3D Simulation of Inductive sensor sensing a cylindrical surface.

C1887856
PEIDONG WU

I hereby declare:

that except where reference has clearly been made to work by others, all the work presented in this report is my own work;
that it has not previously been submitted for assessment; and
that I have not knowingly allowed any of it to be copied by another student.

I understand that deceiving or attempting to deceive examiners by passing off the work of another as my own is plagiarism. I also understand that plagiarising the work of another or knowingly allowing another student to plagiarise from my work is against the University regulations and that doing so will result in loss of marks and possible disciplinary proceedings against me.

Signed: PEIDONG WU (electronic)

Date: 23/04/2021

Total number of words = 6539

Total number of pages = 51

Abstract:

The report details the author's analysis and simulation of the detection method of using a single mobile inductive sensor to sense cylindrical workpieces through Matlab. The previous work carried out by Professor Roger Grosvenor provided mathematical methods and sensor movement paths for the project, and suggested some possible misalignment situations. This work aims to develop a program that assists in processing sensor data and 3D simulation of cylindrical workpieces.

After completing the processing of the data obtained in the ideal state, the author also researched and completed the program of detecting and reporting the data of the misalignment of the cylindrical workpiece and the program of repairing the misaligned data. By processing the misaligned data, it is verified that the developed program can be used to process some misaligned data.

Acknowledgements:

The author would like to thank Professor Roger Grosvenor for his assistance and guidance throughout the project.

Index

1 Introduction

1.1 Project introduction

This third-year report detailed the project completed by the author to develop and analyze the data on the surface of the cylinder by the sensor sensor, and 3D simulation inductively sensing the program of the surface of the cylindrical workpiece. This project is based on a series of project research conducted by Professor Roger Grosvenor in the past.

The purpose of the past research is to provide a method for measuring the dimensions of mechanical parts on-line, and to provide sensor information for the sensor based machine tool management system (SBMTMS). A single fixed position proximity sensor will cause a series of measurement errors. At the same time, machine tool users do not agree to add additional sensors in the processing area to avoid measurement errors, so a single sensor moves through a loop path to provide sensor information. In past reports, Professor Roger Grosvenor has proposed related mathematical models and completed the simulation of sensor data.

This project provided the detection and analysis software for Professor Roger Grosvenor to rebuild the test bench. At the same time, the project only involves simulation and analysis of data to reconfirm the results of past experiments.

1.2 Measuring system

In the processing process, measurement is a very important part, it can reduce the number of defective parts. Measuring without stopping the machine is called in-process measurement. Compared with the measurement method that requires stopping the machine, it does not increase the processing time. At the same time, it can also provide information for the correction before cutting, which reduces the overall processing. The time required for a single part, and reduce the number of defective parts.

In the past work of Professor Roger Grosvenor, inductive sensors were used. The principle of the inductive sensor is to measure an object through the principle of electromagnetic induction, and the output current or voltage changes with the distance between the inductive sensor and the measuring object. For inductive sensors, it has a measurement range. When the inductive sensor is too far or too close to the sensing target, the output electrical signal will reach the maximum or minimum value, because the output electrical signal will not exceed the maximum and minimum values. Therefore, when the distance between the inductive sensor and the target exceeds the detection range, an invalid electrical signal will be output.

At the same time, two operating modes of the sensor have been proposed. The first is shown in Figure 1 The sensor moves on the plane formed by the x-axis and the y-axis. The other is shown in Figure 2. The sensor performs reciprocating linear motion on one axis of the y-axis.

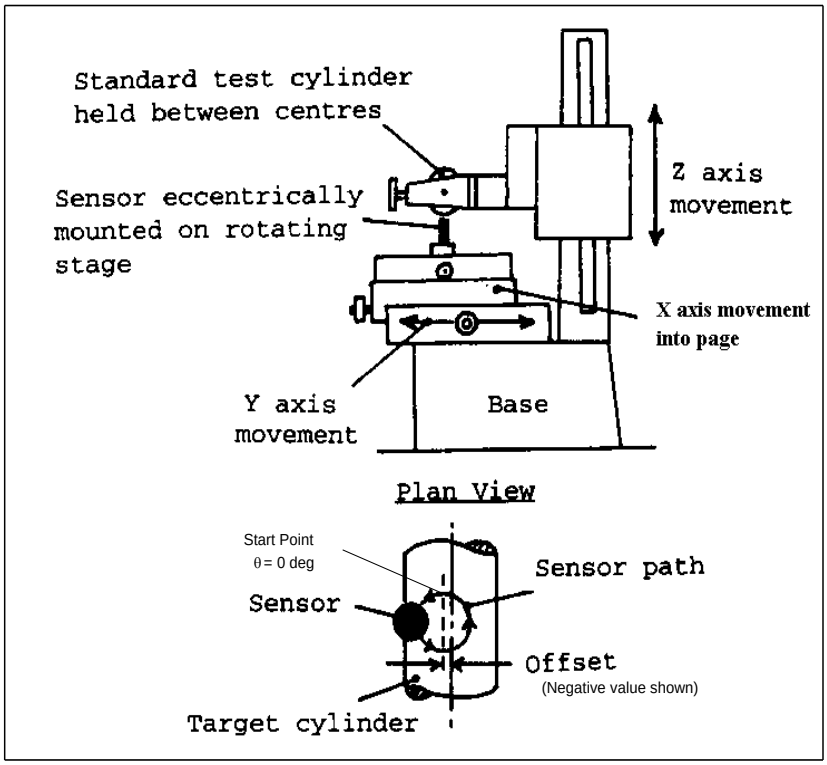

Figure 1. The sensor moves on the plane formed by the x-axis and the y-axis. (Grosvenor 1995)

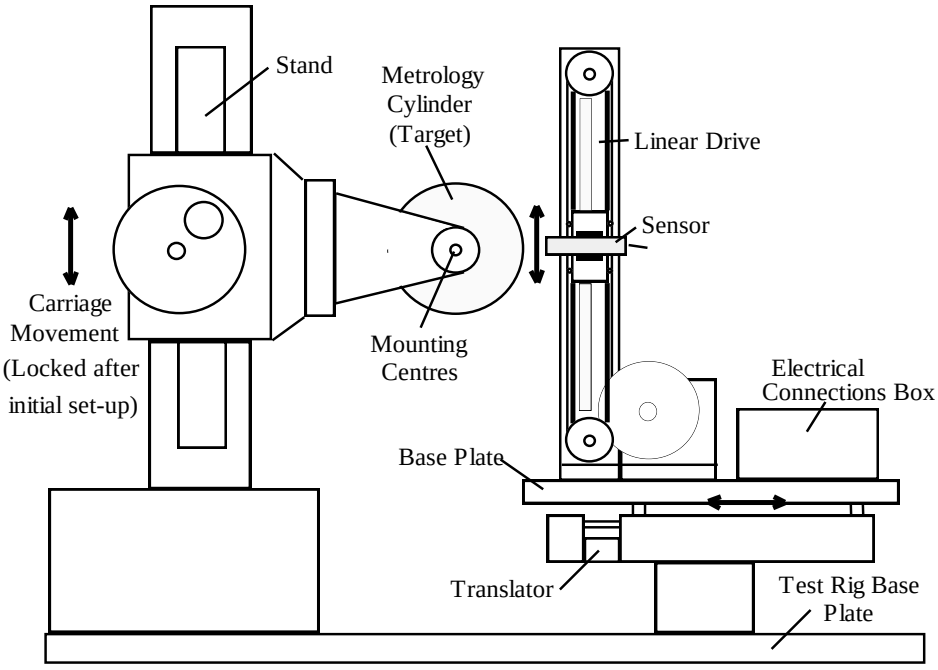

Figure 2. sensor performs reciprocating linear motion on one axis of the y-axis. (Grosvenor 1995)

In this project, software development and analysis will be aimed at analyzing the data of the second sensing method, and the discussion of the first sensing method will be launched in the discussion.

1.3 The definition of the coordinate system of the detection system

As shown in figure 1, when the cylindrical workpiece is in an ideal state, the x-axis of the detection system is parallel to the height of the cylinder and perpendicular to the two circular surfaces of the cylinder. The y-axis of the detection system is parallel to the two circular surfaces of the cylinder and parallel to the surface of the sensor. The z-axis of the detection system is parallel to the two prototype surfaces of the cylindrical workpiece and perpendicular to the surface of the sensor. The distance from the surface of the sensor to the surface of the cylindrical workpiece is g.

2 Program development

2.1 Matlab programming

2.1.1 the reason for choosing Matlab

When the initial experiment was proposed, computer science has not developed to its current state. Prof. Roger Grosvenor used MATHCAD in past experiments, but due to the limitations of past computer technology, a computer simulation of this project was proposed.

When choosing a program to assist analysis, Matlab and Python were both considered. Python has a richer toolkit, richer functions and a wider range of applications. But in the application of this experiment, only need to analyze the experimental data obtained after sensor sensing, so Matlab can make the program more concise.

Matlab can easily perform fast Fourier transform and other large data calculations. These calculations in other languages such as python and C++ may require you to define very complex functions yourself or call third-party toolkits, which will undoubtedly increase Workload. At the same time, Matlab also comes with the function of drawing 3D images. After data analysis is completed, 3D simulation can be completed without a third-party toolkit or other software.

2.1.2 User interface design

In order to be able to complete the data analysis more conveniently and intuitively, a User Interface for reading the file and displaying the diameter of the cylinder obtained after the analysis was designed. As shown in figure 3, there is a box used to display the address of the opened file in the first row. Click the button behind it to find the required file in the folder. In the second line, you can choose the method used for sensing. In the past experiments of Prof. Roger Grosvenor, he once proposed two methods for moving the inductive sensor. The signal waves obtained by the two methods will be different, so this box for selecting the sensing method is created. The program can be run after clicking the button in the second row. The final calculated diameter of the cylinder will be displayed in the bottom box.

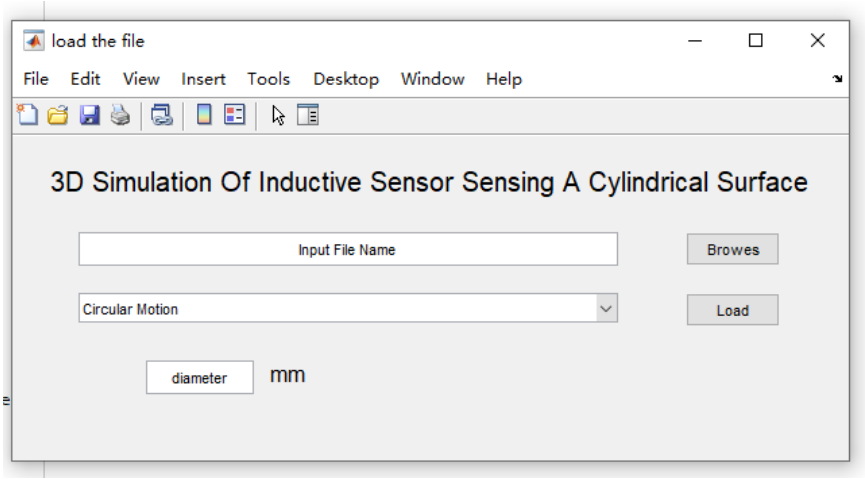

Figure 3. User Interface

First, set a 620*240 (unit pixel) window without title and number title through the figure() function, and use the set() function to make the window invisible. The title of the page is created through the uicontrol() function, the box that displays the file address, the menu that selects the sensor movement mode, two bottons, and the box that displays the calculated diameter of the cylinder. After the creation is completed, the components in the same row are aligned through the align() function to prevent the problem of unaligned components in the user interface control created before. Change the units of the component to normalized so that the component will automatically resize when the window is zoomed. After completing the creation, use the set() function to change the title of the figure to load the file. Then use movegui() to move the window to the center of the screen. After finishing these adjustments, use the set() function to adjust the window to be visible.

2.1.3 the method of data analysis

When the sensor only moves on the y-axis, the image of the data is shown in Figure 4. Analyzing the signal output by the sensor can be analyzed by Fourier Transform, and the amplitudes can be calculated.

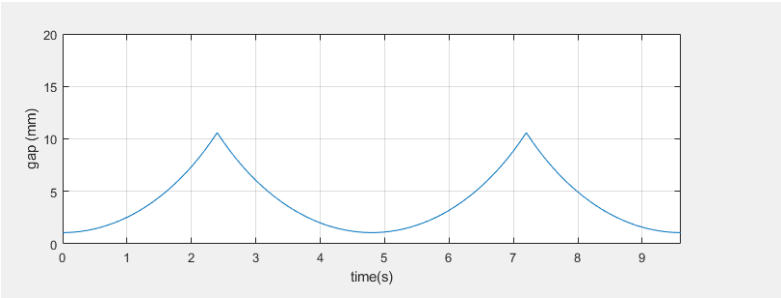

Figure 4. signal output by the sensor

To calculate the diameter of a cylindrical workpiece, the circle equation is used:

$$(x-a)^2+(y-b)^2=R^2 \quad (1)$$

In the analysis, the position of the center of the circle (a, b) and the radius R of the cylinder are both unknown, so it is necessary to select 3 points in the data for analysis.

2.1.4 data analysis program design

Set the callback function, and control the components on the user interface through the callback function. In the callback function of hFilePath, use function get() to directly store the b in the box as a variable named theFileName, so that the user can directly open the file by typing in the file name. In the callback function of hTypeOfMethod, directly save the string in the menu as a variable named MethodType.

Use callback to control hFileBrowes, use function uigetfile() to open a window for newly selected files, select the desired file from the window, store the file name in theFileName, and display the path of the file in the hFilePath box.

Open the file storing the data through function importdata() and save the data in a different array. The file includes the elapsed time, the displacement of the sensor on the y-axis, and the distance value sensed by the sensor. Through function find() to find the position of the data of the 0 point of the y-axis that the sensor passes for the third time, the data obtained by the sensor running a cycle can be intercepted.

After the storage is complete, determine the sample frequency and signal duration, and calculate the display number of points that will be fast Fourier triansformed. Next, use function fft() to perform a fast Fourier transform on the data, and use function abs() to obtain their absolute values.

After that, calculate the frequency axis resolution and frequency range (up to Nyquist limit), which can be used to determine the frequency scale. Finally, the amplitudes are calculated and the analysis of the signal is completed.

Display the data through function plot() and function stem() on a new window called graphs that breaks into function figure()

After completing the signal analysis, the next step is to calculate the diameter of the cylindrical workpiece being sensed. A function named findoutR was created by the author. Input the two arrays of the displacement of the sensor on the y-axis and the distance value sensed by the sensor, and the diameter will be the output value. In this function, first find out the positions of the maximum and minimum values in the array, and intercept the calculation between the first maximum value and the minimum value that is closest to the maximum value before the maximum value. Create three ternary quadratic equations with the maximum, minimum, and intercepted values. Use function solve() to solve. Because the radius of the circle obtained at this time is not a double type data, use function double() to convert the data into double type data for storage. Finally use the radius of the stored circle to calculate the diameter. And output the diameter value as string type data in the box of hDiameterOfCylinder.

Finally, calculate the x-axis, y-axis and z-axis values of the cylindrical workpiece through function cylinder(), and draw a 3D simulation diagram through function surf().

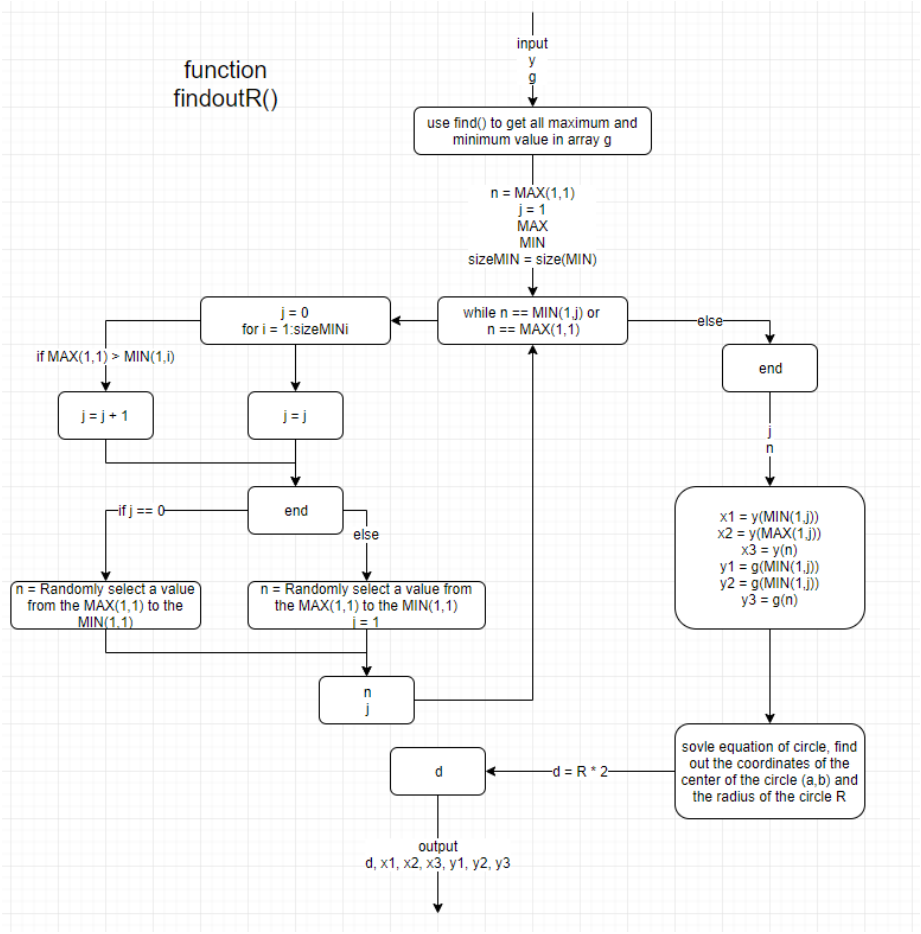

Figure 5, function findoutR()

2.2 Program improvement based on actual experimental data

2.2.1 Questions and assumptions

In the past experiment of Prof. Roger Grosvenor, the misalignment and a series of problems that may occur in the experiment are listed. After summarizing, these can be analyzed through the coordinate axis. For cylinders, the misalignment on the x-axis will not cause errors, so there is no need to consider the impact of this problem. Although this dislocation will affect the processing, it cannot be detected because of the sensing method.

The misalignment that occurs on the z-axis will cause the data sensed by the sensor to increase or decrease at the same time, as shown in Figure 6. Because the sensor used in the experiment is an inductive sensor, the nature of the inductive sensor determines that its sensing range has a maximum and a minimum. The misalignment on the z-axis is likely to cause the distance between the sensor and the target surface to be out of the

sensing range. Inside. In this case, the output data will be as shown in Figure 6. If the program does not eliminate these erroneous data when analyzing the data, the real data of the target will not be obtained.

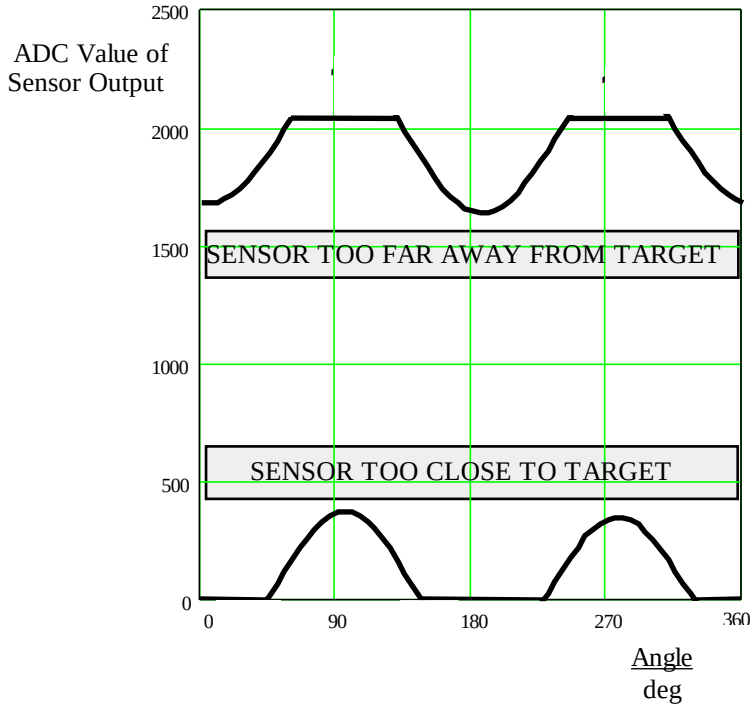

Figure 6. misalignment occurs on the z-axis (Grosvenor 1995)

When the misalignment occurs on the y-axis, as shown in Figure 7, the starting point is changed for the sensor, but this will not affect the data analysis. However, this kind of misalignment will have a certain impact on processing, so it should still be detected and reported.

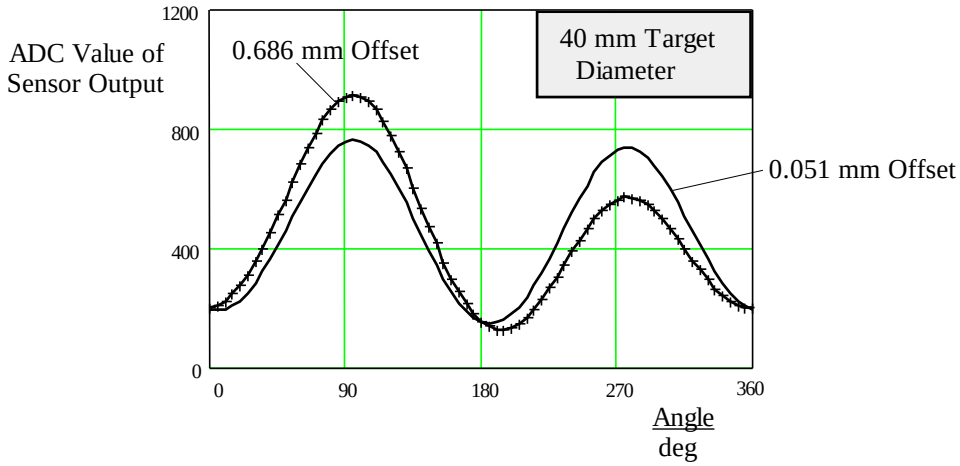

Figure 7 misalignment occurs on the y-axis (Grosvenor 1995)

Finally, there is another misalignment that may occur, that is, the coordinate axis of the cylinder being sensed is deflected, and the angle of deviation is generated from the sensor set by the sensing device. In this case, the maximum and minimum values of the data output by the sensor will deviate, as shown in Figure 8 and Figure 9. In this case, the analyzed data will have obvious errors. When selecting a part of the value, you may get an excessively large diameter or an excessively small diameter.

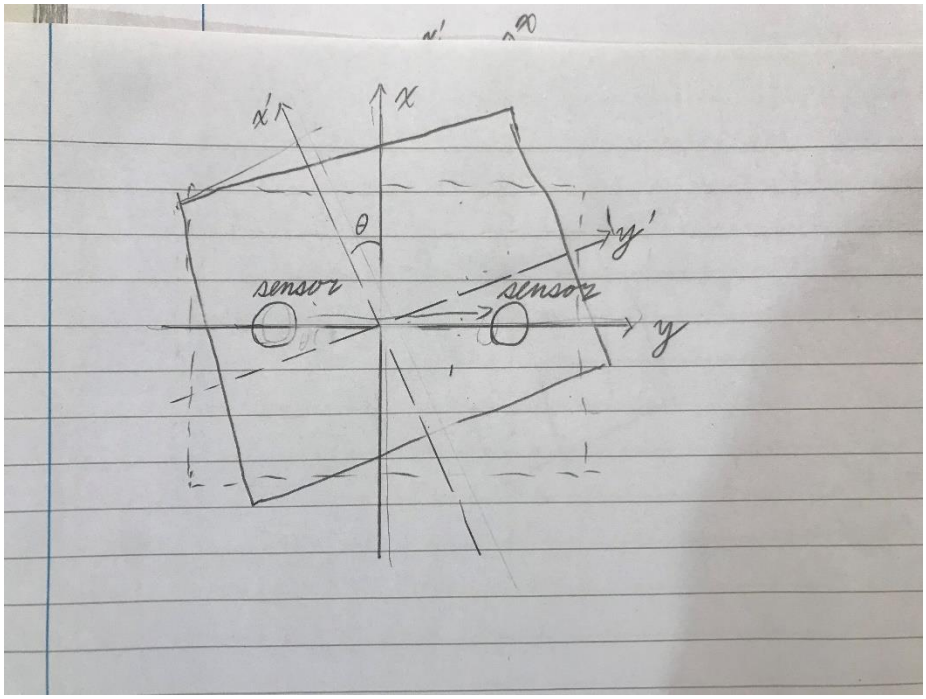

Figure 8. coordinate axis of the cylinder being sensed is deflected A

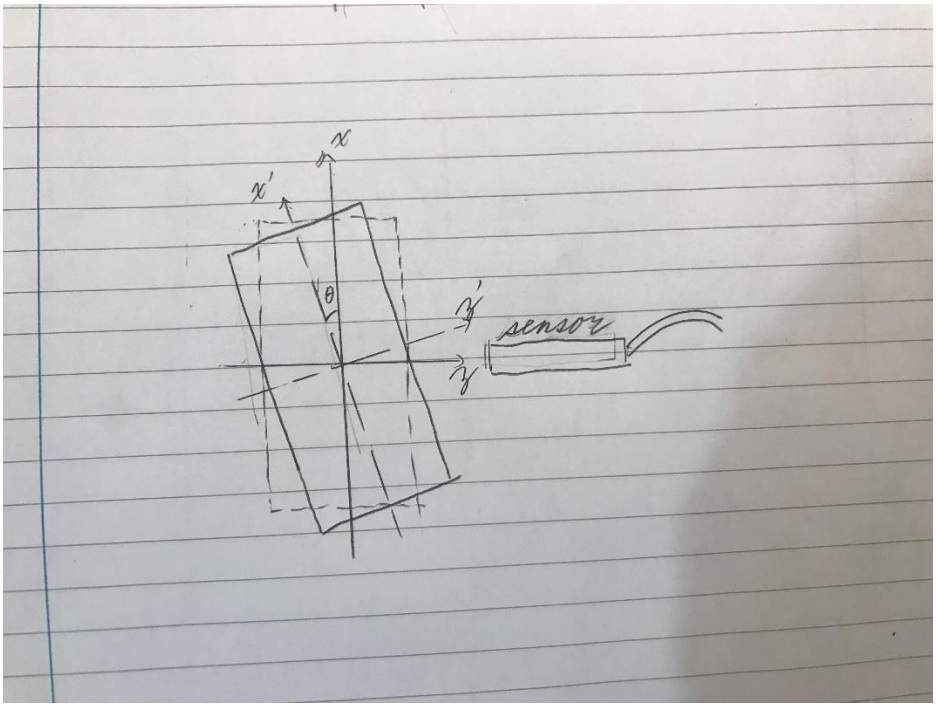

Figure 9. coordinate axis of the cylinder being sensed is deflected B

2.2.2 the detection method and program design using Matlab

As mentioned in 2.1, in order to assist the analysis, several functions for detecting sensor problems have been created. Firstly, by detecting the continuous maximum or minimum value, it is determined whether the distance between the sensor and the target surface is not within the sensing range. The locations of these values are determined and deleted from the data, and the remaining points can be used for analysis.

In order to deal with this problem, a function CheckOutofRange() to detect and extract these values was created. This function needs to input the displacement on the y-axis obtained by a cycle of the intercepted sensor movement and the measured distance value. array. Use function if() to determine whether the point is out of the data measurement range, and store them in two arrays, gOutOfRange and yOutOfRange. By detecting whether the distance value obtained by sensing one point before or one point after the detected point is equal to the distance value obtained by sensing this point. If they are equal, it means that this point is a value outside the measurement range. The way to judge whether the distance to the sensor is too far or too close is to detect the point back to the range, as shown in Figure (8), when the distance value of this point is less than the distance value of the next point, what happens at this time is the distance Too close, if the distance value of this point is greater than the distance value of the next point, then what happens at this time is that the distance is too far. If an out-of-range situation occurs, this function will judge whether the distance is too close or too

far, and output the corresponding string type text to remind the user, and the YorN variable will become 1 and be output.

When the output of the YorN variable is 1, all out-of-range values will be deleted through the function DeleteOutRang() defined by the author. The working principle of this function is to store the values in the gpercycle variable but not in the gOutOfRange in Newg by comparing the values of the two variables gpercycle and gOutOfRange. The reason for choosing to use gpercycle and gOutOfRange for comparison is that the distance output of the sensor will not be 0, which can avoid errors. At the same time, because the values in gpercycle and yellowcycle are related to each other. So store the value stored in Newg corresponding to the value in the yellowcycle to Newy. In this way, all OutRange values can be deleted.

By comparing the distance on the y-axis from the position of the comparison starting point to the point on the cylindrical surface that is closest to the sensor, the misalignment of the y-axis can be obtained.

In order to solve this problem, function yMisalignment() was created. Its principle is to find the nearest value of the sensor to the surface of the cylindrical workpiece through function min(), and select the point where this value is located. The y-axis coordinate of this point is from the sensor. The distance of the starting point is the misalignment of this data on the y-axis.

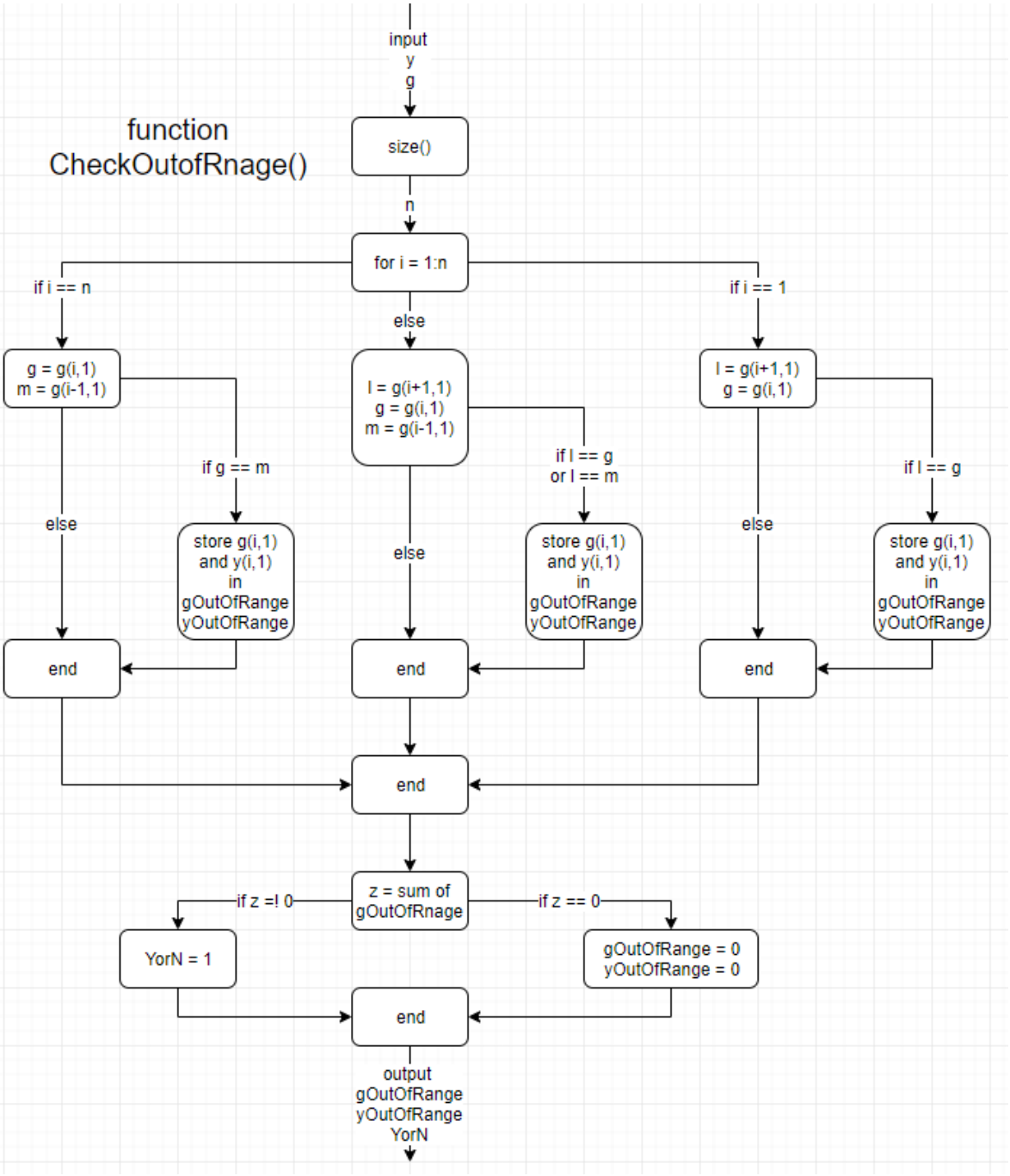

Figure 10. function CheckOutofRange()

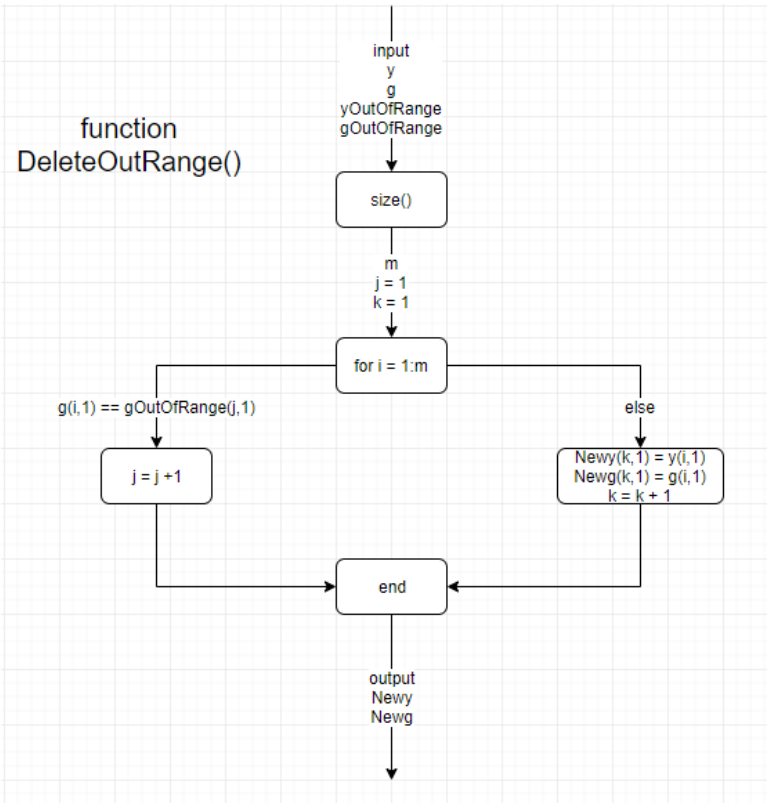

Figure 11. function DeleteOutRange()

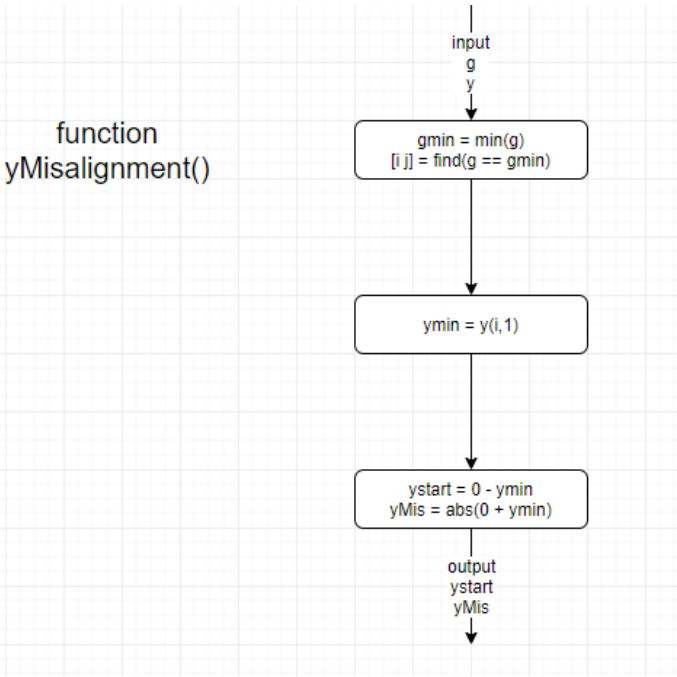

Figure 12. function yMisalignment()

3 Results

3.1 Results and analysis of the initial data analysis

3.1.1 the problems that occurred during the initial data analysis

After the author directly used the data in the table given by Prof. Roger Grosvenor, the analysis results showed serious problems. The problem is shown in the table 1 . The diameter of the cylindrical workpiece obtained by the analysis is seriously larger than the actual diameter of the cylindrical workpiece. At the same time, because in the custom function findoutR() used for calculation, the third point used to calculate the actual diameter of the cylindrical workpiece is randomly selected within a range. With the different selected points, the calculated diameter of the cylindrical workpiece Also changes with it. The author brings all the points into function findoutR() for calculation, and the result is shown in table 1.

x1	y1	x2	y2	x3	y3	d
0	1.0683	24	10.5723	1	1.0848	70.52275
0	1.0683	24	10.5723	2	1.1343	70.96887
0	1.0683	24	10.5723	3	1.2168	71.44852
0	1.0683	24	10.5723	4	1.3323	71.96178
0	1.0683	24	10.5723	5	1.4808	72.50875
0	1.0683	24	10.5723	6	1.6623	73.08952
0	1.0683	24	10.5723	7	1.8768	73.7042
0	1.0683	24	10.5723	8	2.1243	74.35288
0	1.0683	24	10.5723	9	2.4048	75.03566
0	1.0683	24	10.5723	10	2.7183	75.75264
0	1.0683	24	10.5723	11	3.0648	76.50393
0	1.0683	24	10.5723	12	3.4443	77.28962
0	1.0683	24	10.5723	13	3.8568	78.10981
0	1.0683	24	10.5723	14	4.3023	78.96461
0	1.0683	24	10.5723	15	4.7808	79.8541
0	1.0683	24	10.5723	16	5.2923	80.77837
0	1.0683	24	10.5723	17	5.8368	81.73753
0	1.0683	24	10.5723	18	6.4143	82.73166
0	1.0683	24	10.5723	19	7.0248	83.76084
0	1.0683	24	10.5723	20	7.6683	84.82517
0	1.0683	24	10.5723	21	8.3448	85.92471
0	1.0683	24	10.5723	22	9.0543	87.05954
0	1.0683	24	10.5723	23	9.7968	88.22974
0	1.0683	-24	10.5723	-1	1.0848	70.52275
0	1.0683	-24	10.5723	-2	1.1343	70.96887
0	1.0683	-24	10.5723	-3	1.2168	71.44852
0	1.0683	-24	10.5723	-4	1.3323	71.96178
0	1.0683	-24	10.5723	-5	1.4808	72.50875
0	1.0683	-24	10.5723	-6	1.6623	73.08952
0	1.0683	-24	10.5723	-7	1.8768	73.7042
0	1.0683	-24	10.5723	-8	2.1243	74.35288
0	1.0683	-24	10.5723	-9	2.4048	75.03566
0	1.0683	-24	10.5723	-10	2.7183	75.75264
0	1.0683	-24	10.5723	-11	3.0648	76.50393
0	1.0683	-24	10.5723	-12	3.4443	77.28962
0	1.0683	-24	10.5723	-13	3.8568	78.10981
0	1.0683	-24	10.5723	-14	4.3023	78.96461
0	1.0683	-24	10.5723	-15	4.7808	79.8541
0	1.0683	-24	10.5723	-16	5.2923	80.77837
0	1.0683	-24	10.5723	-17	5.8368	81.73753
0	1.0683	-24	10.5723	-18	6.4143	82.73166
0	1.0683	-24	10.5723	-19	7.0248	83.76084
0	1.0683	-24	10.5723	-20	7.6683	84.82517
0	1.0683	-24	10.5723	-21	8.3448	85.92471
0	1.0683	-24	10.5723	-22	9.0543	87.05954
0	1.0683	-24	10.5723	-23	9.7968	88.22974

Table 1.a. Problematic result

x1	y1	x2	y2	x3	y3	d
0	1.0683	-24	10.5723	1	1.0848	69.73072
0	1.0683	-24	10.5723	2	1.1343	69.38466
0	1.0683	-24	10.5723	3	1.2168	69.07179
0	1.0683	-24	10.5723	4	1.3323	68.79205
0	1.0683	-24	10.5723	5	1.4808	68.54537
0	1.0683	-24	10.5723	6	1.6623	68.3317
0	1.0683	-24	10.5723	7	1.8768	68.151
0	1.0683	-24	10.5723	8	2.1243	68.0032
0	1.0683	-24	10.5723	9	2.4048	67.88829
0	1.0683	-24	10.5723	10	2.7183	67.80623
0	1.0683	-24	10.5723	11	3.0648	67.757
0	1.0683	-24	10.5723	12	3.4443	67.7406
0	1.0683	-24	10.5723	13	3.8568	67.757
0	1.0683	-24	10.5723	14	4.3023	67.80623
0	1.0683	-24	10.5723	15	4.7808	67.88829
0	1.0683	-24	10.5723	16	5.2923	68.0032
0	1.0683	-24	10.5723	17	5.8368	68.151
0	1.0683	-24	10.5723	18	6.4143	68.3317
0	1.0683	-24	10.5723	19	7.0248	68.54537
0	1.0683	-24	10.5723	20	7.6683	68.79205
0	1.0683	-24	10.5723	21	8.3448	69.07179
0	1.0683	-24	10.5723	22	9.0543	69.38466
0	1.0683	-24	10.5723	23	9.7968	69.73072
0	1.0683	24	10.5723	-1	1.0848	69.73072
0	1.0683	24	10.5723	-2	1.1343	69.38466
0	1.0683	24	10.5723	-3	1.2168	69.07179
0	1.0683	24	10.5723	-4	1.3323	68.79205
0	1.0683	24	10.5723	-5	1.4808	68.54537
0	1.0683	24	10.5723	-6	1.6623	68.3317
0	1.0683	24	10.5723	-7	1.8768	68.151
0	1.0683	24	10.5723	-8	2.1243	68.0032
0	1.0683	24	10.5723	-9	2.4048	67.88829
0	1.0683	24	10.5723	-10	2.7183	67.80623
0	1.0683	24	10.5723	-11	3.0648	67.757
0	1.0683	24	10.5723	-12	3.4443	67.7406
0	1.0683	24	10.5723	-13	3.8568	67.757
0	1.0683	24	10.5723	-14	4.3023	67.80623
0	1.0683	24	10.5723	-15	4.7808	67.88829
0	1.0683	24	10.5723	-16	5.2923	68.0032
0	1.0683	24	10.5723	-17	5.8368	68.151
0	1.0683	24	10.5723	-18	6.4143	68.3317
0	1.0683	24	10.5723	-19	7.0248	68.54537
0	1.0683	24	10.5723	-20	7.6683	68.79205
0	1.0683	24	10.5723	-21	8.3448	69.07179
0	1.0683	24	10.5723	-22	9.0543	69.38466
0	1.0683	24	10.5723	-23	9.7968	69.73072

Table 1.b. Problematic result

3.1.2 analysis of the problem

After confirmation, the data in the table is calculated by a quadratic function, and the data is not a perfect circle, so this problem occurs. In order to solve this problem, the author determined a set of data for testing through calculation. By equation 2, the radius of the circle is set to 35mm, the minimum distance between the surface of the cylindrical workpiece and the sensor is 1.0683, and the y-axis is not misaligned. So the coordinates of the center of the circle are (0, 36.0683). So the circle equation is:

$$(x-0)^2+(y-36.0683)^2=1225 \tag{2}$$

The unknown number x in the equation corresponds to the y axis in the sensor's coordinates, and the unknown number y corresponds to the distance between the cylinder surface and the sensor surface on the z axis. When the sensor moves on the y-axis, the distance between the surface of the cylinder and the surface of the sensor also changes. So the value of y can be determined by the value of x. The relationship between X and y is as follows:

$$y=\sqrt{1225-(x-0)^2}+36.0683 \tag{3}$$

Through equation (3), a set of ideal state data can be calculated.

3.1.3 analysis conclusions and final data analysis results

Assuming that the range of the sensor moving on the y-axis is (-24, 24), the sensor data is shown in the following table 2:

number	time(s)	y(mm)	g(mm)		time(s)	y(mm)	g(mm)
1	0	0	1.0683	49	4.8	0	1.0683
2	0.1	1	1.0826	50	4.9	-1	1.0826
3	0.2	2	1.1255	51	5	-2	1.1255
4	0.3	3	1.1971	52	5.1	-3	1.1971
5	0.4	4	1.2976	53	5.2	-4	1.2976
6	0.5	5	1.4273	54	5.3	-5	1.4273
7	0.6	6	1.5864	55	5.4	-6	1.5864
8	0.7	7	1.7754	56	5.5	-7	1.7754
9	0.8	8	1.9948	57	5.6	-8	1.9948
10	0.9	9	2.2452	58	5.7	-9	2.2452
11	1	10	2.5273	59	5.8	-10	2.5273
12	1.1	11	2.8418	60	5.9	-11	2.8418
13	1.2	12	3.1897	61	6	-12	3.1897
14	1.3	13	3.5721	62	6.1	-13	3.5721
15	1.4	14	3.9903	63	6.2	-14	3.9903
16	1.5	15	4.4455	64	6.3	-15	4.4455
17	1.6	16	4.9395	65	6.4	-16	4.9395
18	1.7	17	5.4742	66	6.5	-17	5.4742
19	1.8	18	6.0516	67	6.6	-18	6.0516
20	1.9	19	6.6744	68	6.7	-19	6.6744
21	2	20	7.3455	69	6.8	-20	7.3455
22	2.1	21	8.0683	70	6.9	-21	8.0683
23	2.2	22	8.847	71	7	-22	8.847
24	2.3	23	9.6865	72	7.1	-23	9.6865
25	2.4	24	10.5928	73	7.2	-24	10.5928
26	2.5	23	9.6865	74	7.3	-23	9.6865
27	2.6	22	8.847	75	7.4	-22	8.847
28	2.7	21	8.0683	76	7.5	-21	8.0683
29	2.8	20	7.3455	77	7.6	-20	7.3455
30	2.9	19	6.6744	78	7.7	-19	6.6744
31	3	18	6.0516	79	7.8	-18	6.0516
32	3.1	17	5.4742	80	7.9	-17	5.4742
33	3.2	16	4.9395	81	8	-16	4.9395
34	3.3	15	4.4455	82	8.1	-15	4.4455
35	3.4	14	3.9903	83	8.2	-14	3.9903
36	3.5	13	3.5721	84	8.3	-13	3.5721
37	3.6	12	3.1897	85	8.4	-12	3.1897
38	3.7	11	2.8418	86	8.5	-11	2.8418
39	3.8	10	2.5273	87	8.6	-10	2.5273
40	3.9	9	2.2452	88	8.7	-9	2.2452
41	4	8	1.9948	89	8.8	-8	1.9948
42	4.1	7	1.7754	90	8.9	-7	1.7754
43	4.2	6	1.5864	91	9	-6	1.5864
44	4.3	5	1.4273	92	9.1	-5	1.4273
45	4.4	4	1.2976	93	9.2	-4	1.2976
46	4.5	3	1.1971	94	9.3	-3	1.1971
47	4.6	2	1.1255	95	9.4	-2	1.1255
48	4.7	1	1.0826	96	9.5	-1	1.0826
				97	9.6	0	1.0683

Table 2 sensor data

The result of processing using the program written by Matlab is shown in figure 13.

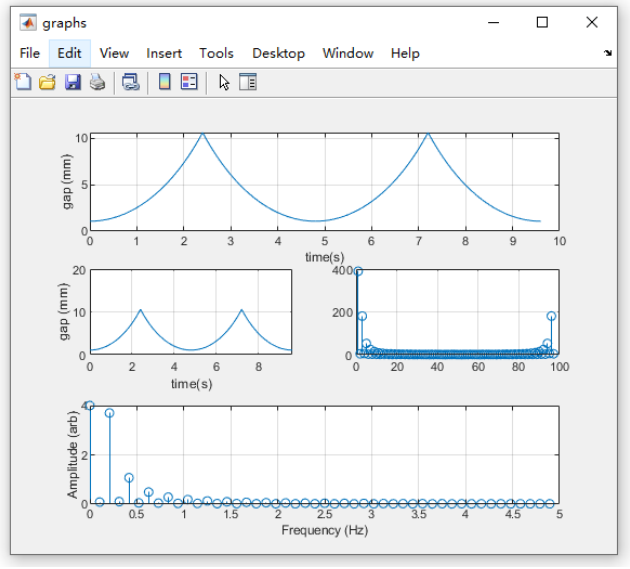

Figure 13 result

The dislocation information sent by this group of data is shown in figure 14.

```
>> Main_program
ystart =
     0
yMis =
     0
ans =
     0
ans =
     0
```

Figure 14 misalignment information

The result of running it 5 times is shown in table 3.

n of test	y1(mm)	y2(mm)	y3(mm)	g1(mm)	g2(mm)	g3(mm)	d(mm)
1	0	24	3	1.0683	10.5928	1.1971	69.9996
2	0	24	21	1.0683	10.5928	8.0683	70.0007
3	0	24	13	1.0683	10.5928	3.5721	69.9991
4	0	24	8	1.0683	10.5928	1.9948	69.9986
5	0	24	10	1.0683	10.5928	2.5373	70.0007

Table 3 result

As shown in the table, the error of the diameter of the circle is within ±0.0015mm, and the calculated result can be considered acceptable. The 3D simulation of the cylindrical surface is shown in figure 15.

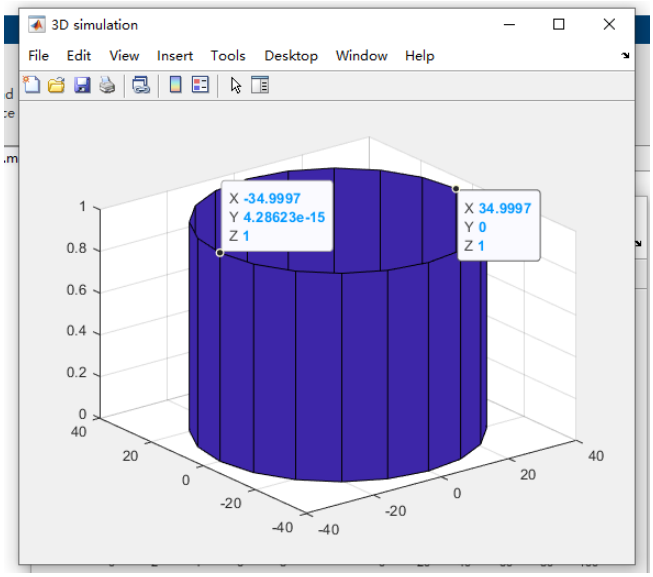

Figure 15 The 3D simulation

3.2 Analyze the misaligned data

3.2.1 Detection and analysis of data beyond the measurement range

3.2.1.1 This happens when the cylindrical surface is too far away from the sensor

The table 4 shows the data when the cylinder surface is too far away from the sensor

n of test	time(s)	y(mm)	g(mm)	n of test	time(s)	y(mm)	g(mm)
1	0	0	1.0683	50	4.9	-1	1.0826
2	0.1	1	1.0826	51	5	-2	1.1255
3	0.2	2	1.1255	52	5.1	-3	1.1971
4	0.3	3	1.1971	53	5.2	-4	1.2976
5	0.4	4	1.2976	54	5.3	-5	1.4273
6	0.5	5	1.4273	55	5.4	-6	1.5864
7	0.6	6	1.5864	56	5.5	-7	1.7754
8	0.7	7	1.7754	57	5.6	-8	1.9948
9	0.8	8	1.9948	58	5.7	-9	2.2452
10	0.9	9	2.2452	59	5.8	-10	2.5273
11	1	10	2.5273	60	5.9	-11	2.5273
12	1.1	11	2.5273	61	6	-12	2.5273
13	1.2	12	2.5273	62	6.1	-13	2.5273
14	1.3	13	2.5273	63	6.2	-14	2.5273
15	1.4	14	2.5273	64	6.3	-15	2.5273
16	1.5	15	2.5273	65	6.4	-16	2.5273
17	1.6	16	2.5273	66	6.5	-17	2.5273
18	1.7	17	2.5273	67	6.6	-18	2.5273
19	1.8	18	2.5273	68	6.7	-19	2.5273
20	1.9	19	2.5273	69	6.8	-20	2.5273
21	2	20	2.5273	70	6.9	-21	2.5273
22	2.1	21	2.5273	71	7	-22	2.5273
23	2.2	22	2.5273	72	7.1	-23	2.5273
24	2.3	23	2.5273	73	7.2	-24	2.5273
25	2.4	24	2.5273	74	7.3	-23	2.5273
26	2.5	23	2.5273	75	7.4	-22	2.5273
27	2.6	22	2.5273	76	7.5	-21	2.5273
28	2.7	21	2.5273	77	7.6	-20	2.5273
29	2.8	20	2.5273	78	7.7	-19	2.5273
30	2.9	19	2.5273	79	7.8	-18	2.5273
31	3	18	2.5273	80	7.9	-17	2.5273
32	3.1	17	2.5273	81	8	-16	2.5273
33	3.2	16	2.5273	82	8.1	-15	2.5273
34	3.3	15	2.5273	83	8.2	-14	2.5273
35	3.4	14	2.5273	84	8.3	-13	2.5273
36	3.5	13	2.5273	85	8.4	-12	2.5273
37	3.6	12	2.5273	86	8.5	-11	2.5273
38	3.7	11	2.5273	87	8.6	-10	2.5273
39	3.8	10	2.5273	88	8.7	-9	2.2452
40	3.9	9	2.2452	89	8.8	-8	1.9948
41	4	8	1.9948	90	8.9	-7	1.7754
42	4.1	7	1.7754	91	9	-6	1.5864
43	4.2	6	1.5864	92	9.1	-5	1.4273
44	4.3	5	1.4273	93	9.2	-4	1.2976
45	4.4	4	1.2976	94	9.3	-3	1.1971
46	4.5	3	1.1971	95	9.4	-2	1.1255
47	4.6	2	1.1255	96	9.5	-1	1.0826
48	4.7	1	1.0826	97	9.6	0	1.0683
49	4.8	0	1.0683				

Table 4 sensor data

As shown in the table, when the sensor displacement is greater than or equal to ±10mm, the distance between the sensor output and the surface of the cylindrical workpiece is maintained at 2.5273mm, and the distance between the sensor and the workpiece surface exceeds the working range of the sensor.

The wave and characteristics of the output signal at this time are shown in Figure 16:

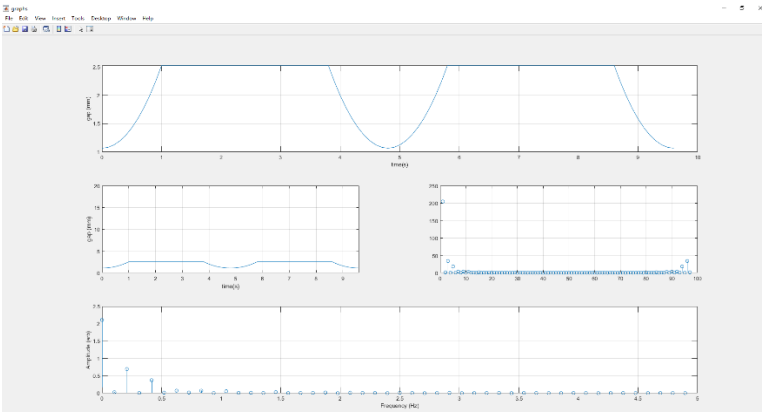

Figure 16 result

The out-of-range values detected by the program are shown in Figure 17 and Figure 18

```
ystart =

     0


yMis =

     0


ans =

  Columns 1 through 13

    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273

  Columns 14 through 26

    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273

  Columns 27 through 39

    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273

  Columns 40 through 52

    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273

  Columns 53 through 65

    2.5273    2.5273    2.5273    2.5273    2.5273    2.5273         0         0         0         0         0         0         0

  Columns 66 through 78

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 79 through 91

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 92 through 97

         0         0         0         0         0         0
```

Figure 17 misalignment information

```
ans =

  Columns 1 through 21

    10    11    12    13    14    15    16    17    18    19    20    21    22    23    24    23    22    21    20    19    18

  Columns 22 through 42

    17    16    15    14    13    12    11    10   -10   -11   -12   -13   -14   -15   -16   -17   -18   -19   -20   -21   -22

  Columns 43 through 63

   -23   -24   -23   -22   -21   -20   -19   -18   -17   -16   -15   -14   -13   -12   -11   -10     0     0     0     0     0

  Columns 64 through 84

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 85 through 97

     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Figure 18 misalignment information

The valid data after excluding out-of-range values is shown in figure 19.

```
ans =

  Columns 1 through 21

     0     1     2     3     4     5     6     7     8     9     9     8     7     6     5     4     3     2     1     0    -1

  Columns 22 through 39

    -2    -3    -4    -5    -6    -7    -8    -9    -9    -8    -7    -6    -5    -4    -3    -2    -1     0

ans =

  Columns 1 through 13

    1.0683    1.0826    1.1255    1.1971    1.2976    1.4273    1.5864    1.7754    1.9948    2.2452    2.2452    1.9948    1.7754

  Columns 14 through 26

    1.5864    1.4273    1.2976    1.1971    1.1255    1.0826    1.0683    1.0826    1.1255    1.1971    1.2976    1.4273    1.5864

  Columns 27 through 39

    1.7754    1.9948    2.2452    2.2452    1.9948    1.7754    1.5864    1.4273    1.2976    1.1971    1.1255    1.0826    1.0683
```

Figure 19 valid data after excluding out-of-range values

Calculate the processed value, and after repeated calculations 5 times, the result is shown in table 5.

n of test	y1(mm)	y2(mm)	y3(mm)	g1(mm)	g2(mm)	g3(mm)	d(mm)
1	0	9	7	1.0683	2.2452	1.7754	69.9935
2	0	9	2	1.0683	2.2452	1.1255	70.0058
3	0	9	5	1.0683	2.2452	1.4273	70.0077
4	0	9	8	1.0683	2.2452	1.9948	69.9871
5	0	9	6	1.0683	2.2452	1.5864	69.9999

Table 5 result

As shown in table 5, the error of the circumference does not exceed ±0.01mm, and the calculated result is acceptable.

3.2.1.2 When the sensor is too close to the cylindrical surface

Table 6 shows the data when the cylinder surface is too far away from the sensor:

n of test	time(s)	y(mm)	g(mm)	n of test	time(s)	y(mm)	g(mm)
1	0	0	2.2452	50	4.9	-1	2.2452
2	0.1	1	2.2452	51	5	-2	2.2452
3	0.2	2	2.2452	52	5.1	-3	2.2452
4	0.3	3	2.2452	53	5.2	-4	2.2452
5	0.4	4	2.2452	54	5.3	-5	2.2452
6	0.5	5	2.2452	55	5.4	-6	2.2452
7	0.6	6	2.2452	56	5.5	-7	2.2452
8	0.7	7	2.2452	57	5.6	-8	2.2452
9	0.8	8	2.2452	58	5.7	-9	2.2452
10	0.9	9	2.2452	59	5.8	-10	2.5273
11	1	10	2.5273	60	5.9	-11	2.8418
12	1.1	11	2.8418	61	6	-12	3.1897
13	1.2	12	3.1897	62	6.1	-13	3.5721
14	1.3	13	3.5721	63	6.2	-14	3.9903
15	1.4	14	3.9903	64	6.3	-15	4.4455
16	1.5	15	4.4455	65	6.4	-16	4.9395
17	1.6	16	4.9395	66	6.5	-17	5.4742
18	1.7	17	5.4742	67	6.6	-18	6.0516
19	1.8	18	6.0516	68	6.7	-19	6.6744
20	1.9	19	6.6744	69	6.8	-20	7.3455
21	2	20	7.3455	70	6.9	-21	8.0683
22	2.1	21	8.0683	71	7	-22	8.847
23	2.2	22	8.847	72	7.1	-23	9.6865
24	2.3	23	9.6865	73	7.2	-24	10.5928
25	2.4	24	10.5928	74	7.3	-23	9.6865
26	2.5	23	9.6865	75	7.4	-22	8.847
27	2.6	22	8.847	76	7.5	-21	8.0683
28	2.7	21	8.0683	77	7.6	-20	7.3455
29	2.8	20	7.3455	78	7.7	-19	6.6744
30	2.9	19	6.6744	79	7.8	-18	6.0516
31	3	18	6.0516	80	7.9	-17	5.4742
32	3.1	17	5.4742	81	8	-16	4.9395
33	3.2	16	4.9395	82	8.1	-15	4.4455
34	3.3	15	4.4455	83	8.2	-14	3.9903
35	3.4	14	3.9903	84	8.3	-13	3.5721
36	3.5	13	3.5721	85	8.4	-12	3.1897
37	3.6	12	3.1897	86	8.5	-11	2.8418
38	3.7	11	2.8418	87	8.6	-10	2.5273
39	3.8	10	2.5273	88	8.7	-9	2.2452
40	3.9	9	2.2452	89	8.8	-8	2.2452
41	4	8	2.2452	90	8.9	-7	2.2452
42	4.1	7	2.2452	91	9	-6	2.2452
43	4.2	6	2.2452	92	9.1	-5	2.2452
44	4.3	5	2.2452	93	9.2	-4	2.2452
45	4.4	4	2.2452	94	9.3	-3	2.2452
46	4.5	3	2.2452	95	9.4	-2	2.2452
47	4.6	2	2.2452	96	9.5	-1	2.2452
48	4.7	1	2.2452	97	9.6	0	2.2452
49	4.8	0	2.2452				

Table 6 sensor data

As shown in table 20, when the position of the sensor on the y-axis is between -9 and 9, the value of g remains at 2.2352mm. It means that in this interval, the sensor is too close to the surface of the cylindrical workpiece.

The wave and characteristics of the output signal at this time are shown in Figure ():

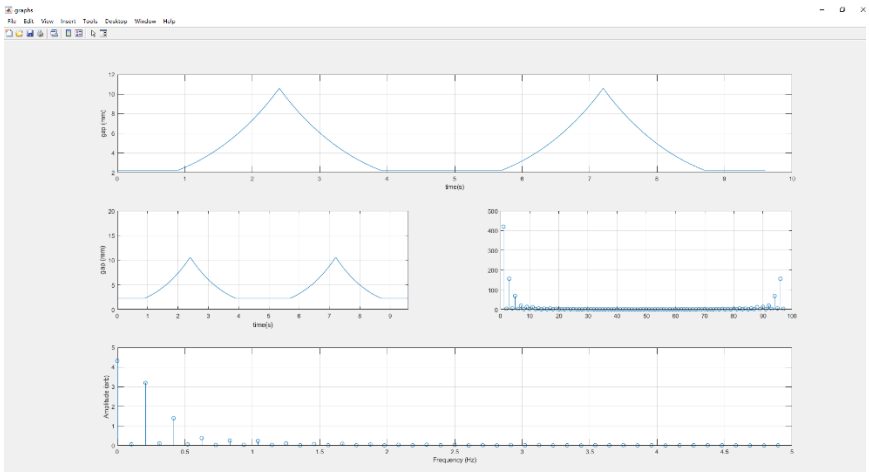

Figure 20 result

The out-of-range values detected by the program are shown in Figure 21 and Figure 22

```
ystart =

     0

yMis =

     0

ans =

  Columns 1 through 13

    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452

  Columns 14 through 26

    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452

  Columns 27 through 39

    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452    2.2452

  Columns 40 through 52

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 53 through 65

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 66 through 78

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 79 through 91

         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 92 through 97

         0         0         0         0         0         0
```

Figure 21 misalignment information

```
ans =

  Columns 1 through 21

     0     1     2     3     4     5     6     7     8     9     9     8     7     6     5     4     3     2     1     0    -1

  Columns 22 through 42

    -2    -3    -4    -5    -6    -7    -8    -9    -9    -8    -7    -6    -5    -4    -3    -2    -1     0     0     0     0

  Columns 43 through 63

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 64 through 84

     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0

  Columns 85 through 97

     0     0     0     0     0     0     0     0     0     0     0     0     0
```

Figure 22 misalignment information

The valid data after excluding out-of-range values is shown in figure 23.

```
ans =

  Columns 1 through 21

    10    11    12    13    14    15    16    17    18    19    20    21    22    23    24    23    22    21    20    19    18

  Columns 22 through 42

    17    16    15    14    13    12    11    10   -10   -11   -12   -13   -14   -15   -16   -17   -18   -19   -20   -21   -22

  Columns 43 through 58

   -23   -24   -23   -22   -21   -20   -19   -18   -17   -16   -15   -14   -13   -12   -11   -10

ans =

  Columns 1 through 13

    2.5273    2.8418    3.1897    3.5721    3.9903    4.4455    4.9395    5.4742    6.0516    6.6744    7.3455    8.0683    8.8470

  Columns 14 through 26

    9.6865   10.5928    9.6865    8.8470    8.0683    7.3455    6.6744    6.0516    5.4742    4.9395    4.4455    3.9903    3.5721

  Columns 27 through 39

    3.1897    2.8418    2.5273    2.5273    2.8418    3.1897    3.5721    3.9903    4.4455    4.9395    5.4742    6.0516    6.6744

  Columns 40 through 52

    7.3455    8.0683    8.8470    9.6865   10.5928    9.6865    8.8470    8.0683    7.3455    6.6744    6.0516    5.4742    4.9395

  Columns 53 through 58

    4.4455    3.9903    3.5721    3.1897    2.8418    2.5273
```

Figure 23 valid data after excluding out-of-range values

Calculate the processed value, and after repeated calculations 5 times, the result is shown in table 7.

n of test	y1(mm)	y2(mm)	y3(mm)	g1(mm)	g2(mm)	g3(mm)	d(mm)
1	10	24	19	2.5273	10.5928	6.6749	69.9987
2	10	24	18	2.5273	10.5928	6.0516	69.9976
3	10	24	21	2.5273	10.5928	8.0683	70.0007
4	10	24	20	2.5273	10.5928	7.3455	70.0013
5	10	24	15	2.5273	10.5928	4.4455	69.9977

Table 7 result

As shown in table 7, the error of the circumference does not exceed ±0.01mm, and the calculated result is acceptable.

3.2.2 The result of data analysis of the sensor's y-axis misalignment

By adjusting the starting point, assuming that the y-axis has an offset of -5mm, the sensed data at this time is shown in table 8.

n of test	time(s)	y(mm)	g(mm)	n of test	time(s)	y(mm)	g(mm)
1	0	0	1.4273	42	4.1	1	1.2976
2	0.1	-1	1.5864	43	4.2	2	1.1971
3	0.2	-2	1.7754	44	4.3	3	1.1255
4	0.3	-3	1.9949	45	4.4	4	1.0826
5	0.4	-4	2.2452	46	4.5	5	1.0683
6	0.5	-5	2.5273	47	4.6	6	1.0826
7	0.6	-6	2.8418	48	4.7	7	1.1255
8	0.7	-7	3.1897	49	4.8	8	1.1971
9	0.8	-8	3.5722	50	4.9	9	1.2976
10	0.9	-9	3.9903	51	5	10	1.4273
11	1	-10	4.4455	52	5.1	11	1.5864
12	1.1	-11	4.9395	53	5.2	12	1.7754
13	1.2	-12	5.4742	54	5.3	13	1.9949
14	1.3	-13	6.0516	55	5.4	14	2.2452
15	1.4	-14	6.6744	56	5.5	15	2.5273
16	1.5	-15	7.3455	57	5.6	16	2.8418
17	1.6	-16	8.0683	58	5.7	17	3.1897
18	1.7	-17	8.847	59	5.8	18	3.5722
19	1.8	-18	9.6865	60	5.9	19	3.9903
20	1.9	-19	10.5928	61	6	20	4.4455
21	2	-20	11.5734	62	6.1	19	3.9903
22	2.1	-19	10.5928	63	6.2	18	3.5722
23	2.2	-18	9.6865	64	6.3	17	3.1897
24	2.3	-17	8.847	65	6.4	16	2.8418
25	2.4	-16	8.0683	66	6.5	15	2.5273
26	2.5	-15	7.3455	67	6.6	14	2.2452
27	2.6	-14	6.6744	68	6.7	13	1.9949
28	2.7	-13	6.0516	69	6.8	12	1.7754
29	2.8	-12	5.4742	70	6.9	11	1.5864
30	2.9	-11	4.9395	71	7	10	1.4273
31	3	-10	4.4455	72	7.1	9	1.2976
32	3.1	-9	3.9903	73	7.2	8	1.1971
33	3.2	-8	3.5722	74	7.3	7	1.1255
34	3.3	-7	3.1897	75	7.4	6	1.0826
35	3.4	-6	2.8418	76	7.5	5	1.0683
36	3.5	-5	2.5273	77	7.6	4	1.0826
37	3.6	-4	2.2452	78	7.7	3	1.1255
38	3.7	-3	1.9949	79	7.8	2	1.1971
39	3.8	-2	1.7754	80	7.9	1	1.2976
40	3.9	-1	1.5864	81	8	0	1.4273
41	4	0	1.4273				

Table 8 sensor data

As shown in table 8, when y is equal to 0 at the starting point, the value of g is 1.4273mm. Compare table (), when y = ±5, g = 1.4273mm. And because when the sensor moves to -20mm on the y-axis, the value of g is greater than the value when y = 20mm. So the starting point should be -5mm misaligned relative to the original y-axis.

The wave and characteristics of the output signal at this time are shown in Figure 24:

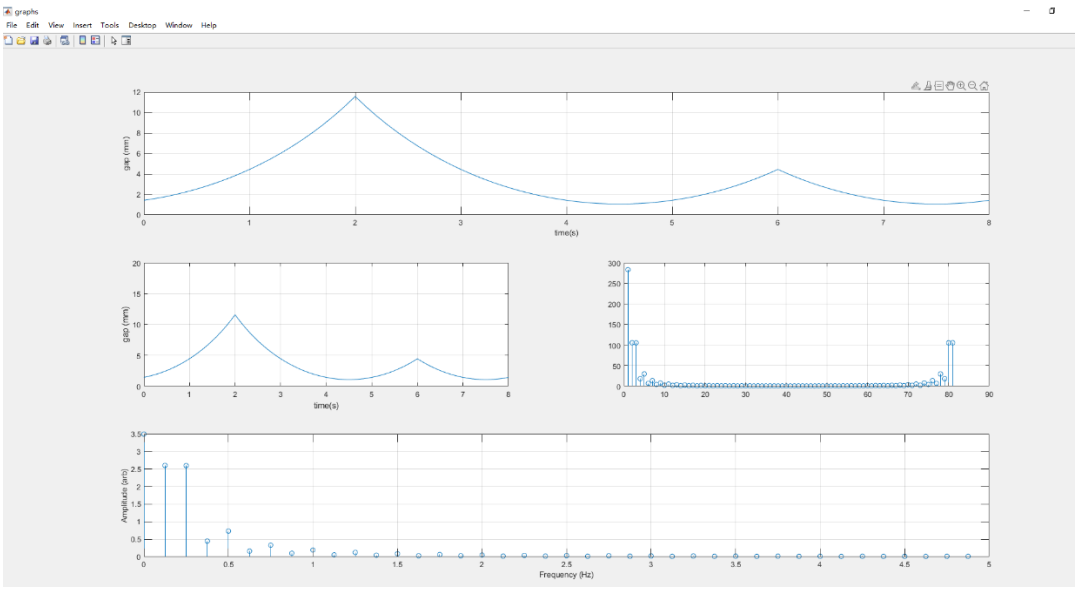

Figure 24 result

After inspection, the misalignment of the y-axis is shown in the figure 25.

```
ystart =

    -5


yMis =

     5


ans =

     0


ans =

     0
```

Figure 25 misalignment information

After the test is completed, the data is repeated 5 times, and the result is shown in table 9.

n of test	y1(mm)	y2(mm)	y3(mm)	g1(mm)	g2(mm)	g3(mm)	d(mm)
1	5	-20	4	1.0683	11.5734	1.0826	70.0020
2	5	-20	-8	1.0683	11.5734	3.5722	70.0012
3	5	-20	-9	1.0683	11.5734	3.9903	70.0007
4	5	-20	-7	1.0683	11.5734	3.1897	69.9992
5	5	-20	1	1.0683	11.5734	1.2976	69.9989

Table 9 result

As shown in table 9, the error of the circumference does not exceed ±0.01mm, and the calculated result is acceptable.

4 Discussion

4.1 Other possible dislocations

In addition to the two dislocations proposed in the experiment, there are other possible dislocation forms: the coordinate axis of the cylindrical workpiece is deflected relative to the original coordinate axis.

If the x and z axes of the cylindrical workpiece are deflected around the y axis, as shown in Figure 26, because the modified report only analyzes the sensor's reciprocating movement on the y axis, this sensing method may not only cause the sensor to be exceeded The problem of the measurement range of the sensor, and may cause the cross-section of the sensor to form an ellipse. Therefore, a program should be added at this time to test the 5 output values. If their errors are extremely large, it can be judged that an ellipse with an Eccentricity smaller than 1 has been formed, which has exceeded the negligible error range. At this time, it can be calculated by the ellipse equation, and the length of the short axis obtained is the diameter of the cylindrical workpiece.

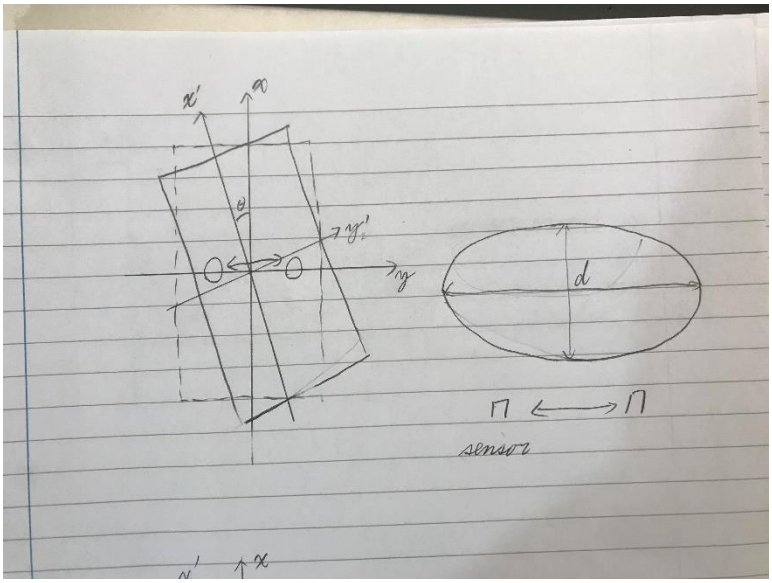

Figure 26 the x and z axes of the cylindrical workpiece are deflected around the y axis

If the z- and x-axes of the cylindrical workpiece are deflected around the y-axis, as shown in Figure 27, this will cause the sensor to sense that the cross-section is not a perfect circle. In this case, the output diameter will have a problem similar to the first part of the Result, which will vary with the selected points. In this case, the function to solve the diameter of the cylindrical workpiece created in the project will give the correct result, so at this time, a program should also be added to test the 5 output values. If their errors are extremely large, you can judge An ellipse with an Eccentricity slightly smaller than 1 has been formed, which has exceeded the negligible error range. At this

time, it can be calculated by the ellipse equation, and the length of the short axis obtained is the diameter of the cylindrical workpiece.

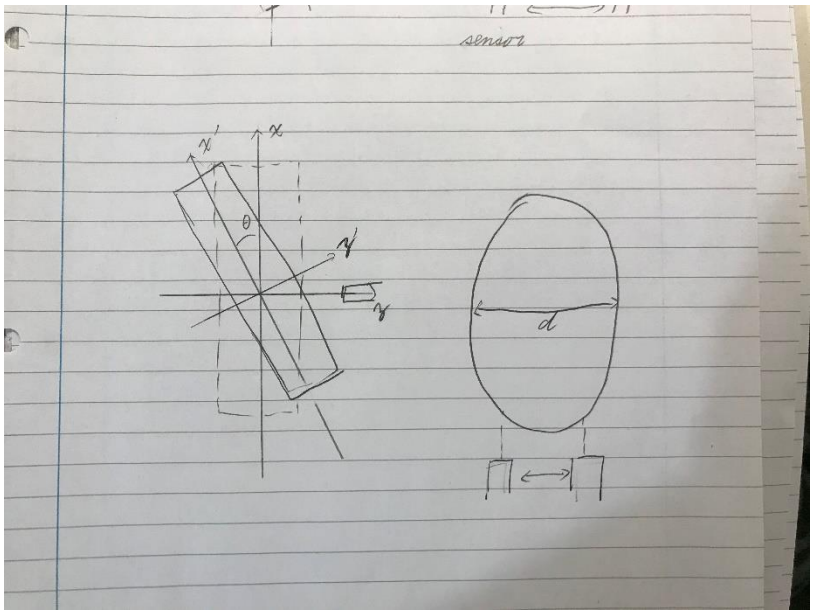

Figure 27 the z- and x-axes of the cylindrical workpiece are deflected around the y-axis

Finally, the author thinks of another possibility that the above two kinds of deflection of the cylindrical workpiece occur at the same time, as shown in Figure 28. In this case, if the ellipse equation is used for calculation, the resulting diameter will be larger than the diameter of the cylindrical workpiece. If the deflection angles of the x and z axes around the y axis are small, the calculation error will be acceptable, but if the deflection angle is too large, it will be seriously larger than the diameter of the cylindrical workpiece. This will have a serious impact on parts processing. In this case, the characteristics of the cylindrical workpiece obtained from the input data are insufficient, which will result in the inability to perform a correct analysis.

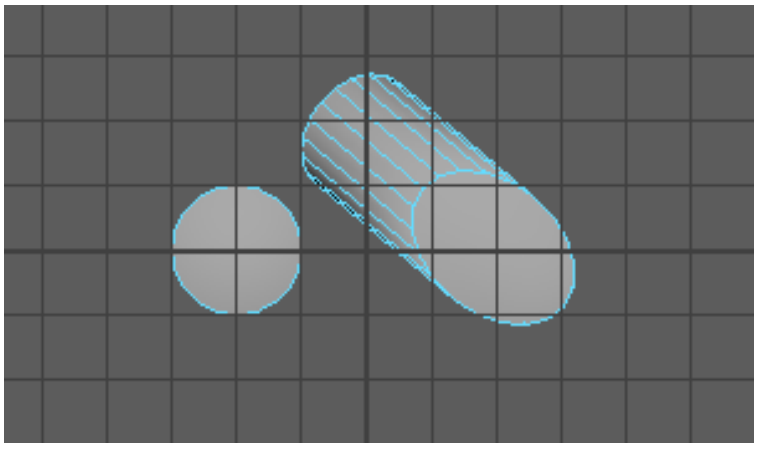

Figure 28 two kinds of deflection of the cylindrical workpiece occur at the same time

4.2 The difference in the analysis of the operation modes of the two sensors

In the introduction, two sensor movement methods are mentioned. One is that the sensor used in the simulation in this project moves on a single y-axis, and the other is that the sensor moves in a circular motion on the plane composed of the y-axis and the x-axis. The signals output by these two sensing methods are different. As shown in Figure 29, the output signal when the sensor performs circular motion is closer to the sine wave.

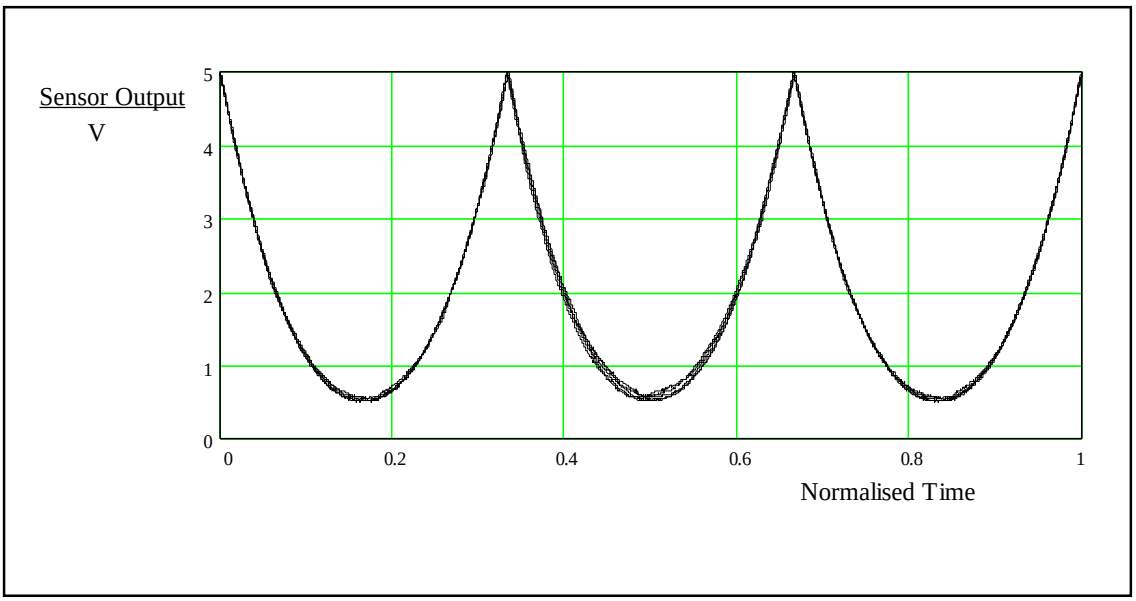

Figure 29.a signal output when sensor moves on a single y-axis (Grosvenor 1995)

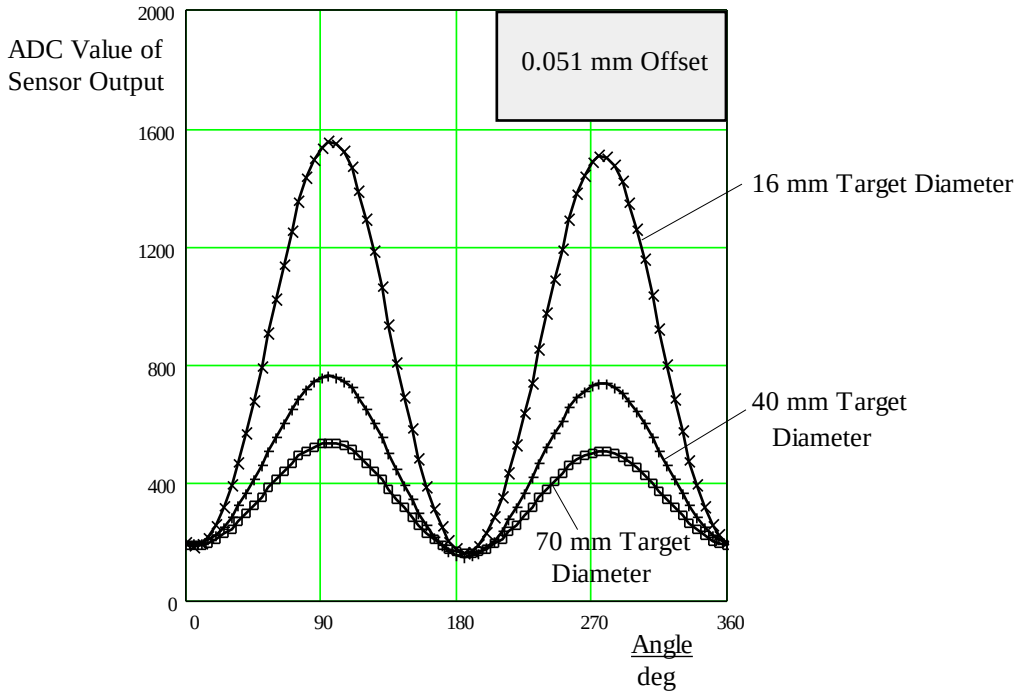

Figure 29.b sensor moves in a circular motion on the plane composed of the y-axis and the x-axis (Grosvenor 1995)

These two sensing methods can use the same method to calculate the diameter of a cylindrical workpiece when the workpiece to be sensed is in an ideal state (no misalignment occurs). In an ideal state, the change of the coordinate on the x-axis of a cylindrical workpiece will not affect the sensing data, so in an ideal state, the data obtained during the circular motion of the sensor can also be directly used to calculate the diameter of the project.

However, there are also differences in obtaining the characteristics of the cylindrical workpiece. The single-axis movement of the sensor can only obtain the data on the y-axis, and cannot make complex judgments. The circular motion of the sensor can obtain the data of the x-axis and the y-axis. Such data helps to judge the misalignment of the coordinate axis of the cylindrical workpiece relative to the original coordinate axis.

4.3 The method of detecting misalignment when the sensor is moving in a circular motion

Because the sensor circular motion sensing method can acquire more target features, the method for detecting misalignment should also be different.

The first discussion is when the dislocation of the workpiece occurs on the coordinate axis (moving on the coordinate axis). The misalignment on the X axis still cannot be detected, because when the cylindrical workpiece moves on the x axis, the distance between the surface of the cylinder and the surface of the sensor will not change. The problem caused by the misalignment of the y-axis and the z-axis is similar to the

sensing method of the sensor in the y-axis movement, so the detection and processing methods can also be carried out by the same method.

However, when the coordinate axis of the cylindrical workpiece is deflected relative to the coordinate axis of the measurement system, the sensor's circular motion sensing method can provide more target features than the sensor's y-axis motion sensing method, so it can be more accurate Detect the misalignment information of the measurement target.

When the x- and z-axes of the cylindrical workpiece are deflected around the y-axis, the equation of the x-axis deflection of the cylindrical workpiece can be determined by calculating the line where the point on the surface of the cylindrical body is closest to the sensor. When the z-axis and x-axis of the cylindrical workpiece are deflected around the y-axis, the equation after the x-axis deflection is determined by comparing the increase in the distance between each row of data.

When the above-mentioned misalignment occurs, the points where the slope of the tangent line is zero can be determined by differentiation, and then whether these points are the maximum or minimum points can be determined by double differentiation. These data are used to determine the x-axis and y-axis of the cylindrical workpiece, and then combine the above-mentioned detection methods for detection.

5 Conclusions

The main purpose of this project is to develop a software that can assist Professor Roger Grosvenor to rebuild a new test bed.

Although the project can currently detect some of the misalignments and repair them, further work is needed in order to be able to complete the analysis of the completely real situation. As the author pointed out in the discussion, the detection method of sensor movement on the y-axis may not be able to complete the inspection of all misalignments, and further analysis of the circular movement may be required. This requires the addition of new detection functions.

In the completed work, there is sufficient evidence to show that the program can detect, report and repair the two types of misalignments contained in the design, which can provide certain analysis assistance for rebuilding the test bench.

The work carried out shows that the detection method of sensor movement on the y-axis can be used in most cases, but when the coordinate axis of the detected target is severely deflected, this detection method will be difficult to provide sufficient target characteristics for analysis. At this time, the circular motion detection method of the sensor will be more suitable.

6 References

Grosvenor, R.I. 1997. An Assessment of a Novel Method for the In-Process Measurement of Machined Component Dimensions. Paper presented at Sensors and their Applications VIII, pp 123-128

Grosvenor, R.I. 1994. Analysis of Rotating Sensor Method for Cylindrical Component Measurement. Journal of Systems Engineering, pp 23-31

Wojciech, K., Lukianowicz, C. 2013. Non-Contact Optical Metrology for Automated In-Process Inspection of Machined Surfaces [Online] Available at:

https://www.researchgate.net/publication/259705334

Shipsides, G. 2015. *In Process Measurement of Component Diameters* BEng Dissertation: Cardiff University

Bentley, J.P. 1995. Principles of Measurement Systems. 3rd ed. Essex: Longman Group Limited.

7 Appendix

7.1 Appendix A: Nomenclature

Symbol	Explanation	unit
g	The distance from the sensor surface to the target surface on the z-axis	mm
y	The distance the sensor moves on the y axis	mm
a	The x-axis coordinate of the center of the cross section of the cylindrical workpiece	mm
b	The y-axis coordinate of the center of the cross section of the cylindrical workpiece	mm
R	The radius of the cross section of the cylindrical workpiece	mm
d	Diameter of the cross section of the cylindrical workpiece	mm

7.2 Appendix B: code of Matlab Program:

B.1 Function Main()

```
function Main_program
% This program is for 3D Simulation of Inductive sensor sensing a cylindrical surface
% It includes four main tasks:
%     1, UI and open the file.
%     2, Data analysis
%     3, Calculation correction when misalignment occurs
%     4, 3D simulation of surface

%-------------------------------task 1----------------------------------
%**************************************************************************
%**************************************************************************
% window
f1 = figure('NumberTitle','off','Position',[0 0 620 240]);
set(f1,'visible','off');
% title
hTitle = uicontrol('Style','text','String',...
    '3D Simulation Of Inductive Sensor Sensing A Cylindrical Surface',...
    'Position',[20 190 580 30],'FontSize',15,...
    'HorizontalAlignment','center');
% Editable text field,The place to imput the file name
hFilePath = uicontrol('Style','edit','String','Input File Name',...
    'Position',[50 145 400 25],'Callback',{@FilePath_Callback});
%list of type of sensing method
hTypeOfMethod = uicontrol('Style','popupmenu',...
    'String',{"Circular Motion","Linear Motion"},...
    'Position',[50 100 400 25],'Callback',{@TypeOfMethod_Callback});
%Button for Browesing the file
hFileBrowes = uicontrol('Style','pushbutton','String','Browes',...
    'Position',[500 145 70 25],'Callback',{@FileBrowes_Callback});
%text field to show radius of cylinder
hDiameterOfCylinder = uicontrol('Style','edit','string','diameter',...
    'Position',[100 50 80 25]);
%Button for loading the program
hLoad = uicontrol('Style','pushbutton','String','Load',...
    'Position',[500 100 70 25],'Callback',{@Load_Callback});
%text to show the unit of radius
hUnit = uicontrol('Style','text','String','mm',...
    'Position',[185 50 40 25],'FontSize',12);
%**************************************************************************
%align
align([hFilePath,hFileBrowes],'None','Middle');
align([hTypeOfMethod,hLoad],'None','Middle');
align([hDiameterOfCylinder,hUnit],'None','Middle');
```

```
%Change units to normalized
f1.Units = 'normalized';
hTitle.Units = 'normalized';
hFilePath.Units = 'normalized';
hTypeOfMethod.Units = 'normalized';
hFileBrowes.Units = 'normalized';
hLoad.Units = 'normalized';
hDiameterOfCylinder.Units = 'normalized';
hUnit.Units = 'normalized';

%Assign a name to appear in the window title.
set(f1,'Name','load the file');
% Move the window to the center of the screen.
movegui(f1,'center');
% Make the UI visible
set(f1,'visible','on');
%************************************************************************
%Inithialize variables
theFileName = string();
MethodType = string();
%************************************************************************
% UI callback functions
function FilePath_Callback(source,eventdata)
   theFileName = get(source,'String');
end

function TypeOfMethod_Callback(source,eventdata)
   MethodType = hTypeOfMethod.String(hTypeOfMethod.Value);
end

function FileBrowes_Callback(source,eventdata)
   [theFileName hFilePath.String] = uigetfile({'*.txt'},'File Selector');
end

function Load_Callback(source,eventdata)
   MethodType = hTypeOfMethod.String(hTypeOfMethod.Value);
   %calculation start here:
   sensingdata = importdata(theFileName,',');
   time = sensingdata(:,1);
   y = sensingdata(:,2);
   g = sensingdata(:,3);%import data in the file

   [m n] = find(y == 0);
   timepercycle = time(1:m(3,1),1);
   ypercycle = y(m(1,1):m(3,1),1);
   gpercycle = g(m(1,1):m(3,1),1);
```

```
lt = timepercycle(size(timepercycle),1);

Ng = size(g);
Tt = time(size(time),1);
ws = Ng(1)/Tt(1);
nombfft = ws*Tt(1);

gfft = fft(g);
F = abs(gfft);

deltaf = 1/Tt;
frequencyrange = (nombfft/2)*deltaf;
F2 = 2*F/nombfft ;
F2(1,1) = F2(1,1)/2;
freqencyscale = [0:deltaf:(frequencyrange-deltaf)]';
sizeFS = size(freqencyscale);
sizeFS = sizeFS(1);

figure('Name','graphs','NumberTitle','off');
subplot(3,2,[1,2]);
plot(time,g);
xlabel('time(s)');
ylabel('gap (mm)');
grid

subplot(3,2,3);
plot(timepercycle,gpercycle);
xlabel('time(s)');
ylabel('gap (mm)');
axis([0 lt(1) 0 20])
grid

subplot(3,2,4);
stem(F);
grid

subplot(3,2,[5,6]);
stem(freqencyscale,F2([1:sizeFS]));
xlabel('Frequency (Hz)');
ylabel('Amplitude (arb)')
grid
% find R

[ystart,yMis] = yMisalignment(g,y)

[gOutOfRange,yOutOfRange,YorN] = CheckOutofRange(ypercycle,gpercycle);
```

```
    gOutOfRange'
    yOutOfRange'
    if YorN == 1
        [Newy,Newg] =
DeleteOutRange(ypercycle,gpercycle,yOutOfRange,gOutOfRange);
        Newy'
        Newg'
        ypercycle = Newy;
        gpercycle = Newg;
    end

    [d,y1,y2,y3,g1,g2,g3] = findoutR(ypercycle,gpercycle)


    hDiameterOfCylinder.String = d;
    %3D simulation
    figure('Name','3D simulation','NumberTitle','off');
    [X,Y,Z] = cylinder(d/2);
    surf(X,Y,Z);
end

end
```

B.2 function findoutR()

```
function [d,x1,x2,x3,y1,y2,y3] = findoutR(ypercycle,gpercycle)
    gmax = max(gpercycle);
    gmin = min(gpercycle);
    MAX = find(gpercycle == gmax,1)';
    MIN = find(gpercycle == gmin)';
    sizeMIN = size(MIN);
    sizeMINi = sizeMIN(1,2);
    n = MAX(1,1);
    j = 1;
    while n == MIN(1,j)||n == MAX(1,1)
        j = 0;
        for i = 1:sizeMINi
            if MAX(1,1)>MIN(1,i)
                j = j+1;
            else
                j = j;
            end
        end
        if j == 0
            n = randsample([MAX(1,1):MIN(1,1)],1);
            j = 1;
        else
            n = randsample([MIN(1,j):MAX(1,1)],1);
        end
    end
    x1 = ypercycle(MIN(1,j));
    y1 = gpercycle(MIN(1,j));
    x2 = ypercycle(MAX(1,1));
    y2 = gpercycle(MAX(1,1));
    x3 = ypercycle(n);
    y3 = gpercycle(n);

    syms a b R
    eqn1 = (x1-a)^2+(y1-b)^2==R;
    eqn2 = (x2-a)^2+(y2-b)^2==R;
    eqn3 = (x3-a)^2+(y3-b)^2==R;
    eqns = [eqn1 eqn2 eqn3];

    S = solve(eqns,[a b R]);
    a = S.a;
    b = S.b;
    R = double(S.R);
    r = sqrt(R);
    d = r*2;
end
```

B.3 function CheckOutofRange()

```
function [gOutOfRange,yOutOfRange,YorN] = CheckOutofRange(ypercycle,gpercycle)

n = size(gpercycle);
n = n(1,1);
m = -1;
l = -1;
g = 0;
j = 1;
gOutOfRange = zeros(n,1);
yOutOfRange = zeros(n,1);
YorN = 0;

for i = 1:n
    if i == 1
        l = gpercycle(i+1,1);
        g = gpercycle(i,1);
        if l == g
            gOutOfRange(j,1) = gpercycle(i,1);
            yOutOfRange(j,1) = ypercycle(i,1);
            j = j + 1;
        end
    elseif i == n
        m = gpercycle(i-1,1);
        g = gpercycle(i,1);
        if m == g
            gOutOfRange(j,1) = gpercycle(i,1);
            yOutOfRange(j,1) = ypercycle(i,1);
            j = j + 1;
        end
    else
        l = gpercycle(i+1,1);
        m = gpercycle(i-1,1);
        g = gpercycle(i,1);
        if g == m
            gOutOfRange(j,1) = gpercycle(i,1);
            yOutOfRange(j,1) = ypercycle(i,1);
            j = j + 1;
        elseif g == l
            gOutOfRange(j,1) = gpercycle(i,1);
            yOutOfRange(j,1) = ypercycle(i,1);
            j = j + 1;
        end
    end
end
```

```
x = size(gOutOfRange);
x = x(1,1);
z = 0;
for i = 1:x
    z = z + gOutOfRange(i,1);
end

if z ~= 0
    YorN = 1;
end

if z == 0
    gOutOfRange = 0;
    yOutOfRange = 0;
end

end
```

B.4 function DeleteOutRange()

```
function [Newy,Newg] = DeleteOutRange(y,g,yOutOfRange,gOutOfRange)
m = size(g);
m = m(1,1);
n = size(gOutOfRange);
n = n(1,1);
l = m - n;
Newy = zeros(l,1);
Newg = zeros(l,1);
j = 1;
k = 1;
for i= 1:m
   if g(i,1) == gOutOfRange(j,1)
      j = j + 1;
   else
      Newy(k,1) = y(i,1);
      Newg(k,1) = g(i,1);
      k = k + 1;
   end
end

end
```

B.5 function yMisalignment()

```
function [ystart,yMis] = yMisalignment(g,y)
gmin = min(g);
[i j] = find(g == gmin);
ymin = y(i(1,1),1);
ystart = 0 - ymin;
yMis = abs(0 + ymin);
end
```