01 Apr 1991

Using Software to Demonstrate Kinematics of Gear Sets

John F. Hall

USING SOFTWARE TO DEMONSTRATE KINEMATICS OF GEAR SETS

John F. Hall

Abstract

The research was done in the area of developing software which could be used to demonstrate the motion of gears. To demonstrate this motion involved learning about programming using graphics. Engineering skills were used in the programming in various ways. One way was in developing and implementing the equations and theory. The more critical engineering aspect of this project is in determining what is effective in demonstrating the kinematic principles associated with gear sets.

The programs will be a helpful aid in showing students what is happening with the motion of gear sets. Not only will instructors be allowed to display drawings of gears quickly, but the important data will be displayed also. The instructor will be able to demonstrate how a slight change in a gear size will change the output in an instant. Overall this program will allow students to cover much more material pertaining to gear sets in much less time.

Introduction

The design and analysis of gear trains involves geometrical restraints, angular velocity ratios, and iteration. As a consequence of these factors visualization of the motion of gear sets is not always intuitive. This is particularly true for planetary gear sets in which some of the gear axes are not stationary.

The basic equation which governs gear ratios is:

$$e = \frac{\text{Product of teeth which are driving}}{\text{Product of teeth which are being driven}}$$

where 'e' is the train value for a set of gears which are in mesh with each other. This equation will allow the output speeds to be found since 'e' is proportional to gear speeds.

The first program, Gear Train Design, makes computations for a gear train and will also create a scaled display of a model designed by the user. There is no actual movement in Gear Train Design, however, the program is able to demonstrate the direction of motion from the computations made.

The second program, Planetary Gear Design, will make computations for a planetary gear set. This program involved displaying the rotation of gears. The program is still in the developing stage. However, the subroutines which animate the motion of the gears have been written as well as a subroutine which verifies data.

In the following text, the ways that these programs will be helpful to students of engineering will be shown by discussing the usage. The design of the program will also be discussed to give a better feel for the program.

Gear Train Design

Program Usage

Gear Train Design was written as a means to analyze and design a five-gear gear train. The user is asked to enter certain characteristics of the gears, such as size and input speed, and in doing so a system is generated. The program itself is user-friendly and self explained.

There are five gears available to the user to construct the gear train. There are four available gear axles which means that two gears must be placed on a single axle when using all five gears. The components can be seen on the following page. Gear 2 serves as the driver gear, while gears 3 and 4 share a common axis to control the speeds of the following gears, gear 5 can be used to give the desired direction for the output, and gear 6 is the output gear. The user has the option to use only the driver and output gears, but he/she may incorporate the other two axles which gears 3 and 4, and 5 rotate about. The user must use either both or neither of gears 3 and 4 since 3 is only in contact with gear 2 and gear 4 is only in contact with gear 5 or 6 (depending on whether or not gear 5 is used as an idler). The program is very descriptive in explaining use of the program functions. There are axis examples of gear trains, along with a description, which can be designed with Gear Train Design.

Not only could students use Gear Train Design to visualize the working part of a gear train and to make basic computations, but design is also possible. For example, say that one wanted to design a gear train with a certain train value; The user would start off by entering what is thought to give the desired output value. Once the output is obtained it can be checked to see if the value was correct. If the output is incorrect, say the train value was too low, the user could press the 'R' key to redesign the train. Once we have the desired values, we may want to reduce the size gear 5 for space purposes. However we also should consider wear on the axle bearings.

The user has the opportunity to interact with the program. To start with, the user will have the opportunity to enter gear sizes based on radius, diameter, or number of gear teeth. The next screen will give the user an organized table where values can be entered. Once the user has entered the values, the program makes some computations and then graphically displays the designed gear train with direction arrows and output table which contains calculated gear speeds. At this time the design part of the program becomes possible. At the output screen the user may enter 'R' to Redesign the gear train. For example, the user may leave out gear 5 to change the direction of the output, decrease the output speed by making gear 3 larger or gear 4 smaller, or completely take out gears 3 and 4. Minor to major modifications can be made an unlimited number of times. While in the design mode, the user may enter a piece of data which the program finds to be unsuitable (i.e. a negative gear value). However, the program will catch the error and alert the user with an explanation.

After the user has decided to exit the design mode (by pressing a key other than 'R' after new gear screen), there will be an opportunity to exit the program or to design another gear train. The program is capable of doing this by recording the last data input by the user and then allowing the user the option to reuse this data.

The help screens at the start of the program have made it possible for a student, even with little understanding of gear ratios, to use the program. Once the student has begun to use Gear Train Design, the student will be made aware of any mistakes he/she has made in entering data and why. Students can find this program to e a great time saver. Computations can be made in an instant as well as scaled diagrams. A copy of the screen can be made by pressing 'PrtSc' on the keyboard while the system is in the GRAPHICS mode.

Program Design

Gear Train Design was written using Borland Turbo Basic Software. The program has a main body, which consists mainly of gear train computations, and a series of subroutines at the end. Subroutines are used to generate the graphical displays. A brief description of each part of the program will help an observer to understand the logic behind the program. Some of the general parts of the program which pertain mostly to this project will be discussed.

At the beginning of the program we can observe help screens to explain the program. As we go down the page, we see a label entitled 'GEARTRAIN', this is the heart of the program. 'GEARTRAIN' consists of a structured IF-THEN statement which is about 75 lines long. In 'GEARTRAIN' I have actually made the computer determine if the user is entering useful data while at the same time having the capacity to make computations for the gear speeds. Towards the top of the 'GEARTRAIN' block we can see the calculations being made for a gear train which consists of all five gears. As we move even lower, the input error messages can be seen.

Some other features that I would like to point out is that one part of the program actually centers the gear train in the upper part of the screen, while another part of the program looks at the horizontal and vertical distances to allow the largest gear train possible without going off the screen.

Planetary Design

Program Usage

Planetary gear design is a program which will delve into the motion and computations associated with a planetary gear set. The gear sets will be based on the twelve types gear sets according to Levi.[1]

This program will be useful to the student as it will be very user friendly. The data entry subroutine will make sure that no 'bad' data is entered. In a planetary gear set, such as one where we have some gears in between a center gear and an outer gear with inside teeth, we must make sure that the number of teeth will mesh. For example, suppose that we have a sun gear of diameter 12 inches in the center of the gear train and on the outside we have a ring gear with a diameter of 24 inches. This means that the planet gear must be at a diameter of:

$$d = (24 - 12)/2 = 6 \text{ inches}$$

If the diameter is any bigger the gear will not fit into the mesh. On the other hand, if the planet gear is too small, and it is the only one in between, it will not work either. For this reason, the computer will only allow the user to enter the sizes of a certain amount of gears since the computer will be able to compute the sizes needed for the remaining. This also goes for the input of gear speeds. The program will only allow the user to enter a certain amount so that the remaining can be computed. I should mention that the program does let the user decide what gears are going to be assigned which values pertaining to size and speed.

Once the program is operational, the student will be able to see the motion of the gear set by both a front and side view. I felt as if both were needed due to the complex configuration of some planetary gear sets. When looking at the front view, one can only see the gears which are in front. However, in some cases, where we have many gears composing the planetary gear set, we may not be able to see the background work. By the side view, we can completely understand what is going on with the program. Although I have not yet experimented with showing a back view of the gear set, this is one of my future plans and it will be even better in helping one grasp an understanding of planetary gear sets. The program is useful for students to visualize what happens with the motion of a planetary gear set since the planet gears have rotation about their own axis, which is itself being carried by the arm in a circular path. This has been found to be a particularly difficult concept to visualize.

Program Design

To understand the usage of the program utilizing motion we will begin by analyzing what happens with the motion of a single gear. The gear looks like four arcs of 75o each. The arcs are separated by 15o. With these four arcs in a circular formation, we can change the orientation of the arcs to animate rotation. There is a part of the program which determines how much rotation is needed for each arc. I used about 0.1 radians for rotation since this proved to give a smooth rotating effect. Each time the program runs through the subroutine for the arcs, the value which represents the start of an arc is decreased by 0.1 while the value which represents the end of the arc is increased by 0.1 for counter-clockwise motion. Since counter-clockwise motion is positive for programming, motion in the clockwise direction requires another subroutine. The program will go through this loop of rotating the arcs for a number of times which corresponds to the speed at which the gear is turning. I should also note that the program could be changed to make the gear rotation increment correspond to the gear speed instead of using 0.1 radians. In doing this we would only need to send the program through the rotation procedure one time per gear, However, our rotation may not appear to be as smooth for incrementation which is large.

The next item which should be incorporated with gear rotation is actual movement of a gear from one location to another. This is will need to take into consideration the speed at which the arm of the planetary gear set is moving. By considering the amount of rotation of the arm we can determine how far a gear has moved around the center of the planetary gear set. We must now move the gears which are attached to the arm in a circular path and rotate it at the same time. What I have done is made the program look at each gear to determine if its location has changed since the last time it went through the rotation loop. If the gear has changed, the program completely erases the old and re-draws the new gear with the appropriate rotation.

I have established the screen format for the program and error traps for the data which is entered. The screen is laid out so that the data is entered in the left side of the screen while the gear is drawn in the right side of the screen. Two views of the planetary gear set will be seen, a front view and a side view.

Conclusions

Writing this program not only made use of technical skills that I have learned through my studies in mechanical engineering, but it also required intuitive skills. I had to decide which functions should be created to best aid the engineer who uses the program.

Whenever I wrote this program, I thought of a similar project I once had during a co-op term with General Motors. It pertained to having a certain amount of gears which could be used in a transfer cased to cause certain motion of the output gear. With a program like this I now see how much time one can save.

Future plans include animation of the motion of the gears. The capacity for this can be seen in the planetary gear program. Other future plans include allowing the user unlimited number of axles and gears to design a gear train.

Reference

Shigley J. E. and Uicker, J. J., Theory of Machines and Mechanisms. McGraw Hill Book Company, New York, New York, 1980.