be determined analytically for the solution of space mechanisms. In this paper the analytical method for solving the function generation problems for planar four-bar and 3D four bar (RSSR) is presented. This method was implemented within the ADAMS software in form of AVIEW functions to solve some of the practical problems encountered in various industries. A graphical interface was built over these AVIEW functions to make to them user friendly. For the planar four-bar the AVIEW functions can solve three, four and five point synthesis problems while for a 3D four bar the three-point synthesis algorithm was developed.

Introduction

The four bar mechanism is potentially one of the versatile and practical configurations for the mechanization of the motion requirement. This is because it is one of the simplest mechanical constructions for 2D and 3D mechanisms.

The present paper will implement methods for the synthesis of those four bar mechanisms, known as function generators, in which the motions of the driving and driven link are related by a prescribed function. Generally due to the finite number of specified parameters of a mechanism, the function generated by the mechanism coincides with the prescribed function only at a finite number of points, called precision points, and deviates from the given mathematical function between these points.

It is well known that the four bar mechanism has one of the simplest mechanical constructions and, furthermore, that one of the most important types of function generators involves rotation angle coordination between input and output. Therefore using revolute pairs at the input

and output terminal and restricting the design to a four bar mechanism, we will choose planar four bar and 3D four bar (Revolute-Spherical-Spherical-Revolute i.e. RSSR) mechanism for a particular detailed investigation.

The technique of kinematic inversion is used here for four bar function generator synthesis. AVIEW functions were created for three, four and five point synthesis of a planar four bar and for three-point synthesis of a RSSR mechanism. Finally the user interface was developed which makes the efficient use of the developed AVIEW functions in the design of new mechanisms.

Kinematic Synthesis

In order to understand how and why synthesis can complement the use of ADAMS, it is first important to note that synthesis is generally performed in what ADAMS would consider a kinematic model. In kinematics, the complete description of the motion of all bodies can be described as a function of time based on joint and motion constraints in the system. The synthesis of mechanisms depends on this known functional relationship between the parts in the mechanism. Once the mechanism is completed, it can be added to a complete dynamics model.

Current methods of synthesis are limited to particular classes of mechanisms. Most of these mechanisms are a combination of 3 or more parts constrained to a single plane. That is, the parts have only 2 translational and 1 rotational degree of freedom. There are special cases of 3 dimensional mechanism otherwise known as a spatial mechanism like RSSR mechanisms, spherical mechanisms, etc. that can be determined by these methods of synthesis. Few details about RSSR mechanisms will be presented later in this paper.

Although there are many configurations of kinematic mechanisms, each can be categorized as path generation, motion generation, or function generation. Although the focus of this paper is on function generation methods, it is useful to note the other types of mechanisms and their application to synthesis analysis.

The first type of synthesis is path generation. Path generation synthesis is used in the case where specific locations along the path of motion are known and is required. The synthesis generation provides a mechanism, which will trace a path, which includes these points as it articulates. An example application would be to create a mechanism which would move the tip of a paint robot through three specific points (1) beginning of spray, (2) end of spray, and (3) tip cleaning. Although it would be possible to design a controller to move the robot, synthesis of a single mechanism to follow this path would have the benefits of higher speed, increased reliability, and decreased cost.

Motion synthesis is another method of creating mechanisms to perform specified motions. Motion synthesis is similar to path synthesis, except that with motion synthesis the orientations of the parts, or at least one of the parts is desired. This type of synthesis is required when a mechanism includes a part that must be positioned at specific locations and orientations during the mechanism's articulation cycle. An example application of motion synthesis would be a stamping mechanism, which requires the stamping applicator to be parallel to the target surface at a specific point in the articulation of the mechanism. In another point in the cycle, it may require another

orientation, for example if the stamping pad required re-inking or cleaning after each stamp.

The final type of synthesis is function generation. Function generation is the synthesis of a mechanism that, through kinematic actuation, positions one part of the mechanism, by a specific functional relationship to another part. This allows for mechanisms, which can take advantage of mechanical advantage and connect the mechanism in many useful applications. A common example of the need for functional synthesis is levers and switches. These switches transfer small motions required for, electronic devices, to larger parts, which can be easily handled by human interaction. An example of a mechanism that was designed through functional synthesis is the braking mechanism on some popular brands of roller blades. In this case the relatively large motion of the ankle is transferred to the braking pad while allowing the stability of all of the wheels on the ground.

The next section will go through the theory followed in developing the 3-point function generating routine for planar four bars.

Synthesis Of 2d, 3-Point Function Generation

The Fig. 1 show a typical planar four bar mechanism. Linkage AB is the input side or the driver or the crank while linkage DC is the output side or the follower. Linkage BC is called the coupler since it couples the input and the output of the mechanism.

The dimensions and the angular information for this mechanism are shown in the figure. θ is the angle between the input link AB and the positive x axis while ϕ is the angle between the output link DC and the positive x axis. Throughout this work, for a planar mechanism, angles measured counterclockwise are positive.

For the coordinates system passing through point A and orientated with the positive x axis pointing towards D, the coordinates of point A, B, C and D are,

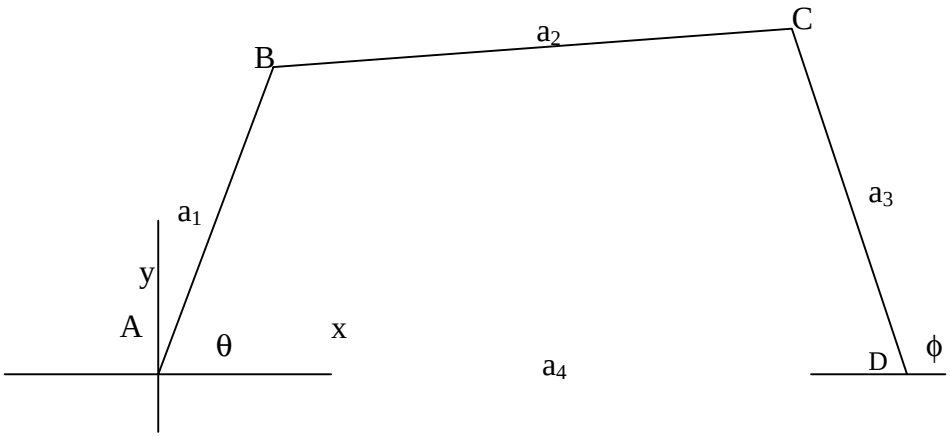

Figure 1: Planar Four Bar Mechanism

$A = (0, 0)$
$B = (a_1\cos\theta, a_1\sin\theta)$
$C = (a_4+a_3\cos\phi, a_3\sin\phi)$
$D = (a_4, 0)$

The synthesis equation for this mechanism ($BC = a_2$ = constant) is,

$$a_2^2 = (a_4+a_3\cos\phi-a_1\cos\theta)^2 + (a_3\sin\phi-a_1\sin\theta)^2$$

Expanding and collecting the terms together yields,

$$A_0 + B_n a_3 + C_n a_1 a_3 + D_n a_1 = a_2^2 \qquad (1)$$

Where:
$A_0 = a_1^2 + a_3^2 + a_4^2$
$B_n = 2a_4\cos\phi = 2a_4\cos(\phi_1+\phi_{1n})$
$C_n = -2\cos(\theta - \phi) = -2\cos(\theta_1+\theta_{1n}-\phi_1-\phi_{1n})$
$D_n = -2a_4\cos\theta = -2a_4\cos(\theta_1+\theta_{1n})$

Note, $\theta = \theta_1+\theta_{1n}$ and $\phi = \phi_1+\phi_{1n}$ where θ_1 and ϕ_1 are the initial angular positions of the input and the output linkages of the mechanism, respectively. Therefore, by definition θ_{11} , $\phi_{11} = 0$.

Equation (1) can be solved for a_1, a_2 and a_3 if the terms A_0, B_n, C_n and D_n are known. For a typical three point function generation synthesis problem the values of θ_{12}, θ_{13}, ϕ_{12}, ϕ_{13} are known. Also since θ_{11} , $\phi_{11} = 0$ if we assume some values for θ_1 , ϕ_1 equation (1) can be solved for a_1, a_2 and a_3. Note that a_4 acts as a scaling factor. Increasing the value of a_4 will scale up the mechanism while decreasing its value will scale the mechanism down.

In a similar way we can solve the four point and five point function generation synthesis equations. The exact formulation for these methods is beyond the scope of this paper. For the 3D four bar mechanism, there are more parameters than a planar four bar. The next section will go through the nomenclature and definitions of these parameters.

Symbolic Notations For 3D Four Bar Mechanism

This section is concerned with the kinematic notations of 3D four bar mechanisms. This mechanism is shown in Fig. 2.

The following axes are used to define the frame of reference for the constructing the equations:

{ EMBED Equation.2 **}** axis of rotation of the crank, oriented arbitrarily

{ EMBED Equation.2 **}** axis of rotation of the follower, oriented arbitrarily

{ EMBED Equation.2 **}** common perpendicular to **{** EMBED Equation.2 **}** and **{** EMBED Equation.2 **}**, oriented from its intersection with **{** EMBED Equation.2 **}**toward its intersection with **{** EMBED Equation.2 **}**

{ EMBED Equation.2 **}** perpendicular to **{** EMBED Equation.2 **}** through the center of **{** EMBED Equation.2 **}** oriented arbitrarily

{ EMBED Equation.2 **}** perpendicular to **{** EMBED Equation.2 **}** through the center of **{** EMBED Equation.2 **}** oriented arbitrarily

{ EMBED Word.Picture.8 **}**

Figure 2: Symbolic Notations

In terms of these axes, the mechanism parameters and variables are

{ EMBED Equation.2 **}** frame length, measured along **{** EMBED Equation.2 **}** from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}**; this distance is always positive

{ EMBED Equation.2 **}** frame angle, measured from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}** counterclockwise when looking from the end of **{** EMBED Equation.2 **}**; this angle may have any value from 0 to 360**{** EMBED Equation.2 **}**

{ EMBED Equation.2 **}** crank axial position, measured along **{** EMBED Equation.2 **}** from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}**; this distance may be positive or negative, depending on the position of the crank with respect to the common perpendicular **{** EMBED Equation.2 **}**

{ EMBED Equation.2 **}** follower axial position, measured along **{** EMBED Equation.2 **}** from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}**; like **{** EMBED Equation.2 **}**, this distance may be positive or negative

{ EMBED Equation.2 **}** crank length, measured along **{** EMBED Equation.2 **}**; this distance may be positive or negative depending on the orientation of **{** EMBED Equation.2 **}**

{ EMBED Equation.2 **}** coupler length, always positive

{ EMBED Equation.2 **}** follower length, measured along **{** EMBED Equation.2 **}**; like **{** EMBED Equation.2 **}**, this distance may be positive or negative

{ EMBED Equation.2 **}** crank angle, measured from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}** counterclockwise when looking from the end of **{** EMBED Equation.2 **}**; this angle may have any value from 0 to 360**{** EMBED Equation.2 **}**

{ EMBED Equation.2 **}** follower angle, measured from **{** EMBED Equation.2 **}** to **{** EMBED Equation.2 **}** counterclockwise when looking from the end of **{** EMBED Equation.2 **}**; this angle may have any value from 0 to 360**{** EMBED Equation.2 **}**

These are the notations and conventions used for the development of the three-point function generation synthesis function that was build for the RSSR mechanism. We will be referring to these notations in our next section.

Kinematic Synthesis ADAMS/View Functions

Using the theory, symbolic notations and the sign conventions described in the previous sections, ADAMS/View functions were developed which can be called within the ADAMS/View environment. This section will describe the input and the output from these ADAMS/View functions.

The names of the functions developed are,

- **syn_fg_3** for 3-point function generation of a 2D four bar
- **syn_fg_4** for 4-point function generation of a 2D four bar
- **syn_fg_5** for 5-point function generation of a 2D four bar
- **syn_fg_3_3d** for 3-point function generation of a 3D four bar or RSSR mechanism

All these functions require three input variables, syn_a, syn_b, syn_c and assign the output to a variable syn_d which has the location of points A, B, C, D and some other extra information. syn_a will pass the value of θ_{1n}'s, syn_b will pass the value of ϕ_{1n}'s,while syn_c will pass the assumed values of θ_1, ϕ_1 and few others depending on the function used.

Presently the solutions of syn_fg_3, syn_fg_4, syn_fg_5 will always lie in the x-y plane with point A or point 1 always at the origin i.e. coordinate transformation is not yet implemented within the function. But this transformation can be done outside these functions using the standard view functions. The same is true for syn_fg_3_3d where point A will always be at the origin.

syn_fg_3

The following code segment illustrates how to set up the function call in ADAMS/View command language ("!" denote comment lines)

```
! Define syn_a = θ12 , θ13
      variable  modify &
         variable = syn_a &
         real = 15, 30
! Define syn_b = ϕ12 , ϕ13
      variable  modify &
         variable = syn_b &
         real = 10.024, 19.852
! Define syn_c = θ1, ϕ1, a 4
      variable modify &
         variable = syn_c &
         real =  90, 90, 1000
! Call the synthesis function syn_fg_3
  variable modify &
     variable = syn_d &
     real=(syn_fg_3(syn_a,syn_b,syn_c))
```

The variable syn_d will be an array of size 14. The output will be in the following format, (#, error, x_A, y_A, z_A, x_B, y_B, z_B, x_C, y_C, z_C, x_D, y_D, z_D). Where:

- # = number of possible mechanisms
- error = 0 (no error), 1 (error)
- x_i, y_i, z_i = coordinates of points

syn_fg_4

The following code segment illustrates how to set up the function call in ADAMS/View.

```
! Define syn_a = θ12 , θ13, θ14
   variable modify &
      variable = syn_a &
      real = 10, 20, 30
! Define syn_b = ϕ12 , ϕ13, ϕ14
   variable modify &
      variable = syn_b &
      real = 6.6864,13.339,19.852
! Define syn_c = θ1, a 4
   variable modify &
      variable = syn_c &
      real =  90, 1000
! Call the function
  variable modify &
     variable = syn_d &
     real = (syn_fg_4(syn_a,syn_b,syn_c))
```

The variable syn_d will be an array of size 38. The output will be in the following format,
(#, error, x_{A1}, y_{A1}, z_{A1}, x_{B1}, y_{B1}, z_{B1}, x_{C1}, y_{C1}, z_{C1}, x_{D1}, y_{D1}, z_{D1}, x_{A2}, y_{A2}, z_{A2}, x_{B2}, y_{B2}, z_{B2}, x_{C2}, y_{C2}, z_{C2}, x_{D2}, y_{D2}, z_{D2}, x_{A3}, y_{A3}, z_{A3}, x_{B3}, y_{B3}, z_{B3}, x_{C3}, y_{C3}, z_{C3}, x_{D3}, y_{D3}, z_{D3})

- # = number of possible mechanisms 1,2,3
- error = 0 (no error), 1 (error)
- x_{A1}, y_{A1}, z_{A1} = coordinates of A,B,C,D

syn_fg_5

The following code segment illustrates how to set up the function call in ADAMS/View.

```
! Define syn_a = θ12 , θ13, θ14, θ15
   variable modify &
      variable = syn_a &
      real   =   -22.5,-45.0,-67.5,-
   90.0
! Define syn_b = ϕ12 , ϕ13, ϕ14, ϕ15
   variable modify &
      variable = syn_b &
      real=-16.9,-37.5,-61.9,-90.0
! Define syn_c = a 4
   variable modify &
      variable = syn_c &
      real =  1000
! Call the function
   variable modify &
     variable = syn_d &
     real=(syn_fg_5(syn_a,syn_b,syn_c))
```

The variable syn_d will be an array of size 38. The output will be in the following format,
(#, error, x_{A1}, y_{A1}, z_{A1}, x_{B1}, y_{B1}, z_{B1}, x_{C1}, y_{C1}, z_{C1}, x_{D1}, y_{D1}, z_{D1}, x_{A2}, y_{A2}, z_{A2}, x_{B2}, y_{B2}, z_{B2}, x_{C2}, y_{C2}, z_{C2}, x_{D2}, y_{D2}, z_{D2}, x_{A3}, y_{A3}, z_{A3}, x_{B3}, y_{B3}, z_{B3}, x_{C3}, y_{C3}, z_{C3}, x_{D3}, y_{D3}, z_{D3})

- # = number of possible mechanisms 1,2,3
- error = 0 (no error), 1 (error)
- x_{A1},y_{A1},z_{A1}= coordinates of point A,B,C,D

syn_fg_3_3d

The following code segment illustrates how to set up the function call in ADAMS/View.

```
! Define syn_a = θ12 , θ13
   variable modify &
      variable = syn_a &
      real = 15, 30
! Define syn_b = ϕ12 , ϕ13
   variable modify &
      variable = syn_b &
      real = 10.024,19.852
! Define syn_c = θ12 , ϕ12 ,a 4 , α, s1, s4
   variable modify &
      variable = syn_c &
      real = 90,90,1000,0,0,0
! Call the function
variable modify &
  variable = syn_d &
  real=(syn_fg_3_3d(syn_a,syn_b,syn_c))
```

The variable syn_d will be an array of size 14. The output will be in the following format,
(#, error, x_A, y_A, z_A, x_B, y_B, z_B, x_C, y_C, z_C, x_D, y_D, z_D)

- # = number of possible mechanisms
- error = 0 (no error), 1 (error)
- x_A, y_A, z_A= coordinates of point A,B,C,D

Known Limitations

1. None of the functions developed will be able to synthesis a parallelogram mechanism i.e. if $\theta_{1n} = \phi_{1n}$ for all n, the function will return the value of error equal to 1.
2. In four and five point synthesis, most of the time, the corresponding functions will return three (or two) mechanisms. Of these three (or two) mechanisms, only one will be the correct solution. The user will have to actually build these mechanisms in ADAMS/View, check the functional relationship between input and output angles by simulating the mechanism using ADAMS/Solver and thus discard invalid mechanisms.

Advanced Application

With little insight in inverse kinematics, the user should be able to design function generator mechanisms which are in series. The examples are some special inversions of six bar, eight bar and so on. The user would like to go to these mechanisms for it to satisfy functional relationship at more that five points or for some special case described in the next paragraph.

So far we have discussed function generators as mechanisms, which satisfies the input and output angular relationship. But it possible to design mechanisms with given relationship between input angle - output angle and input angular velocity - output angular velocity and all the possible combinations of these. The functions developed and presented in this paper can be used as a powerful tool to conduct these types of synthesis. The method can be extended to cases where mechanisms that have a given velocity ratio over a certain range of motion are required. Note that for a mechanism with no friction mechanical efficiency is equal to velocity ratio.

The functions presented here can be used to draw the Bermester curves (whenever applicable) within ADAMS/View, which are very useful in designing a mechanism with given practical constraints like space, size, initial position and locations.

User Interface

The user interface was built using the user-defined functions in ADAMS/View Most of the development was done on Windows NT. But the method is valid for platforms with by recompiling the synthesis routines on the required system.

Figure 3 shows the interface to enter the inputs required for three-point function generation synthesis, syn_fg_3. A Similar interface can be used for the other functions developed in this work. The fields under the INPUT SIDE accept the values entered and send it to the syn_fg_3 function. syn_fg_3 provides the output which is then drawn in the view. The four bar mechanism that is shown in the view consists of four markers on ground and an outline representing the three moving parts.

For the mechanism shown in Figure 3, the value of θ_0 and ϕ_0 is 90 degrees. The buttons under the DESIGN label allow the user to increment the value of θ_0 and ϕ_0 and observe other valid mechanisms. The fields in the Bermester Curve section draw the locus of point B (or 2) when the value of θ_0 is changed over the given Range with the given Increment. The same is true for point C (or 3) where the value of ϕ_0 is varied over the given Range with given Increment. The button "Create the assembly" will make the designed four bar as one entity so that the user can move and rotate the mechanism as to fit it in model at required location and orientation.

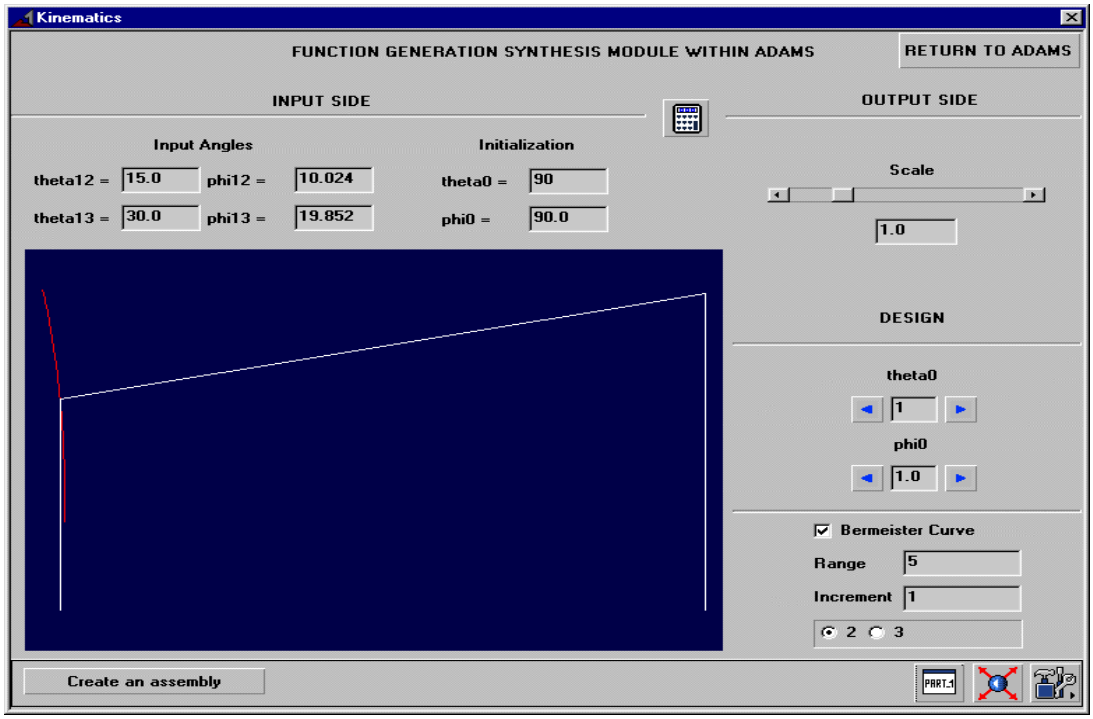

Figure 3: User Interface for syn_fg_3

Conclusions

The synthesis tool was created to gain awareness and observe the interest in inverse kinematics and synthesis among the computer simulation community. The tool that was created in conjunction with this paper is available to any ADAMS user upon request. It is currently available for Windows NT for ADAMS version 9.0.4 and higher. Contact the authors for further information and availability.

Contact Information

Nitin B. Talekar (Synthesis issues)
Application Engineer
Mechanical Dynamics, Inc.
2100 Commonwealth Blvd.
Ann Arbor, MI 48105
(734) 994-3800
{ HYPERLINK mailto:ntale@adams.com **}**

Todd C. DePauw (ADAMS issues)
Mechanical Dynamics, Inc.
2300 Commonwealth Blvd.
Ann Arbor, MI 48105
(734) 994-3800
{ HYPERLINK mailto:tdepa@adams.com **}**

Filename:	pap_talekar.doc
Directory:	\\Davinci\UserConf
Template:	C:\Program Files\Microsoft Office\Templates\Normal.dot
Title:	KINEMATIC SYNTHESIS (FUNCTION GENERATION) IN AVIEW
Subject:	
Author:	Marilyn Lee
Keywords:	
Comments:	
Creation Date:	06/25/98 2:00 PM
Change Number:	2
Last Saved On:	06/25/98 2:00 PM
Last Saved By:	Marilyn Lee
Total Editing Time:	0 Minutes
Last Printed On:	08/06/98 3:27 PM
As of Last Complete Printing	
Number of Pages:	10
Number of Words:	3,072 (approx.)
Number of Characters:	17,512 (approx.)