Design of Sag Measurement System of High Voltage Line Based on RTK Positioning Technology

ABSTRACT

High voltage transmission line is an important part of power transmission system, and sag is one of the important factors affecting its safe operation. Overhang will cause the line to contact with the ground or other objects, resulting in power transmission failure, resulting in safety accidents; If the sag is too small, the line will bear too much mechanical stress and shorten its service life. The current sag measurement methods often rely on manual observation and indirect measurement, which have the problems of low efficiency, insufficient precision and high operational risk. With the progress of science and technology, more and more researchers begin to explore automated and intelligent measurement means to improve work efficiency and measurement accuracy.

In this paper, a high voltage line sag measurement system based on RTK is designed with line mobile robot as the carrier. Based on catenary model and parabola model, the maximum point and maximum value of sag are analyzed, and the error correction of the maximum value of sag is realized by considering the parabola deformation under concentrated load. The sag measurement of HV transmission lines can be realized by measuring the location information of two adjacent transmission towers and line mobile robots by RTK.

In order to test the measurement accuracy of Micius 4G RTK carried by the line mobile robot, the measurement data of Zhonghaida TS5 Pro intelligent inertial navigation RTK is taken as the calibration value and compared with the measured data of Micius 4G RTK. The latitude and longitude error of Micius 4G RTK is about 0.0001%, and the elevation error is about 0.2%. By Vincenty formula, the longitude and latitude measured by RTK are converted to horizontal distance, and the distance measured by TS5 Pro intelligent inertial navigation RTK is compared with the error of about 0.014%.

Finally, in order to further improve the user experience and operation convenience, this paper also uses Python programming language to design a set of user interaction interface based on PyQt5. This interface can not only import the transmission line parameters, but also display the data measured by RTK, such as latitude and longitude, elevation and so on. Moreover, the sag algorithm embedded in the interface can calculate the horizontal position of the maximum sag point and the maximum sag value and display them in real time. In addition, the animation function is also added to the interface, which can simulate the real-time position of the line mobile robot, and display the sag value under different positions to help users better observe.

Key words:high voltage transmission line ; sag algorithm; RTK;accuracy test ;user interface

CATALOG

1 Introduction

1.1 Background and significance

With the advancement of information and communication technology, the traditional power system network is transforming into a smart grid system. High-voltage transmission lines are an important part of the power system, which are responsible for transmitting the power generated by power plants to various power consuming areas.

Transmission lines in a certain tension, high-voltage transmission lines and line suspension points connected to the vertical distance between the straight line, said the sag. High voltage transmission line sag will be affected by a variety of factors, such as temperature, wind, snow and ice and other natural conditions, as well as wire materials, tension, span and other engineering factors. The size of the sag is directly related to the safety distance of high-voltage transmission lines, crossing capacity and transmission efficiency. Therefore, it is necessary to control the sag within the scope of the design regulations. In order to ensure the safe operation of the power system network, the high voltage transmission line needs to set a safe arc sag value. Thereby, the tension of the line between the high-voltage transmission towers can be reasonably controlled in order to avoid faults caused by the overhead conductor being too close to the ground or other objects. It can also prevent excessive tension from causing damage to the transmission towers, while ensuring the current-carrying capacity of the conductor.

At present, the arc sag measurement methods of high-voltage transmission lines mainly include direct measurement method, indirect measurement method and remote monitoring method. The direct measurement method needs to be carried out in the field, the operation is more complicated, and easily affected by the environment. Indirect measurement method by measuring other related parameters, so as to indirectly deduce the arc sag value. Remote monitoring method is based on advanced sensor technology and communication means to realize real-time monitoring of transmission line status. With the progress of science and technology, UAV aerial survey, satellite remote sensing and other emerging technologies are also beginning to be applied to the arc sag measurement of high-voltage transmission lines. Accurate and timely monitoring of the arc sag of transmission lines can not only discover potential safety hazards of the line in time, but also help to develop a reasonable line design and tower layout program.

1.2 Research status of sag measurement of high voltage transmission lines

1.2.1 Traditional high voltage transmission line sag measurement method

Traditional sag measurement methods mainly include: sag plate observation method, angle method and head-up method， Among them, the sag plate observation method is divided into equal length method (parallelogram method) and different length method[1]。

In the actual transmission line measurement, in order to avoid the influence of the stall distance or the height difference of the suspension point, and to reduce the amount of field calculation, the first choice should be the arc prolapse plate observation method. Equal-length method of observing arc prolapse must be applied in the case that the height difference of hanging point of overhead line is zero or very small, so as to ensure the accuracy of observing arc prolapse. Heterogeneous length method is mainly used in the case that the parallelogram method cannot be used due to the constraints of terrain, tower height, etc., which can observe the line arc sag more conveniently.

When the objective conditions are limited and it is impossible to use the method of different lengths or equal lengths, the angular method can be used for observation. According to the terrain conditions and the size of the arc dip, can be placed in different positions of the instrument such as latitude and longitude for arc dip observation. According to the different position of the instrument, it is divided into the observation method of the end of the file, the observation method of the outside of the file and the observation method of the inside of the file.

If the above methods are not applicable, then consider using the plain view method for arc dip observation. In the case of large height difference, large span or other special terrain, the arc prolapse value is usually greater than the tower height, which can be observed by the level view method.

1.2.2 High-voltage transmission line sag measurement method based on physical parameters

In order to improve the efficiency of sag measurement and reduce the influence of terrain, climate and other factors on the measurement, a variety of real-time sag monitoring devices have been designed and developed by domestic and foreign power departments and research institutions.

The CAT-1 manufactured by The Valley Group Inc. in the United States measures the line stress in real time by means of a stress sensor, transmits the data to the master station, and realizes the function of calculating the arc sag and temperature of the transmission line by means of the system software[2].

The Power-Donut 2 produced by USI and the MT series produced by Hangzhou Haikang Thunderbird calculate the arc sag in real time by measuring the temperature of the transmission line[2]. The working principle is to use the numerical correspondence between temperature and inclination or inclination and force, through the high-precision temperature sensor in the device, real-time measurement of the temperature of the transmission line, to determine the dynamic changes in the stress on the line, so as to determine the arc sag value.

In 2011, E. Cloet et al[3] proposed a method for determining arc sag and transmission line capacity by detecting vibrations in transmission lines. The method utilizes a vibration sensor to detect vibrations to determine the arc sag of the line and the sensor can be placed directly at any point on the transmission line to measure the vibration frequency. The measured data is processed using a digital signal processor and sent to a remote server via a GPRS/GSM system.

In 2011, Zhimin, H. et al[4] proposed a tension-based DLR system to measure line arc sag.The DLR system consists of two parts: a data acquisition terminal and a monitoring and management platform. The acquisition terminal installed at the end of the transmission line measures the axial tension with a tension sensor. The collected data are then transmitted using GPRS/GSM to the monitoring and management platform, which collects load data from the supervisory control and data acquisition (SCADA) system and combines it with the data received from the data acquisition terminal to calculate the arc sag of the transmission line. Experiments have shown that the error in arc droop measurement is less than 2% using this system.

In 2012, Nazare, F.V.B.d. et al[5] proposed a hybrid photovoltaic system that can measure temperature and current on overhead transmission lines and can be used to measure arc sag. The system uses photonic technology for data transmission and power transfer and it consists of two subsystems: one on the ground and the other on the transmission line. The subsystem on the transmission line consists of a current sensor, a photovoltaic converter, a temperature sensor, a low-power microcontroller with serial outputs, and an 850 nm LED fiber optic transmitter. The ground subsystem utilizes a power laser diode to transmit 830 nm wavelength light to the high voltage subsystem. The high-voltage subsystem then converts the light signal into an electrical signal with the help of a photovoltaic cell and powers all the electronic components. The subsystem measures the temperature and current and sends the digitized signals to the receiving subsystem on the ground, which in turn measures the arc droop of the transmission line from the temperature.

In 2019, Hsu, T.S. et al[6] proposed an IoT based arc droop measurement system. The system uses a sensor module which is installed on a high voltage transmission line to measure different parameters such as triaxial acceleration, ambient temperature, humidity, etc. The triaxial acceleration is then utilized and the arc sag can be determined based on the height difference elevation angle of the transmission tower.

1.2.3 High-voltage transmission line sag measurement method based on signal and image processing technology

In 2011, Tong Weiguo et al[7]Based on the aerial sequence images, the coordinates of the center of the isolation bar are measured by image processing, and the sag of the transmission line is obtained by stereo vision technology and least square linear regression algorithm. When extracting feature points on transmission conductors, this method mostly uses image matching algorithms to identify spacers, but the matching accuracy is not high, which cant solve the interference problem of complex background in images, and needs manual intervention, so the efficiency is not high.

In 2012, Mahajan, S.M. et al[8]A sag measurement method based on GPS signal is proposed. Place a GPS mobile station on the transmission line and set a base station on the ground. After receiving the GPS signal, the original GPS data is sent to the sag monitoring center using the established radio frequency or cellular link. Then the specific GPS software is used to process the GPS data and input it into the system. The height or sag information is then extracted from a set of GPS data using a data processing module.

In 2015, Mo Zhiyue et al[9] proposed to capture a part of the image of the cable with a high definition camera and acquire the data. Median filtering as well as Canny edge detection techniques were applied in the image preprocessing, and then the key parameters of the curve were extracted by using the hanging chain line model. Thus, the complete hanging chain curve is reconstructed and the arc droop is calculated from it.

In 2019, Ma Weifeng et al[10] used UAV LiDAR technology to construct a 3D model of transmission lines by analyzing static point cloud data, and combined with the equation of state to obtain the key parameters, and then used the arc-drop model equation to complete the 3D morphology simulation of the arc-drop of the transmission lines under different climatic conditions. However, due to the high cost of LiDAR, and the large amount of point cloud data required and the difficulty of obtaining it, the cost of this technology is too high.

In 2019, Xu, Q. et al[11]A method of monitoring power lines by using Mr sensors on the ground is proposed. The method uses 5 sensors to estimate the sag value of a three-phase transmission line. In order to determine the sag value of the line, an optimization method based on the AIS algorithm is used, which can also measure the tilt angle of the transmission tower, and the error of calculating the sag using this method is less than 0.3%.

In 2022, Bin Hao et al[12]In view of the complex conditions of transmission lines, such as complex ground conditions, large gear spaces, and high tower heights, DJI Innovation Technology Co., Ltd. was selected（DJI）Launched the Phantom 4 RTK drone platform,By carrying out image photography on feature points, using Python language program design, editing and specifying algorithm to realize automatic capture of photo elevation data, obtaining horizontal elevation information, and then designing related algorithms to realize automatic calculation,In actual measurement scenarios, the measurement and calculation efficiency are improved.

In 2023, Qian et al[13] improved the method of Tong Weiguo et al. They utilized the interval bar automatic segmentation algorithm CBAM-Mask-RCNN, which incorporates the attention mechanism of convolutional block attention module (CBAM), based on the aerial video of a UAV, combined with beam This method is combined with the beam method leveling, spatial front meeting and spatial curve fitting algorithms to measure the arc sag of the transmission line. However, this method still requires manual intervention when using beam leveling to calculate the external orientation parameters and the forward rendezvous algorithm to recover the ground 3D coordinates of the center of the spacer bar.

1.2.4 High-voltage transmission line sag measurement method based on line mobile robot

With the continuous development of online monitoring technology in transmission line condition monitoring, and due to the advantages of line mobile robot that can realize all-weather, all-weather, and all-terrain measurements, more and more scholars are combining line monitoring with line mobile robot, supplemented with wireless communication technology and satellite positioning technology, to realize real-time monitoring and arc droop observation of dynamic changes in transmission lines.

In 2012, Abraham, A.P. et al[14] used a semi-automatic Line Reconnaissance Robot (LRR) to measure different parameters such as arc droop, current, and inclination on real-time high voltage transmission lines. The robot was charged by the electric field of the high voltage transmission line and two AVR microcontrollers were used for better control and coordination of sensors and actuators. The robot uses a three-axis accelerometer and gyroscope to accurately measure its acceleration and tilt angle on the transmission line. The arc sag is measured by double integration along the z-axis, while the tilt data from the gyroscope compensates for errors.

In 2020, AYDIN TARIK ZENGIN et al[15] proposed the use of a line moving robot to measure the arc sag by remotely controlling the ROSETLineBot line moving robot through wireless communication technology, sending all the data collected by the sensors to the central computer in real time, and then processing the collected acceleration data to obtain the robot's velocity and position, and A method for filtering noise and de-trending sensor offsets is proposed. It can not only measure the line arc sag, but also detect other possible faults such as obstacles and icing.

In 2022, Kai Li et al[16]Digital communication technology is used to establish a local wireless communication network. By outputting engineering parameters and positioning base station coordinates and elevation, the inclination sensor is used to correct the attitude of the observation device, the existing position and coordinates of the device are collected and displayed by Beidou positioning, and a complete set of intelligent sag observation devices are designed by laser radar to level the sag of other sub-conductors.

In 2022, Qiyun Han et al[17] developed a guide line on-line measurement system, which utilizes satellite positioning technology and radar measurement and control technology to transmit the measurement data to the monitoring terminal wirelessly, based on the spatial positioning to realize the arc droop observation of the guide line. Moreover, the device integrates the inclination sensor, satellite positioning system and LIDAR, which improves the accuracy of data and positioning in arc dip observation.

In 2023, Cheng Yansheng et al[18] installed two information collection devices at different locations of the transmission line to collect height data at the low hanging point and the lowest point of the conductor. The device performs high-precision positioning by receiving positioning signals and differential data from the BeiDou system and collects the data synchronously, then transmits the data to the monitoring platform via 4G/5G network, and calculates the maximum arc sag value of the transmission line based on the parabolic model. In addition, the differential base station can correct the data, which further enhances the accuracy of localization.

In the same year, Wang, Remember et al[19] designed a hybrid positioning system that fuses satellite and inertial data in response to the problem of transient signal loss or poor signal that may be encountered by satellite positioning systems. The system is equipped with three motion modes, which can select the optimal data source for data fusion under different signal conditions, thus improving the positioning accuracy.

1.2.5 Analysis of research status

After the above overview of the current status of research on arc sag measurement methods for high-voltage transmission lines, it is not difficult to find that the measurement methods have undergone a transformation from traditional observation to modern intelligent technology, mainly including:

①Traditional methods：Such as sag plate observation method, angle method and head-up method, these methods are simple and easy to implement, but limited by human error and environmental factors.

②Physical parameter-based approach: The use of sensors to monitor physical parameters such as wire stress and temperature significantly improves measurement accuracy and efficiency.

③Method Based on Signal and Image Processing Technology： Using technologies such as aerial photography, GPS, high-definition cameras and lidar, highly automated and high-precision measurements are achieved.

④Method based on line mobile robot：The line mobile robot combines satellite positioning, wireless network and radar measurement and control technology to realize all-weather and all-terrain automatic monitoring, which improves the real-time and accuracy of data acquisition.

1.3 Main research contents

(1) Sag measurement scheme of high-voltage transmission line

Build a high voltage transmission line arc sag measurement system with RTK measurement technology, so that it can receive the parameters and measurement data of the transmission line, and calculate the arc sag value of the high voltage transmission line, which is then displayed by the upper computer. By comparing various arc sag measurement methods, analyzing the advantages and disadvantages and rationality of each scheme, and finally determining the scheme with higher precision, higher efficiency and more accurate results.

(2) High voltage transmission line sag algorithm

The algorithm system is developed for the arc droop measurement and error correction of high-voltage transmission lines, and the maximum point of arc droop and the maximum value of arc droop of high-voltage transmission lines are analyzed by combining the hanging chain line model and the parabolic model, and the arc droop error compensation is realized by the model under the centralized load, and the algorithm is optimized.

(3) Accuracy test

The data required by the algorithm is collected by RTK， And test the measurement accuracy and algorithm accuracy of RTK,Ensure high accuracy as much as possible, thereby reducing errors.

(4) Program design

Write a program to display and calculate the collected data, find out the arc sag value of the high-voltage transmission line and display it in real time. Design the user interactive interface, users can import the parameters of the transmission line from the local file, and then input the latitude, longitude and elevation data of the two transmission towers, calculate the arc droop value through the embedded algorithm, and display the above parameters and the calculation results in real time on the interface. In addition, an animation is designed to simulate the real-time position of the line mobile robot and display the arc droop values at different positions.

2 Overall Program Design for Arc Sag Measurement of High Voltage Transmission Lines

2.1 Selection of sag measurement method of high voltage transmission line

2.1.1 Overview of sag measurement methods of high-voltage transmission lines

Through the analysis of the current research status of arc sag measurement of high-voltage transmission lines in the previous chapter, the arc sag measurement methods can be simply classified into direct measurement method and indirect measurement method. In recent years, with the continuous development of remote monitoring technology, a variety of emerging methods have been used for arc sag measurement, which are mainly realized by UAV aerial survey and line mobile robot. In this paper, the line mobile robot will be selected to measure the arc sag of high-voltage lines.

（1）Direct measurement method

Direct measurement is a more traditional measurement method, mainly through the observation instrument to directly measure the distance between the transmission line and the ground or reference point. The direct measurement method is simple and low-cost, but its accuracy is greatly affected by the operator's skills and the measurement environment, and there is a certain safety risk in the high-voltage environment. Therefore, this method is gradually being replaced by safer and more accurate methods, which are now mainly used in small-scale or temporary measurement projects.

（2）Indirect measurement method

The indirect measurement method utilizes sensors (such as inclination sensors, temperature sensors and stress sensors, etc.) installed on the pole tower or high-voltage transmission line to indirectly obtain arc sag data. This method can reduce manual work, improve the safety and accuracy of measurement, and is suitable for fixed long-term monitoring, but the accuracy of indirect measurement cannot be guaranteed and is easily affected by environmental factors.

（3）Remote monitoring method

Remote monitoring method is an emerging technology in recent years, mainly covering wireless communication network, satellite positioning technology, image processing and other modern technological means, which not only improves the frequency of real-time data updating, but also greatly improves the work efficiency and safety. The current measurement method is mainly realized by drone aerial survey and line mobile robot.

UAV aerial survey by carrying high-resolution camera, LIDAR and other equipment, accurate and efficient to complete the detection and measurement of high-voltage lines, but the drone by the weather factors have a greater impact, and the drone aerial survey of several methods are through the image or three-dimensional point cloud data processing, so as to get the location of the transmission tower information, belongs to the indirect access, is not a direct measurement to get, so in terms of accuracy is still somewhat lacking. Therefore, this thesis selects the line mobile robot to measure the arc sag of high-voltage lines.

Now, a variety of line mobile robots have been developed at home and abroad, as shown in Figure 2.1, which are mainly used in the real-time monitoring of high-voltage transmission lines and arc sag measurement. Common line mobile robot measurement of high-voltage line arc sag method has laser ranging method, visual recognition method, sensor measurement method, ultrasonic measurement method, RTK measurement method and so on. Each of these methods has its own advantages, but also has its limitations. In this section, various methods based on line mobile robot will be compared and the final solution will be selected.

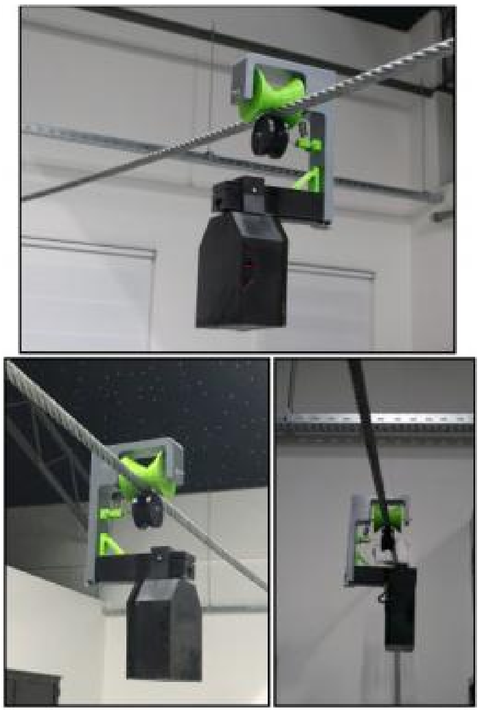

Figure 2.1 Line Mobile Robot[15]

2.1.2 Laser ranging method

The laser ranging method is measured by a laser rangefinder installed on the inspection robot. The laser rangefinder emits a laser beam, and the laser beam is reflected back after hitting the ground. By calculating the laser round trip time, the lowest point of the high-voltage line can be obtained. The distance to the ground.

The laser ranging method has high accuracy and intuitive results. The laser rangefinder can be used to quickly obtain distance data to the centimeter level. Moreover, the equipment is easy to integrate and automate, so it is very suitable for installation on robots for large-scale high-voltage line inspection and measurement.

However, laser ranging method is greatly affected by environmental factors. In an environment with heavy haze, particles in the air will scatter the laser beam, causing the laser reflection signal to weaken, thus affecting the accuracy of ranging results. Under severe weather conditions such as rain and snow, water mist and wind and snow will also interfere with laser propagation, causing serious measurement errors.

2.1.3 Visual recognition method

The vision measurement method uses a line mobile robot equipped with a high-resolution camera to capture real-time images of transmission lines. The image is transmitted to the control center, and the outline of the transmission line is determined by edge detection technology, then the actual size and position of the transmission line are estimated by geometric transformation, and then the relative height between the transmission line and the ground is calculated, and the sag value is obtained by combining the physical characteristics of the transmission line.

Vision measurement is relatively low-cost and uses cameras and image processing technology to make long-distance measurements without touching power lines. In addition, this method is highly automated and can achieve fast and continuous line detection.

However, visual measurements can be affected by illumination factors. At night or in foggy days, the quality of the images captured by the camera will be reduced due to insufficient light, and in bad weather environments, the clarity of the images is not high, thus affecting the accuracy of the measurement results.

In addition, similar to UAV aerial survey, visual measurement also indirectly obtains the location information of transmission towers and cannot guarantee particularly high accuracy.

2.1.4 Sensor measurement method

The sensor measurement method is to install sensors on the line mobile robot or directly on the transmission line, such as inclination sensors, acceleration sensors, etc., to monitor the motion and tilt information of the transmission line in real time through the sensors, and then obtain the sag value through data analysis. This method is not affected by illumination and can realize real-time monitoring. Its accuracy is improved compared with the visual measurement method, but it is still indirect measurement, and the sensor needs to be maintained and calibrated regularly, so there is still a problem that the accuracy is not high enough.

The sensor measurement method is realized by installing various sensors on the line mobile robot or directly on the transmission line, mainly including inclination sensor, acceleration sensor, strain gauge, etc., which can monitor the motion, tilt, tension and pressure parameters of the transmission line in real time. By analyzing these data, the sag value of the high-voltage transmission line can be accurately calculated.

The sensor has strong adaptability to the environment, especially when the lighting conditions are poor, it can still keep working normally. Moreover, the sensor can provide continuous monitoring data, which greatly improves the measurement efficiency of sag.

Although the accuracy of sensor measurement is higher than that of visual measurement, it is still indirect measurement. The measurement results are affected by many factors, such as the accuracy of the sensor itself, the accuracy of the installation position, etc. In addition, because the sensor is exposed to the external environment for a long time, environmental factors such as temperature and humidity will affect its performance and life, so it needs regular maintenance and calibration, and the operation and maintenance cost is high.

2.1.5 Ultrasonic measurement method

The ultrasonic measurement method measures the distance between the line and the ground through ultrasonic signals, and then calculates the sag value of the high-voltage line. Using ultrasonic waves for measurement is not affected by light and has a wide range of applications. However, ultrasonic equipment is greatly affected by climate, and wind speed and temperature may affect its measurement accuracy.

Ultrasonic measurement method can accurately measure the distance between high-voltage lines and the ground by sending ultrasonic signals and calculating their return time from the transmitting point to the receiving point. Since the propagation speed of ultrasonic waves is known, the sag value of high-voltage lines can be quickly calculated.

Lighting factors have little influence on ultrasonic measurement, so it can be measured by ultrasonic measurement whether it is day or night.

However, climate factors have great influence on ultrasonic measurement method. At high wind speeds, the ultrasonic signal deflects, causing its path and timing to be inaccurate, and the propagation speed of ultrasonic waves increases with increasing air temperature. Therefore, changes in wind speed and temperature will affect the propagation speed of ultrasonic waves, thus affecting the accuracy of measurement results. In addition, ultrasonic signals will also be reflected when encountering obstacles, resulting in signal loss. Therefore, when using ultrasonic measurement methods, it is necessary to ensure that there are no obstacles in the path of signal propagation.

2.1.6 RTK measurement method

RTK (Real - Time Kinematic) carrier phase differencing technology, which sends the carrier phase collected by the reference station to the user receiver for differencing and solving the coordinates.RTK measurement method is an emerging satellite positioning measurement method, which is capable of real-time positioning with centimeter-level accuracy outdoors.

RTK measurement method through the line mobile robot carrying RTK, using GPS, GLONASS, BeiDou and other satellite navigation systems, combined with the ground base station correction information, real-time measurements and record the accurate latitude, longitude and elevation information of the two adjacent transmission towers, and to get the robot's precise position (latitude, longitude and elevation), and then real-time data back to the control center, so as to determine the value of the arc plumb.

The RTK measurement method has a very high measurement accuracy, and the arc sag value of the high-voltage line is derived by directly measuring the position of each point. In addition, because the robot moves directly along the high-voltage line, its measurement activities are not easily affected by weather conditions, especially in bad weather conditions, and the stability and reliability of the RTK system far exceeds that of other methods.

Compared with UAV aerial surveys, the RTK measurement method not only has higher measurement accuracy, but also is capable of continuous operation on power lines and realizes long-time monitoring. Moreover, the RTK system carried by the line mobile robot can collect data continuously, reducing the information omission caused by intermittent data collection.

However, the equipment cost of the RTK system and the operation and maintenance cost of the ground base station are relatively high.

2.1. 7 High voltage line sag measurement scheme

Several main methods based on line mobile robots are compared, as shown in Table 2.1.

Table 2.1 Comparison of main methods of line mobile robot

Method	Accuracy	Cost	Weather adaptability	Processing difficulty	Applicability
Laser ranging method	Tall	Low	Poor	Low	Middle
Visual recognition method	Middle	Low	Middle	Middle	Middle
Sensor measurement method	Tall	Middle	Good	Tall	Tall
Ultrasonic measurement method	Middle	Low	Poor	Low	Middle
RTK measurement method	Extremely high	Tall	Good	Tall	Tall

There are many methods to measure the sag of high-voltage transmission lines, and each method needs to consider many influencing factors. Among them, the measurement accuracy is very important, mainly because the measurement accuracy will affect the subsequent adjustment of conductor sag. High measurement accuracy can directly and effectively keep the sag of transmission line within the specified range of design, and avoid the harm to the line and environment caused by the large or small sag value.

Through the previous analysis of various high-voltage transmission line sag measurement schemes and the comparison in Table 2.1, it is not difficult to find that the position information of the transmission tower can be measured most directly and accurately by using the line mobile robot equipped with RTK, thereby improving the accuracy. Moreover, the robot can move continuously on the high-voltage transmission line, and can obtain and calculate data in real time to ensure real-time performance.

2.2 Overall scheme

Comprehensive analysis of the above, the overall scheme of high-voltage line arc sag measurement is as follows: measuring the arc sag of high-voltage lines by line mobile robot with RTK measurement technology, obtaining the position information of two adjacent transmission towers and the real-time position of the robot, transmitting the data to the supercomputer, calculating the arc sag value of the high-voltage transmission line through the program and displaying it in real time.

2.3 Summary

In this chapter, by comparing the methods of high-voltage transmission line arc sag measurement and analyzing the advantages and disadvantages of various measurement methods based on the line mobile robot, the RTK measurement technology of the line mobile robot to measure the arc sag of high-voltage lines is finally selected. The arc sag algorithm will be studied in the following.

3 Research on Sag Algorithm of High Voltage Transmission Lines

3.1 Calculation principle of sag of high voltage transmission line

According to the catenary model, when the suspension points of the two towers climb, the sag of the high-voltage transmission line reaches the maximum value at the center of the gear pitch, and the horizontal position of the maximum sag point is located at the midpoint of the gear pitch; When the two towers are suspended at unequal heights, according to the calculation results of catenary equation, the maximum sag point appears at the side where the midpoint of the gear pitch is slightly higher than the suspension point. Through the catenary model, the horizontal position of the maximum sag point can be accurately calculated.

Based on the parabolic model, the arc droop value of the high-voltage transmission line under the action of self-weight load can be obtained. However, in the actual measurement, the line mobile robot, due to its own gravity, will produce a certain degree of concentrated load on the target transmission line, resulting in the deviation of the transmission line from the standard parabolic model, so that the measured arc sag value is greater than the actual arc sag value. If the measured value is compared with the design arc sag value according to this measured value, it will cause a negative deviation of arc sag, resulting in the actual tight line arc sag higher than the design value, making the line tension too large. Therefore, it is also necessary to consider the parabolic deformation after the introduction of the centralized load and correct it according to the actual situation.

Therefore, the specific method of measuring the sag of high-voltage transmission line is as follows: firstly, according to the catenary model, the horizontal position of the maximum sag point is obtained, and then according to the sag value of high-voltage transmission line obtained by the standard parabola model, combined with the maximum sag value under the load, the correction value of the influence of the self-weight load of line mobile robot on line sag is obtained. Finally, according to the observation calculation model and correction value of any point sag on high-voltage transmission line, the horizontal midpoint sag measurement calculation model is obtained.

3.2 Maximum sag point of high voltage transmission line under catenary model

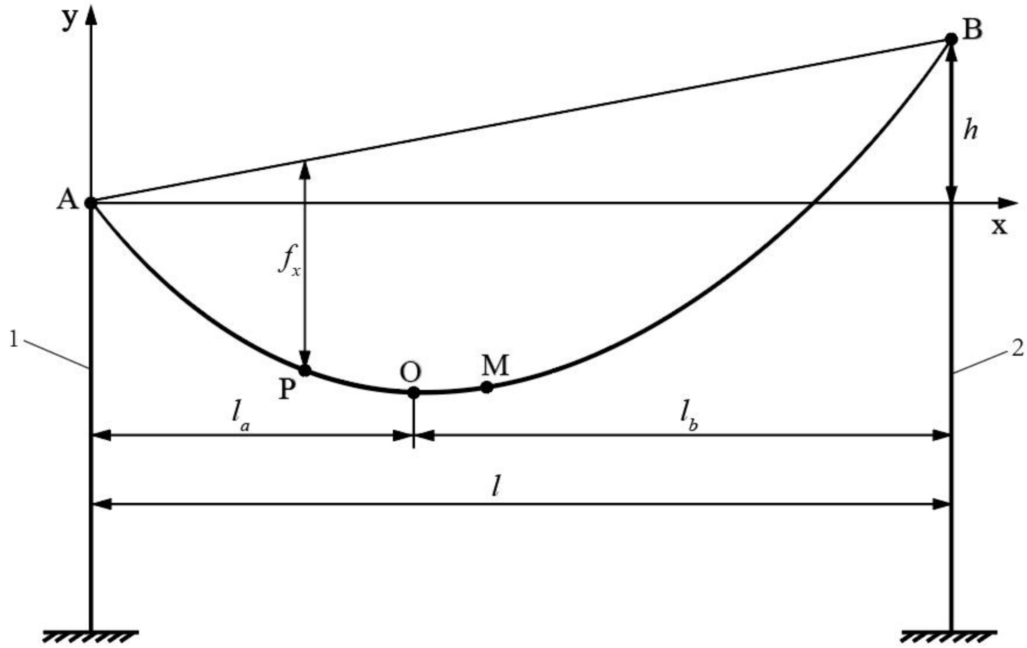

Figure 3.1 Catenary model

Fig. 3.1 shows the catenary model, and the coordinate system is established with the low hanging point A as the coordinate origin. Assuming that the catenary is suspended between points A and B by its own weight, and the catenary is flexible, the linear weight density is equal to ω, point O is the lowest point of the catenary, and the horizontal distance between point O and point A is l_a， The horizontal distance between point O and point B is l_b. And the gear spacing between the first transmission tower and the second transmission tower is l, the elevation difference is h。

In order to find the coordinates of the maximum point of sag, take any point P on the line (x， y), let the OP length be s。

The force analysis of the OP segment is carried out, as shown in Figure 3.2. The three forces acting on OP are horizontal tension H, tangential tension T and its own gravity ωs. Tension at point OH horizontally to the right, tension at point PT along the tangent direction of the point, assuming its direction is the horizontal direction θ point.

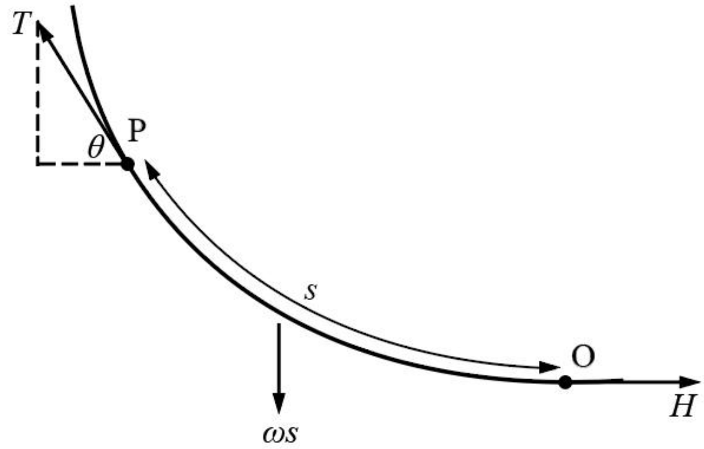

Fig. 3.2 Stress analysis of OP segment under catenary model

According to the static equilibrium equation, we can get

$$T\cos\theta = H \quad (3.1)$$

$$T\sin\theta = \omega s \quad (3.2)$$

Let the catenary equation be $y=y(x)$, has

$$\tan\theta = \frac{dy}{dx} = \frac{\omega}{H}s(x) \quad (3.3)$$

Pair of the above formula x derivation, there is

$$\frac{d^2y}{dx^2} = \frac{\omega}{H}\cdot\frac{ds(x)}{dx} \quad (3.4)$$

Differential equations according to arcs

$$ds(x) = \sqrt{1+y'^2}\,dx \quad (3.5)$$

Available

$$\frac{d^2y}{dx^2} = \frac{\omega}{H}\sqrt{1+y'^2} = \frac{dy'}{dx} \quad (3.6)$$

Separate the variables of equation (3.6) and integrate them, and we can obtain

$$\int\frac{dy'}{\sqrt{1+y'^2}} = \frac{\omega}{H}\int dx \quad (3.7)$$

By solving equation (3.7), we can get

$$ln\left(\sqrt{1+y'^2}+y'\right) = \frac{\omega}{H}x + c_1 \quad (3.8)$$

Because point O is the lowest point of the catenary, there is

$$y'|_{x=l_a} = 0 \quad (3.9)$$

Substitute equation (3.9) into equation (3.8) to obtain

$$c_1 = -\frac{\omega}{H} l_a \quad (3.10)$$

Substitute equation (3.10) into equation (3.8) and solve for y', have

$$y' = sh\left(\frac{\omega}{H}(x - l_a)\right) = tan\,\theta \quad (3.11)$$

For equation (3.11) proceed points, get

$$y = \frac{H}{\omega} ch\left(\frac{\omega}{H}(x - l_a)\right) + c_2 \quad (3.12)$$

Since the catenary passes through the origin, substituting into equation (3.12) yields

$$c_2 = -\frac{H}{\omega} ch \frac{l_a \omega}{H} \quad (3.13)$$

$$y = \frac{H}{\omega}\left[ch\left(\frac{\omega}{H}(x - l_a)\right) - ch\frac{l_a \omega}{H} \right] \quad (3.14)$$

It can be seen from the catenary model

$$y_{AB} = \frac{h}{l} x \quad (3.15)$$

From equations (3.14) and (3.15), it can be seen that the sag formula of any point P on the line is

$$f_x = \frac{h}{l} x - y = \frac{H}{\omega}\left[ch\frac{l_a \omega}{H} - ch\frac{(l_a - x)\omega}{H} \right] + \frac{h}{l} x \quad (3.16)$$

The maximum value of f_x is obtained when the derivative of f_x is 0, i.e.

$$f_x' = sh\frac{\omega}{H}(x - l_a) - \frac{h}{l} = 0 \quad (3.17)$$

It can be obtained from equation (3.17)

$$x = l_a + \frac{H}{\omega} sh^{-1}\frac{h}{l} \quad (3.18)$$

At this time, x is the abscissa of the sag point M when the sag is maximum.

Since the coordinates of suspension point B are (l,h), substituting it into equation (3.14), we can get

$$\frac{H}{\omega}\left[ch\left(\frac{\omega}{H}(l - l_a)\right) - ch\frac{l_a \omega}{H} \right] = h \quad (3.19)$$

Transform relationships using functions[15], Available

$$\frac{2H}{\omega} \cdot sh\frac{\omega l}{2H} \cdot sh\frac{\omega(l - 2l_a)}{2H} = h \quad (3.20)$$

Solve the above formula and get

$$l_a = \frac{l}{2} - \frac{H}{\omega} sh^{-1}\left(\frac{\omega h}{2H \cdot sh \dfrac{\omega l}{2H}} \right) \quad (3.21)$$

Substituting equation (3.21) into equation (3.18), the abscissa of the maximum sag point M can be obtained as

$$x = \frac{l}{2} - \frac{H}{\omega} sh^{-1}\left(\frac{\omega h}{2H \cdot sh \dfrac{\omega l}{2H}} \right) + \frac{H}{\omega} sh^{-1} \frac{h}{l} \quad (3.22)$$

The horizontal coordinate x of the point M with the maximum arc droop is used for navigational use, and the line mobile robot navigates a horizontal distance of AM = x when it starts from point A, and a horizontal distance of BM = l-x when the robot starts from point B.

3.3 Algorithm for sag of high-voltage transmission lines under parabolic model

3.3. 1 Algorithm for sag of high-voltage transmission lines without load

The horizontal position of the maximum sag point has been known from the previous section, and this section will be based on the parabolic model when analyzing the maximum point of sag value.

When only the self-weight of high-voltage transmission line is considered and there is no other load, the transmission line can be approximately regarded as a standard parabolic model, and the maximum sag value at the maximum sag point can be easily obtained by the standard parabolic model. However, in actual operation, the gravity of the line mobile robot will have a good effect on target transmission lines generate concentrated loads effect so that the measured sag value is greater than the actual relaxation value. Therefore, it is necessary to carry out sag error value correction.

The maximum sag value is first calculated based on the standard parabolic model and then determined in the next section sag correction value.

Fig. 3.3 is a standard parabolic model. When there is no load and the transmission line is suspended at different heights, the coordinate system is established with the lowest point as the coordinate origin. Wherein, l is a gear distance between the first transmission tower and the second transmission tower, h is the elevation difference between the two towers, φ is the elevation angle of the height difference between the two towers.

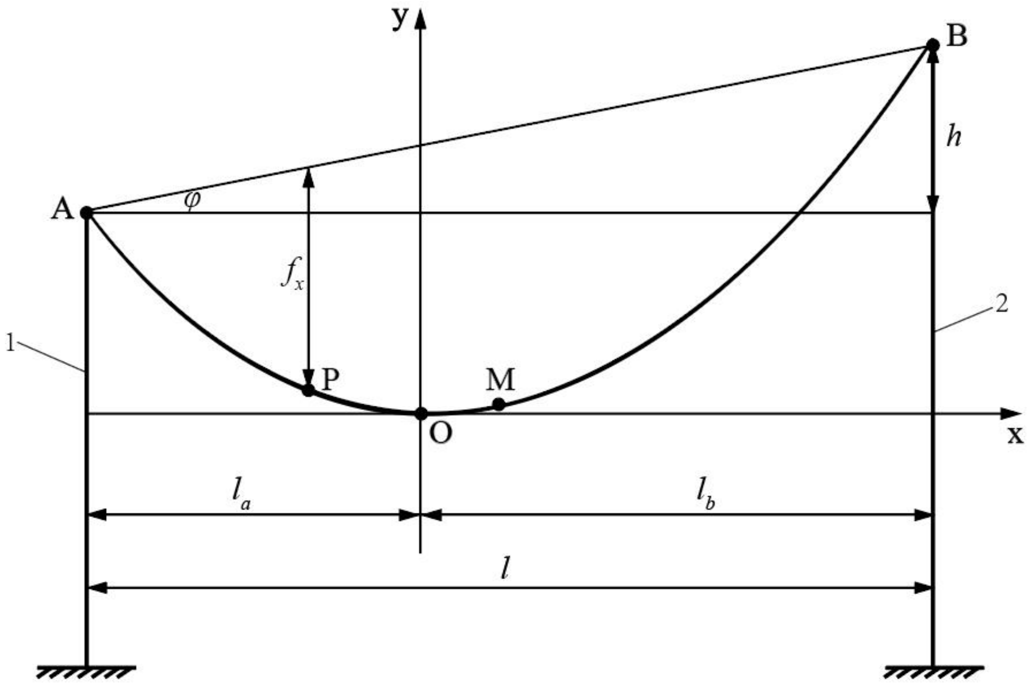

Figure 3.3 Standard parabolic model

Calculate any point P on the line (x， y) of sag. The force analysis for the OP segment is similar to that in the previous section. As shown in Figure 3.4, the three forces acting on the OP are horizontal tension, respectively H and tangential tension T and its own gravity $\omega x/\cos\varphi$。 Tension at point OH horizontally to the right, tension T at point P along the tangent direction of the point, its direction is the same as the horizontal direction θ point. Because the gravity of the transmission line is evenly distributed, the gravity of the OP section can be equivalent to the straight line AB, so the gravity on OP is $\omega x/\cos\varphi$。

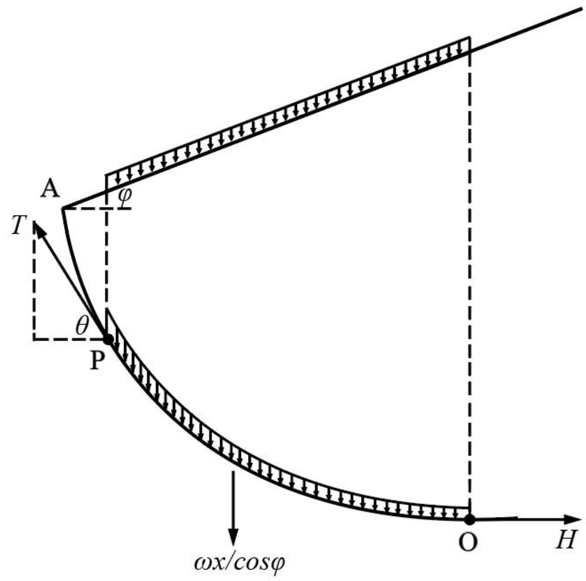

Figure 3.4 Stress analysis of OP segment under standard parabolic model

According to the static equilibrium equation, we can get

$$T\cos\theta = H \quad (3.23)$$

$$T\sin\theta = \frac{\omega x}{\cos\varphi} \quad (3.24)$$

Therefore, it can be obtained

$$\tan\theta = \frac{dy}{dx} = \frac{\omega x}{H\cos\varphi} \quad (3.25)$$

Pair of equation (3.25) proceed points, get

$$y = \frac{\omega x^2}{2H\cos\varphi} + c_1 \quad (3.26)$$

Because the lowest point of the parabola passes through the origin, soc_1= 0, so the parabolic equation is

$$y = \frac{\omega x^2}{2H\cos\varphi} \quad (3.27)$$

The straight line AB intersects the parabola at A (-l_a, h_1), B (l_b, h_2) two points, the elevation difference between two points AB is$h$, the gear pitch is$l$, so the equation of straight line AB is

$$y_{AB} = \frac{h}{l}x + c_2 \quad (3.28)$$

Simultaneously establish equations (3.27) and (3.28), and then according to Vedas theorem, we can get

$$l_b - l_a = \frac{h}{l} \cdot \frac{2H\cos\varphi}{\omega} = \frac{2H\sin\varphi}{\omega} \quad (3.29)$$

Because $l_a+l_b=l$, available

$$l_a = \frac{l - \frac{2H\sin\varphi}{\omega}}{2}, \quad l_b = \frac{l + \frac{2H\sin\varphi}{\omega}}{2} \quad (3.30)$$

By (3.27) formula can be obtained,

$$h_1 = \frac{\omega l_a^{\ 2}}{2H\cos\varphi}, \quad h_2 = \frac{\omega l_b^{\ 2}}{2H\cos\varphi} \quad (3.31)$$

Place point A ($-l_a$, h_1) substituted into (3.28) formula, get

$$c_2 = h_1 - \frac{h}{l}(-l_a) = \frac{l^2\omega}{8H\cos\varphi} - \frac{H\cos\varphi}{2\omega}\left(\frac{h}{l}\right)^2 \quad (3.32)$$

Therefore, the equation of AB is

$$y_{AB} = \frac{h}{l}x + \frac{l^2\omega}{8H\cos\varphi} - \frac{H\cos\varphi}{2\omega}\left(\frac{h}{l}\right)^2 \quad (3.33)$$

From equations (3.27) and (3.33), it can be seen that the sag of any point P on the line is

$$f_x = y_{AB} - y = \frac{l^2\omega}{8H\cos\varphi} - \frac{H\cos\varphi}{2\omega}\left(\frac{h}{l}\right)^2 + \frac{h}{l}x - \frac{\omega}{2H\cos\varphi}x^2 \quad (3.34)$$

According to Vedas theorem, when x take

$$x = \frac{l_b - l_a}{2} = \frac{H}{\omega}\sin\varphi \quad (3.35)$$

At this time, the maximum value f_x is obtained, and the maximum sag point M is located at the midpoint of the gear pitch, so the sag at the midpoint of the gear pitch is

$$f_M = \frac{l^2\omega}{8H\cos\varphi} \quad (3.36)$$

3.3. 2 Correction algorithm under concentrated load

Under actual circumstances, the transmission line will be subject to the gravity load of the line mobile robot. Therefore, the transmission line will be subjected to the gravity load of the line mobile robot.This section will correct the sag values under concentrated loads.

As shown in Figure 3.5, in the isolated gear with unequal height suspension, the coordinate system is established with the low suspension point as the coordinate origin, and the load is concentrated Q acting on point C, the horizontal distance from point C to point A is a.

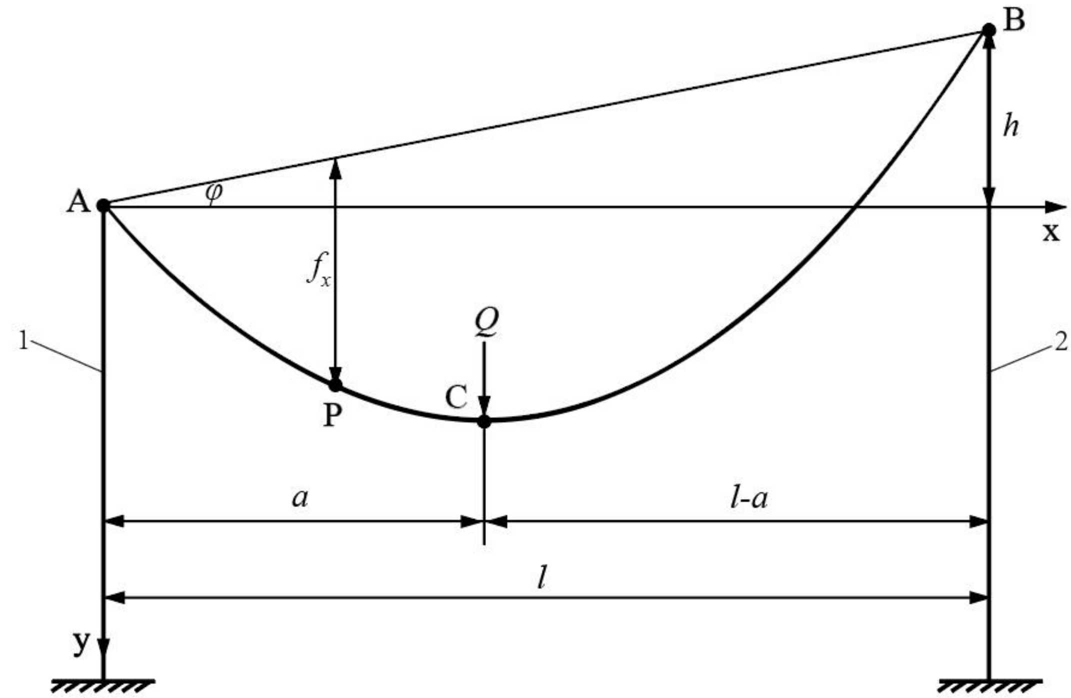

Fig. 3.5 Parabolic model under concentrated load

First, an arbitrary target point P on the high-voltage transmission line is calculated (x,y) The sag value located on the side of point C near the origin. As shown in Figure 3.6 and Figure 3.7, the force analysis of the high-voltage transmission line and AP section is similar to that of the OP section under the standard parabolic model.

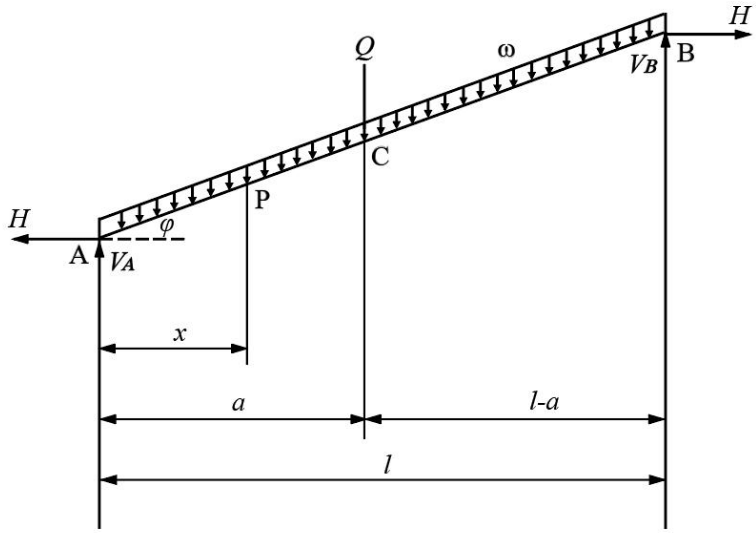

Fig. 3.6 Stress analysis of high voltage transmission lines under concentrated load

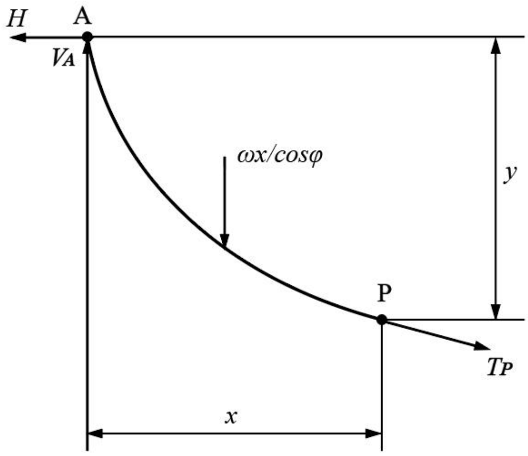

Fig. 3.7 Stress analysis of AP section of parabolic model under concentrated load

Such as figure 3.7Shown in, taking the moment for point P, there is

$$H \cdot y = V_A \cdot x - \frac{x\omega}{cos\,\varphi} \cdot \frac{x}{2} \quad (3.37)$$

By separating the variables of equation (3.37), we get

$$y = \frac{V_A \cdot x - \dfrac{\omega x^2}{2cos\,\varphi}}{H} \quad (3.38)$$

Find the upward support force on suspension points A and B, such as figure 3.6As shown, for suspension point B takes the moment. According to the static balance condition, there is

$$V_A \cdot l + H \cdot h - \frac{l}{2} \cdot \frac{\omega l}{cos\,\varphi} - Q(l-a) = 0 \quad (3.39)$$

Solve equation (3.39) to obtain

$$V_A = \frac{\omega l}{2cos\,\varphi} + \frac{l-a}{l}Q - \frac{h}{l}H \quad (3.40)$$

In the same way, there are

$$V_B = \frac{\omega l}{2cos\,\varphi} + \frac{a}{l}Q + \frac{h}{l}H \quad (3.41)$$

Substituting equation (3.40) into equation (3.38), we can obtain

$$y = \left(\frac{\omega l}{2H\,cos\,\varphi} + \frac{l-a}{l} \cdot \frac{Q}{H} - \frac{h}{l}\right)x - \frac{\omega x^2}{2H\,cos\,\varphi} \quad (3.42)$$

Therefore, the sag at any point on the high-voltage transmission line is

$$f_x = y_{AB} + y = \left(\frac{\omega l}{2H\cos\varphi} + \frac{l-a}{l} \cdot \frac{Q}{H} \right) x - \frac{\omega x^2}{2H\cos\varphi} \quad (3.43)$$

Order $x=a$, at this time, the sag value of the concentrated load action point C is

$$f_c = \frac{a(l-a)}{2H}\left(\frac{\omega}{\cos\varphi} + \frac{2Q}{l} \right) \quad (3.44)$$

Equation (3.44) is used to calculate the sag value of the action point of the robot when the line mobile robot is running.

Take the derivative of equation (3.44) to obtain

$$f'_c = \frac{\frac{\omega}{\cos\varphi} + \frac{2Q}{l}}{2H}(l-2a) \quad (3.45)$$

Order $f_c'=0$, get when $a=l/2$, the sag reaches the maximum value, that is, under the action of concentrated load, the maximum sag value of the high-voltage transmission line appears at the midpoint of the gear pitch. At this time, the sag value of the midpoint under concentrated load is

$$f_{c\,max} = \frac{l^2}{8H}\left(\frac{\omega}{\cos\varphi} + \frac{2Q}{l} \right) \quad (3.46)$$

Comparing equations (3.36) and (3.46), it can be seen that under concentrated load, the sag increment at the maximum point of sag of high-voltage transmission line is

$$f_{\Delta} = \frac{Ql}{4H} \quad (3.47)$$

Similarly, the same sag increment result can be obtained when it is verified that point P is located on the side of point C far away from the origin. Therefore, when the line mobile robot is working, the sag increment generated at the maximum sag point, that is, the midpoint of the gear pitch, is $Ql/4H$。

3.4 High voltage transmission line sag algorithm

Such as figure 3.8,The line mobile robot is equipped with RTK to measure the latitude, longitude and elevation of point A and point B. Then from the abscissa of the maximum sag point M*x n*avigate to the point and measure the points elevation h_{Measure}。Thus, the elevation difference between adjacent transmission towers *h* I.e. h_B-h_A, while the gear pitch *l* then it is converted by the latitude and longitude of two points AB.

According to the formula (3.47), the measured value of sag of high-voltage transmission line is corrected, and it can be obtained

$$f_{测} = \frac{h_B - h_A}{l} \cdot x + h_A - h_{测} - \frac{Ql}{4H} \quad (3.48)$$

Wherein, $x = \frac{l}{2} - \frac{H}{\omega} sh^{-1}\left(\frac{\omega h}{2H \cdot sh\frac{\omega l}{2H}} \right) + \frac{H}{\omega} sh^{-1}\frac{h}{l}$ 。

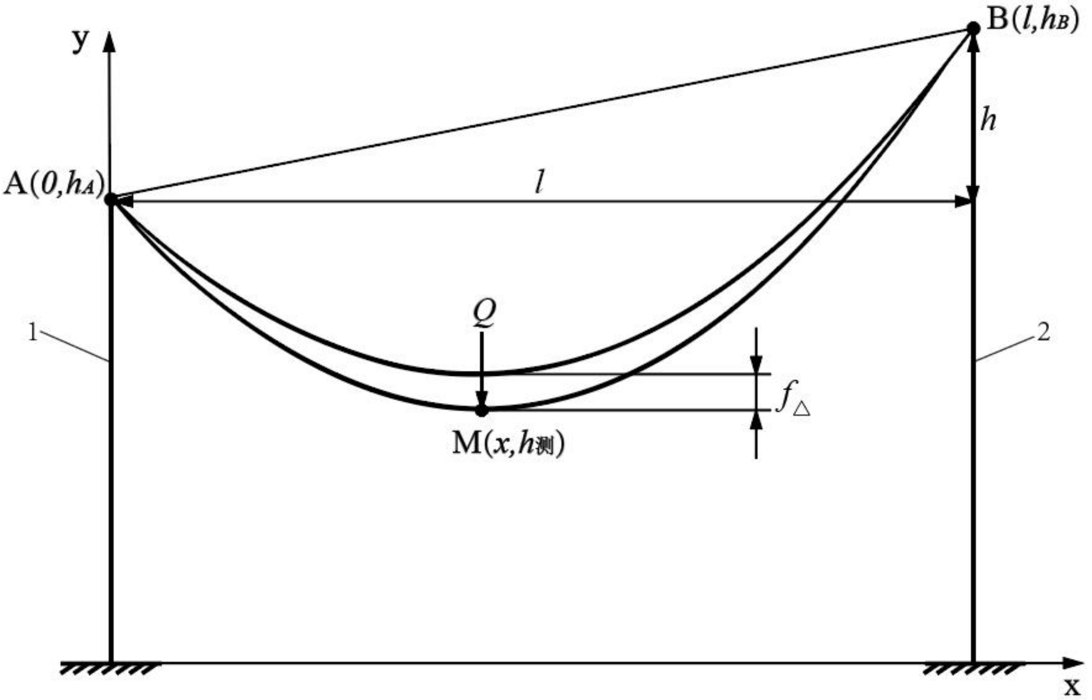

Fig. 3.8 Schematic diagram of measured sag of high-voltage transmission line

3.5 Summary

In this chapter, through the research of sag algorithm, I build up bobot algorithm system for sag measurement and error correction.From the above formula, it can be seen that the parameters required for measuring the sag value of high-voltage transmission lines mainly include: self-gravity per unit length of conductor ω, the dead weight of the line mobile robot Q horizontal tension of high-voltage transmission lines H. The elevation difference between the first transmission tower and the second transmission tower h and gear pitch l Etc，Wherein,ω、Q and H all are known parameters, h and l unknown, subject to RTK, realize the measurement of the position information (longitude, latitude, elevation) of the suspension points of the two transmission towers, so as to obtain elevation difference between two adjacent transmission towers h and gear pitch l the specific value of.

Next step the RTK will be selected and tested for measurement accuracy. In addition, it is necessary to test the accuracy of the conversion method of latitude and longitude and gear pitch.

4 RTK and Accuracy Testing

4.1Introduction to RTK

4.1.1Mozi 4G RTK

The sag algorithm of high-voltage line has been studied above, and the unknowns in the algorithm need to be measured by RTK. The RTK selected this time is Mozihao 4G RTK。

Mozihao 4G RTK is a development board developed by Mozihao Technology, with a 4G module block, can realize remote RTK. The development board has STM32 to provide source code and is connected to the display screen, which can realize the display of data such as latitude, longitude and elevation.

The development board supports the use of mobile station RTK devices in environments where mobile remote acquisition of differential data is required. Users only need to insert a SIM card and configure an RTK account to achieve remote acquisition of RTK data. In addition, it also supports the use of fixed base stations, supports the use of self-built base stations, and sends differential data to different rovers through 4G MQTT, which can avoid the trouble of wiring and difficulty in obtaining differential data, and greatly facilitates users use.

Figure 4.1 isMoziStructural diagram of the 4G RTK backplane.

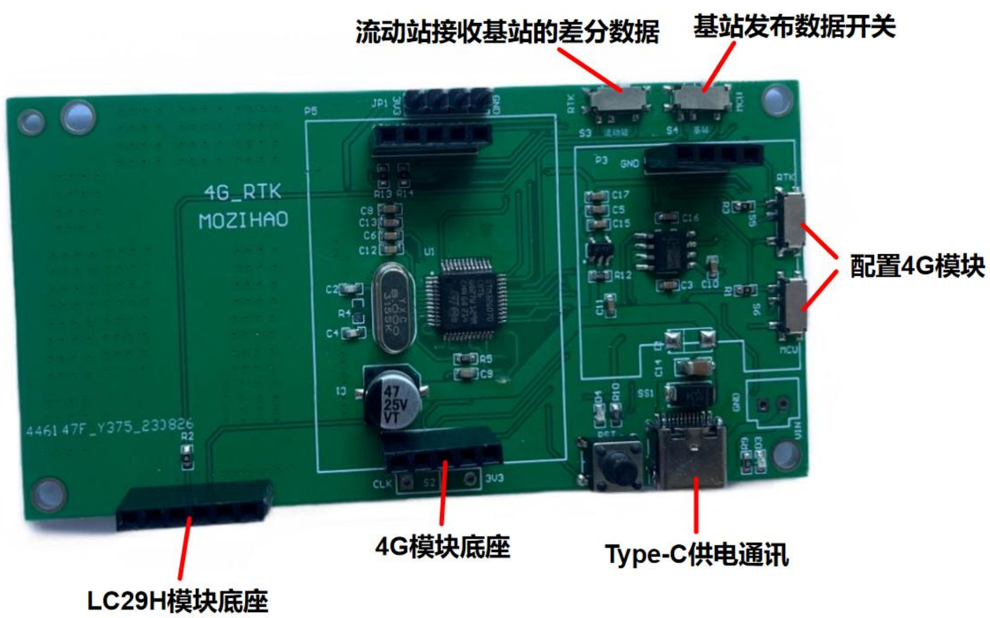

Figure 4.1 Mozihao 4G RTK bottom plate structure diagram

There are two toggle switches on the top right corner of the base plate, which are used for the self-built base station situation, if the self-built base station is not used, these two buttons are useless. The left switch is used to control the 4G module to receive the differential data of RTK and then send it to the LC29H mobile station module to realize RTK positioning, in which the differential data of RTK is sent from the base station, using the MQTT protocol; the right switch controls the base station module to transmit the differential data remotely via 4G data. Therefore, if you need to build your own base station, you need two 4G modules, one of which does the sending of differential data by the 4G module and the other does the receiving of differential data by the 4G module.

The two buttons on the right side of the base plate are used to configure the parameters for using the 4G module, such as the server platform account name and password for RTK, or as a self-built base station to set up the MQTT IP address publishing address as well as subscription.

As shown in Figure 4.2, the LC29H module is inserted on the left side of the base plate with fixing screw holes for fixing the module, 2 double rows of nuts in the middle are used for inserting the 4G module, and the OLED display is inserted on the right side to realize the function of RTK.

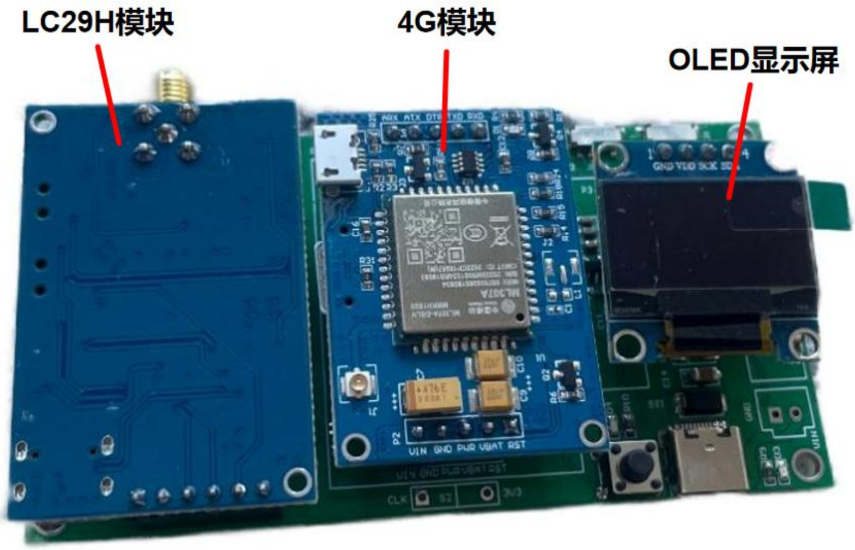

Figure 4.2 Mozihao 4G RTK backplane access module

4.1.2 Haida TS5 Pro Intelligent Inertial Navigation RTK

Because the measurement accuracy of Mozihao 4G RTK is not yet known, it will pass Haida TS5 Pro intelligent inertial navigation RTK(All below are represented by Haida RTK), to test the measurement accuracy.

The accuracy of Zhonghaida RTK is very high, but due to its large volume and mass, it cannot be mounted on line mobile robots. Therefore, its measured value is used as a calibration value and compared with the measured value of Mozihao 4G RTK.

Zhonghaida RTK-TS5 Pro inertial guidance version, integrated with intelligent inertial guidance system and real-time dynamic differential technology, is lightweight and compact, easy to operate, and can realize fast measurement.TS5 Pro supports BeiDou 3 satellite signals, the number of satellites for searching and solving reaches more than 50, and the signal reception is very sensitive, supporting single BeiDou solving. Its built-in national CORS service enables centimeter-level positioning. The integrated inertial navigation system can also maintain the device's positioning and navigation functions in environments with poor GPS signals, with strong adaptive and anti-interference capabilities.

Fig. 4.3 and Fig. 4.4 show the physical drawings of Zhonghaida RTK body and handbook.

Figure 4.3 Zhonghaida RTK fuselage

Figure 4.4 Zhonghaida RTK Handbook

4.2 Accuracy test

4.2. 1 Measurement data accuracy test

Take advantage of Zhonghaida RTK for multiple locations within the school conduct field measurements, the measured locations include the west gate, east gate, south gate, bell tower, library, etc. of the school, as shown in figure 4.5 shown in.

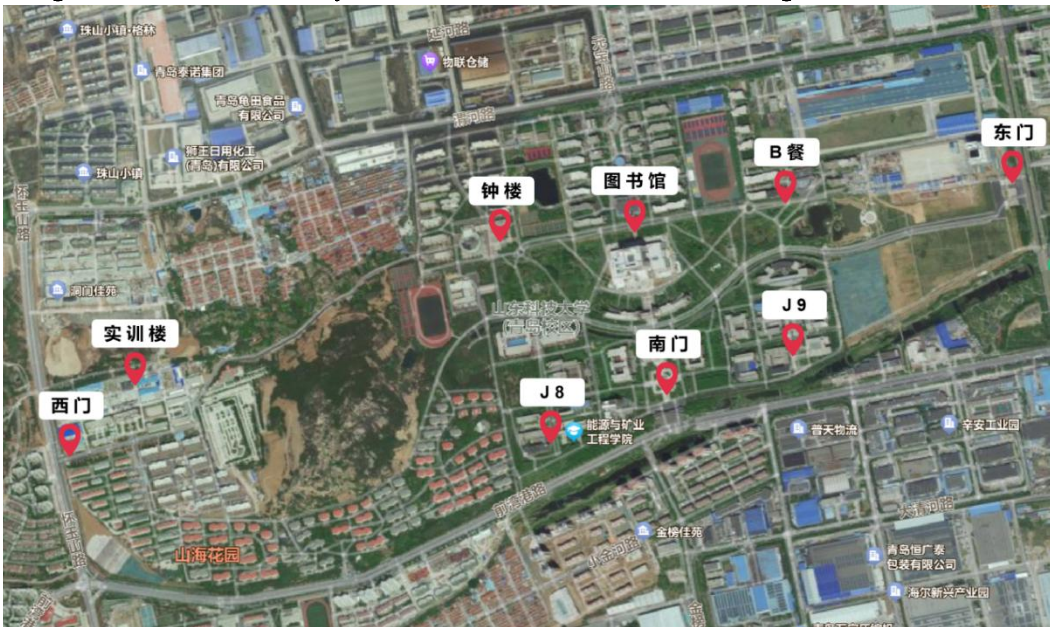

Figure 4.5School plan

Through these locations measurement data, such as accuracy, latitude, elevation and other information, again by the systems built-in algorithm computable get out the distance between the points。

Figure 4.6 is a field measurement photo of Zhonghaida RTK, figure 4.7 is the measured data of each group.

Figure 4.6 Zhonghaida RTK Field Measurement

Figure 4.7 Zhonghaida RTK Measurement Data

Among them, the unit of latitude and longitude is degrees, minutes and seconds, and the unit of elevation is meters. As shown in Table 4.1, multiple positions are selected for multiple measurements, the measured data are sorted out, and the latitude and longitude are converted into degrees to facilitate subsequent calculations.

Table 4.1 Zhonghaida RTK measurement point data

Location	Latitude (degrees,	Longitude (degrees,	Latitude	Longitude	Elevation

West Gate	35:59:53.86621	120:06:01.86632	35.998296	120.100518	77.832
	35:59:53.86657	120:06:01.86793	35.998296	120.100519	77.864
	35:59:53.86562	120:06:01.86650	35.998296	120.100518	77.8627
Clock tower	36:00:13.32640	120:06:51.28363	36.003702	120.114245	42.5624
	36:00:13.32625	120:06:51.28137	36.003702	120.114245	42.597
	36:00:13.32611	120:06:51.28360	36.003702	120.114245	42.6062
Library	36:00:14.56802	120:07:06.27119	36.004047	120.118409	37.3544
	36:00:14.56721	120:07:06.26957	36.004046	120.118408	37.3537
	36:00:14.56553	120:07:06.26812	36.004046	120.118408	37.3517
East Gate	36:00:19.21453	120:07:49.39163	36.005337	120.130387	30.2648
	36:00:19.21391	120:07:49.39127	36.005337	120.130386	30.2525
South Gate	35:59:59.57711	120:07:10.14081	35.999883	120.119484	37.0394
	35:59:59.57731	120:07:10.14112	35.999883	120.119484	37.0401
Training Building	36:00:01.20490	120:06:09.33769	36.000335	120.102594	76.1474
B Meal	36:00:17.72457	120:07:25.58654	36.004923	120.123774	33.1339
J9	36:00:03.17890	120:07:25.28720	36.000883	120.123691	32.6195
J8	35:59:55.38848	120:06:57.07460	35.998719	120.115854	41.2713

The average value of the data of each measurement point is processed, and the results are as followsTable 4. 2.

Table 4.2 Average value of Zhonghaida RTK measurement point data

Location	Average latitude	Average longitude	Average elevation (m)
West Gate	35.998296	120.1005183	77.8529
Clock tower	36.003702	120.114245	42.58853
Library	36.0040463	120.1184083	37.35327
East Gate	36.005337	120.1303865	30.25865
South Gate	35.999883	120.119484	37.03975
Training Building	36.000335	120.102594	76.1474
B Meal	36.004923	120.123774	33.1339
J9	36.000883	120.123691	32.6195
J8	35.998719	120.115854	41.2713

Use Mozihao 4G RTK Measure Zhonghaidain the upper section again RTK measured position, comparison both give an error value， From this, it is judged whether the RTK can be used to measure the line sag value in conjunction with the line mobile robot.

Figure 4.8Measured with Mozihao 4G RTK physical drawing with data.

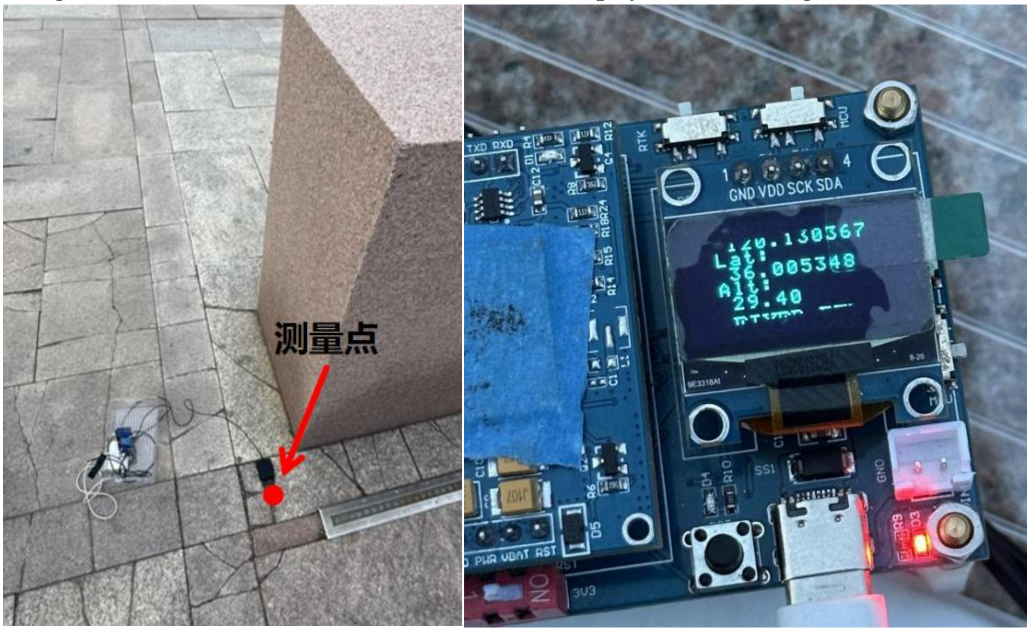

Figure 4.8 Mozihao 4G RTK measurement chart

Table 4.3 shows each set of data measured with Mozihao 4G RTK.

Table 4.3 Mozihao 4G RTK measurement point data

Location	Latitude (degree)	Longitude (degree)	Elevation (m)
West Gate	35.998286	120.100516	77.8
Clock tower	36.003692	120.114243	42.6
East Gate	36.005348	120.130367	30.3
South Gate	35.999872	120.11948	37.0
Training Building	36.000372	120.102578	75.6

By comparing Table 4.2 with Table 4.3, it can be found that the data measured by Mozihao 4G RTK is basically the same as that of Zhonghaida RTK, and the error is shown in Table 4.4.

Table 4.4 Mozi 4G RTK data error

Location	Latitude error	Longitude error	Elevation error
West Gate	0.0000278%	0.0000019%	0.068%
Clock tower	0.0002777%	0.0000017%	0.027%
East Gate	0.0000306%	0.0000162%	0.137%

South Gate	0.0000306%	0.0000033%	0.107%
Training Building	0.0001028%	0.0000133%	0.719%

The above table shows that the error of latitude and longitude is around 0.0001% and the error of elevation is around 0.2%. Therefore, the measurement accuracy of Mozihao 4G RTK is very high, and it can be used to measure the sag by the line mobile robot equipped with Mozihao 4G RTK.

4.2. 2 Selection of data conversion method

Through the analysis of the previous algorithm, we can know that under the condition of concentrated load, we only need to know the elevation difference between the first transmission tower and the second transmission tower by using the error-corrected parabola model *h a*nd horizontal distance *l*, the sag value of the line can be calculated.

Two transmission towers elevation difference between of it is relatively easy to obtain. The elevations of two transmission towers can be directly measured with RTK, and the two can be subtracted. However, the horizontal distance between the two towers cannot be directly obtained through RTK and needs to be obtained through a series of conversions.The main conversion methods are coordinate projection method and formula conversion method.

（1）Coordinate projection law

Because RTK can measure two transmission towers, the longitude and latitude values of the hanging point, so the distance between the two points can be calculated by coordinate conversion by projecting the longitude and latitude into the new spatial coordinate system.

Common coordinate projection methods include Mercator projection, Lambert orthogonal cone projection, equidistant projection, Robinson projection, etc.

① Mercator Projection (Mercator Projection)

The Mercator projection is a cylindrical projection that projects the Earths surface onto a fictional cylinder surrounding the earth, but is severely deformed at high latitudes, and the area and shape are no longer realistic.

② Lambert Conformal Conic Projection (Lambert Conformal Conic Projection)

The Lambert orthogonal cone projection projects the Earths surface onto a cone that touches two standard parallels, maintaining the truth of angles, but the accuracy of distance and area is best only near the standard parallels

③Equidistant projection(Equidistant Projection)

The equidistant projection guarantees the truth of the distance, starting from the center point, the distance to any other point is accurate, but there may be distortions for other points;

④Robinson projection(Robinson Projection)

Robinson projection is an eclectic projection that reduces the shape and size distortion of the polar region, but none of the attributes is completely accurate, such as distance, area, shape, orientation, etc.

The above coordinate projection methods have their own advantages and disadvantages, but they are all calculated by converting latitude and longitude into spatial coordinates level distance, so this type of projection method belongs to indirect conversion, and the accuracy not tall。

（2）Formula conversion method

Compared with indirect conversion, the latitude and longitude of two points are directly converted into between two points level the accuracy of the distance will be much improved. Wherein, formula the conversion method is further divided into Haversine (semi-positive vector) formulas and Vincenty formula.

① Haversine formula

Haversine formula based on spherical geometry, it assumes that the earth is a perfect sphere, and then uses spherical trigonometry to calculate the difference between two points level distance.

The Haversine formula is shown in formula (4.1), which is relatively simple, easy to implement, and has high computational efficiency. It is suitable for most situations where distance estimation needs to be quickly estimated and the accuracy requirements are not particularly high. However, it assumes that the earth is a perfect sphere, ignores the oblateness of the earth itself, and its accuracy is low in areas with higher latitudes and when calculating latitudes and longitudes with large spans.

$$d = 2R \cdot sin^{-1}\left(\sqrt{sin^2\left(\frac{\Delta\theta}{2}\right) + cos(\theta_1) \cdot cos(\theta_2) \cdot sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (4.1)$$

Wherein, d Represents the large circular path distance between two points,RRepresents the radius of the earth,θ_1Andθ_2Represent the latitudes of two points respectively, $\triangle\theta$ represents the latitude difference, $\Delta\lambda$ represents longitude difference, all in radians.

② Vincenty formula

The Vincenty formula is based on the ellipsoidal model, takes into account the oblateness of the earth, and involves iterative calculations, which can more truly reflect the distance between two points on the earths surface, especially for long-distance calculations, with very high accuracy, but convergence problems will be encountered when two points are very close or located at almost the same latitude and longitude positions.

From the above analysis, it can be obtained that the Haversine formula is suitable for the situation where fast calculation is needed and the accuracy is not required, while Vincenty formula is suitable for distance measurements requiring very high accuracy. Because the accuracy of sag measurement is high, and the line mobile robot needs high-precision navigation, so the more complex but very high-precision sag measurement is selected Vincenty formula。

4.2.3Vincenty formula accuracy test

Vincenty formula is based on the ellipsoid model, and two key ellipsoid parameters are used, namely the semi-major axis a and flattening rate f. a denotes the maximum radius of the ellipsoid, i.e. the distance from the center to the equator, f denotes the degree of flattening of the ellipsoid, represented by ($1- b/a$) is calculated, where b is the semi-minor axis (the distance from the center to the poles).

The operation steps of Vincenty formula is as follows:

Input: Enter the geographic coordinates of two points, including longitude and latitude;

Initialization: According to the latitude and longitude difference and ellipsoid parameters, set the initial value and prepare for iteration;

Iterative calculation: Through trigonometric functions and inverse trigonometric functions, the distance and direction are iteratively updated until the change amount is very small and the accuracy requirements are met;

Output: Calculates the distance between two points.

The computation process of Vincenty formula is complicated, involving many iterative steps and trigonometric operation, and convergence problems will occur when the distance between two points is close or close to the inverse point。 But itAble to calculate distances of thousands of kilometers with minimal errors，For most engineering projects, the applicability of Vincenty formula than Haversine formula is better.

Appendix 1 is the Python code for Vincenty formula conversion, which can quickly realize the difference between two points level measurement of distance.

Figure 4.9As an example of implementing the algorithm through Python, according to the average latitude and longitude of the west gate measurement point and the east gate measurement point of the school measured by Haida RTK, the difference between these two points is calculated level distance.

```
def vincenty_distance(lat1, lon1, lat2, lon2, a=6378137.0, b=6356752.314245, f=1 / 298.257223563):
        # Iteration didn't converge
        return None

    u2 = cos2Alpha * (a ** 2 - b ** 2) / (b ** 2)
    A = 1 + u2 / 16384 * (4096 + u2 * (-768 + u2 * (320 - 175 * u2)))
    B = u2 / 1024 * (256 + u2 * (-128 + u2 * (74 - 47 * u2)))
    deltaSigma = B * sinSigma * (cos2SigmaM + B / 4 * (cosSigma * (-1 + 2 * cos2SigmaM ** 2) -
                                                     B / 6 * cos2SigmaM * (-3 + 4 * sinSigma ** 2) * (
                                                                 -3 + 4 * cos2SigmaM ** 2)))

    s = b * A * (sigma - deltaSigma)

    return s


distance = vincenty_distance( lat1: 35.998296,  lon1: 120.1005183,  lat2: 36.004923,  lon2: 120.123774)
print(distance)
```

```
scratch
D:\python38\python.exe C:\Users\lenovo\AppData\Roaming\JetBrains\PyCharmCE2024.1\scratches\scratch.py
2221.9760619513204

Process finished with exit code 0
```

Figure 4.9Vincenty Formula Calculation Example

According to the built-in algorithm of Haida RTK, the measurement points between each measurement point can be obtained actual level distance, compare the results of the Vincenty algorithm implemented by Python code with the results of the built-in algorithm, and the comparison results are shown in Table 4.5.

Table 4.5 Comparison between Vincenty algorithm results and Haida RTK ranging

	Haida RTK ranging	Vincenty formula	Error
West Gate-Clock T	1375.3482 m	1375.33704547 m	0.0008110%
West Gate-Library	1734.6393 m	1734.61477274 m	0.0014140%
West Gate-B Meal	2222.0063 m	2221.97606195 m	0.0013608%
Simon-J9	2108.9192 m	2108.97338619 m	0.0025694%
Simon-J8	1382.6877 m	1383.54567524 m	0.0620512%
Ximen-Training Building	293.5698 m	293.62175180 m	0.0176966%

As can be seen from the above table, pass Vincenty Formula convert data, the accuracy is very high, the error is about 0.014%, and the speed at which the result is obtained by running the code very fast, can fully meet the accuracy and real-time performance requirements of。

At the same time, it is verified again by Vincentys formula Mozihao 4G RTK the measurement accuracy of. As shown in Table 4.6, compare the distances between points measured by Haida RTK and Mozihao 4G RTK.

Table 4.6 Different RTK measurements calculated based on Vincenty quantity point spacing

	Zhonghaida RTK	Mozihao 4G RTK	Error
West Gate-Clock T	1375.33704547 m	1375.36152626 m	0.0017800%
West Gate-East Gate	2804.00306963 m	2803.16418834 m	0.0299172%
West Gate-South Gate	1719.07977393 m	1718.91616227 m	0.0095174%

The above table once again verifies the accuracy of Mozihao 4G RTK measurement data, and ensures that the horizontal distance error between the first transmission tower and the second transmission tower is extremely small.

4.3 Summary

In this chapter, by selecting the appropriate RTK and conducting a series of accuracy tests,The line mobile robot can measure position information such as latitude, longitude and elevation of high-voltage transmission lines by RTK, and then convert the measured data into elevation difference between two adjacent transmission towers h and horizontal distance l from this, the sag value of high-voltage transmission line can be accurately obtained through the algorithm in Chapter 3. In the next step, the user interface will be designed through programming, so that the sag value can be displayed in real time and compared with the designed sag value, realize the visualization of the results。

Chapter 5 User Interaction Interface and Program Implementation

5.1 Implementation of user interaction interface

Through Chapter 3 and Chapter 4, the sag algorithm has been determined, and the required data of the algorithm can be measured in real time by RTK. This chapter will write program code to achieve RTK measurement technology. The sag calculation of high-voltage transmission lines is calculated, and the data is displayed in real time through the interactive interface.

5.1. 1 User interaction interface design scheme

User interactive interface of the design mainly includes requirements analysis, design principles, interface layout, control selection, event handling, interface beautification, testing and optimization, etc.

(1) Demand analysis

Before designing the user interaction interface, we must first conduct requirements analysis, mainly considering the functional requirements and operation difficulty of the interface, so as to ensure that the data measured by RTK and the results calculated by algorithms can be clearly displayed on the interface.

(2) Design principles

When designing the interface, it is necessary to ensure that the interface is concise, clear, intuitive and easy to use, avoid unnecessary complexity and confusion on the interface, keep the whole interface unified in style, layout and operation, conform to users intuition and habits, and feed back the results to users in a timely and accurate manner to improve usage efficiency.

(3) Interface layout

In PyQt, you can use the layout manager to implement the interface layout. When choosing the layout method, you need to consider the overall structure of the interface and the requirements of content display to ensure that the layout is reasonable and beautiful.

(4) Control selection

In PyQt, controls are the basic elements that make up the user interface, such as buttons, text boxes, list boxes, etc. When selecting a control, it is necessary to fully consider its function and coordination with other controls.

(5) Event handling

In PyQt, events refer to various actions or signals generated when users interact with the interface, such as clicking buttons, entering text, etc. In order to respond to these events and trigger corresponding actions or functions, it is necessary to add event handlers to the control, so that it can handle logic, and then realize dynamic interaction with users.

(6) Interface beautification

In PyQt, you can beautify the interface by adjusting the style, color, font and other properties of the control. When beautifying the interface, it is necessary to keep the consistency and coordination of the overall style,

(7) Testing and optimization

After the design of the user interface is completed, it needs to be tested and optimized. Ensure that all functions in the interface can work normally as expected, and ensure the response speed of the interface. According to the test results, find and fix potential problems in time, and make targeted optimization and improvement.

5.1. 2 User interaction interface design

User the interactive interface will be designed as a main window and two child windows(Conductor Parameters sub-window and Tower Data sub-window), and through PyQt and Qt Designer, the design of interactive interface is realized together.

① Qt Designer

Qt Designer is a visual design tool that is part of the Qt framework and is dedicated to designing and creating user interfaces. Developers can lay out the application interface by dragging components, setting properties, etc. without writing a lot of code, and the designed interface is saved as a. ui file.

② PyQt

PyQt is a Python binding of the Qt library. It combines the concise ease of use of the Python language with the power of the Qt framework, allowing developers to use the Python language and Qt library to create cross-platform desktop applications. PyQt uses the pyuic5 command to convert. ui files created by Qt Designer into Python code.

PyQt provides a rich set of APIs for interacting with various components in the interface and implementing the functionality of the application, so developers can quickly load and manipulate the interface created by Qt Designer.

The design of this paper controls such as QWidget, QPushButton, QLabel, QLineEdit and QComboBox are mainly used.

Eventually, the main window, wire parameters child window and tower data sub-window of design interface respectively as shown in Figure 5.1, Figure 5.2, Figure 5.3Shown in.

Figure 5.1vMain window design interface

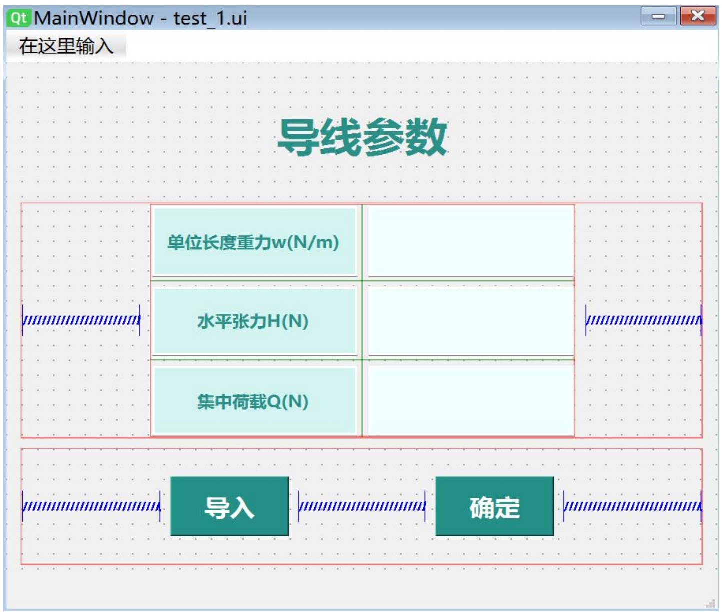

Figure 5.2 Wire parameters child window design interface

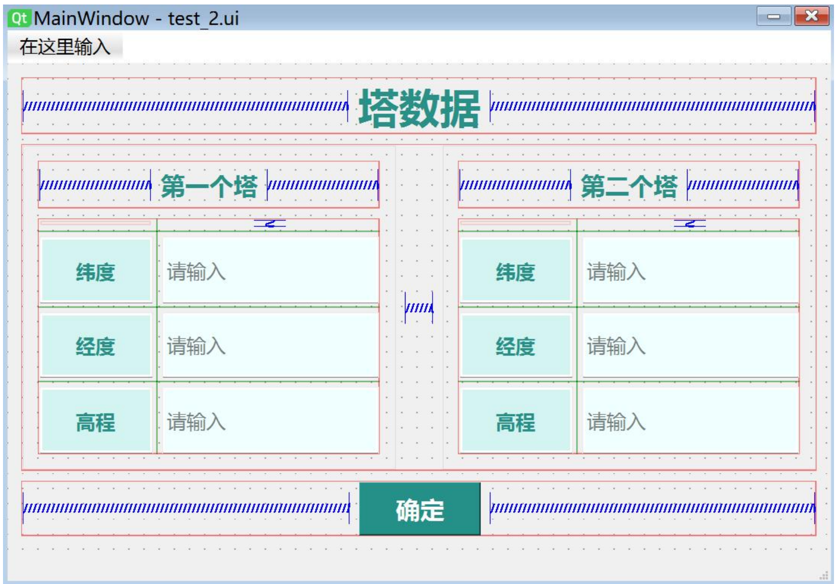

Figure 5.3 Tower DataChild windowDesignInterface

5.1. 3 User Interactive Interface Operation

In the PyCharm virtual environment, the interface files designed by Qt Designer are passed throughpyuic5 converts to Python code, and runs。

(1) Main window interface

Figure 5.4 shows the interface when entering the main window. The main window is divided into four areas, namely, parameter input area, result display area, key area and animation area.

In the button area, click "Import Wire Parameters”And“Setting tower parameters", which can realize the import of transmission line parameters and the input of information of two adjacent transmission towers. The imported and input data are displayed in the parameter input area, such as the parameters of the conductor, the latitude and longitude, elevation and other information of the transmission tower. The "Start" and "Stop" in the key area can control the operation of the program. While the program is running, the result display area will update the calculation results in real time, and the animation area can simulate application of Line Mobile Robot on High voltage transmission line movement process.

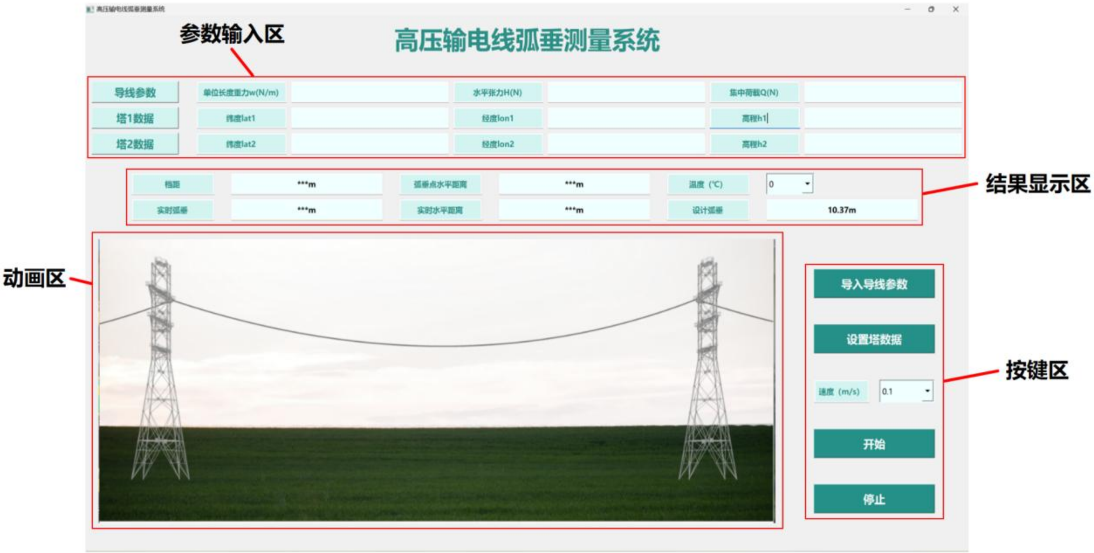

Figure 5.4 Main window Interface

(2) The wire parameter sub-window interface runs

Click the "Import Wire Parameters" button to jump to the Wire Parameters sub-window, as shown in Figure 5.5.

Figure 5.5 Wire Parameters Sub-Window Interface

According to Chapter 3 of the sag algorithm shows that the measurement high voltage transmission the parameters required for line sag value are: self-gravity per unit length of conductor ω, the dead weight of the line mobile robot Q wire horizontal tension H a horizontal distance of the first transmission tower and the second transmission tower l and elevation difference h. Among them, the parameters ω、Q

and H are all known parameters, given by"Sag Robot Configuration Parameter Presentation Field Table"Available,Specific data see Table 5.1.

Table 5.1 Sag Robot Configuration Parameters Presentation Field Table

Name	Unit	Numerical value
Calculated section	mm^2	674
Outer diameter	mm	33.8
Gravity per unit length	N/m	20.36832
Elastic coefficient	N/mm^2	63000
Coefficient of expansion	1/°C	0.0000209
Breaking force	N	150450×0.95
Maximum Use Tension	N	57072
Average running tension	N	35670
Maximum safety factor		2.5
Average/Breaking Force	%	25
Concentrated load	N	50
Horizontal tension	N	36647.5

Click the "Import" button in the Wire Parameters sub-window to import the local Excel file. Select and open the "Sag Robot Configuration Parameters Presentation Field Table", as shown in Figure 5.6.

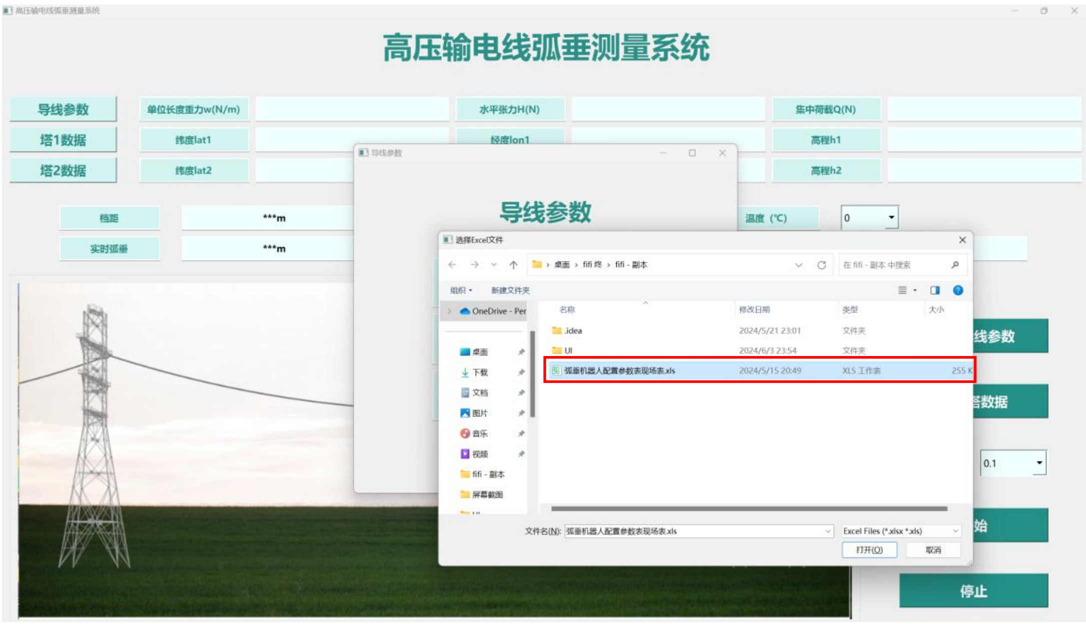

Figure 5.6 Import parameter table

The program automatically recognize that gravity per unit length in the table ω

horizontal tension H and concentrated loads Q is oparametric. The specific values are displayed on the sub-window interface, as shown in Figure 5.7.

Figure 5.7 Import Wire Parameters

(3) The tower data sub-window interface runs

Click the "Set Tower Data" button to jump to the Tower Data sub-window, as shown in Figure 5.8.

Figure 5.8 Tower DataChild window interface

In the tower data sub-window, you can input the data of the first transmission tower and the second transmission tower measured by RTK, such as latitude, longitude and elevation, as shown in Figure 5.9. The unit of latitude and longitude is degrees, and the unit of elevation is meters.

Figure 5.9 Enter tower data

(4) Main window display

After importing wire parameters and setting tower parameters, the parameter input area in the main window will update the parameter values, as shown in Figure 5.10.

Figure 5.10 Parameter display

In addition, according to the actual situation, the running speed of the line mobile robot can be selected through the drop-down box, and the speed range is set to0.1 m/s-1m/s, as shown in Figure 5.11.

Figure 5.11 Operating speed selection

At the same time,Can still According to the actual situation, through the drop-down box select current temperature, different temperatures correspond to different design sag values, as shown in Figure 5.12.

Figure 5.12 Temperature selection and design sag display

Subsequently, the measured maximum sag value will be compared with the design sag value, so as to judge whether the high-voltage transmission line should be tight or loose.Design sag value see Table 5.2， its middle， Two high-voltage transmission towers of the gear pitch is 415m and the elevation difference is 4.2 m, the temperature range is 0-40℃。

Table 5.2 LeadsDesignSag table

Temperature	Sag value (m)		Temperature	Sag value (m)	
0	10.365	10.37	21	11.398	11.40
1	10.415	10.42	22	11.446	11.45
2	10.465	10.47	23	11.494	11.50
3	10.514	10.52	24	11.542	11.54
4	10.564	10.56	25	11.590	11.59
5	10.614	10.61	26	11.638	11.64
6	10.664	10.66	27	11.686	11.69
7	10.713	10.71	28	11.734	11.74
8	10.763	10.76	29	11.782	11.78
9	10.813	10.81	30	11.830	11.83
10	10.863	10.86	31	11.878	11.88
11	10.911	10.91	32	11.926	11.93
12	10.960	10.96	33	11.973	11.97
13	11.009	11.01	34	12.021	12.02
14	11.058	11.06	35	12.069	12.07
15	11.106	11.11	36	12.117	12.12
16	11.155	11.16	37	12.164	12.17
17	11.204	11.20	38	12.212	12.21
18	11.253	11.25	39	12.260	12.26
19	11.301	11.30	40	12.308	12.31
20	11.350	11.35			

(5) The main window runs

Click the "Start" button and the program begins to run. According to the parameters in the parameter input area, the gear pitch and the horizontal distance of sag point are calculated and displayed in the result display area.

According to the running speed of the selected line mobile robot, the real-time sag and the real-time horizontal distance in the result display area are updated every second.

Press the "Stop" button, the program stops running, and the running speed of the line mobile robot can be changed. Click the "Start" button, and the program can continue running.

The calculation result display interface is shown in Figure 5.13.

Figure 5.13 The calculation results show

While the program is running, the running process of the line mobile robot is simulated in the animation area. As shown in Figure 5.14, the robot starts to run from the suspension point of the first transmission tower until it stops at the maximum sag point, and the running speed of the robot changes with the selection of running speed.

Figure 5.14 Animation display

When the line mobile robot runs to the maximum sag point, the system stops running and pops up a pop-up window to inform the user that it has run to the maximum sag point, and displays the maximum sag value at this time, as shown in Figure 5.15.

Figure 5.15 Sag maximum point pop-up reminder

After the whole program is running, click the close button in the upper right corner, and the system will pop up a pop-up window, allowing the user to choose whether to confirm the exit or not, as shown in Figure 5.16.

Figure 5.16 Interface exit pop-up reminder

5.2 Program implementation

Development of a High Voltage Transmission Line Using PyQt5RoadsSag measurement system. The code includes interface initialization, event binding of interface controls, timer setting and use, Excel data reading, geographic calculation (Vincenty algorithm), drawing dispose and other contents。

5.2. 1 Main window programming

（1) Temperature and velocity update

As shown in Figure 5.17As shown, from the corresponding Excel form read the temperature value in the file,Put it add to the drop-down box, and add the speed （0.1-1.0）Options to the drop-down box.

```
1 usage
def temperature_select(self):
    """初始化温度选择下拉框"""
    excel_file = 'C:/Users/lenovo/Desktop/fifi 终/fifi - 副本/UI/your_excel_file.xlsx'  # Excel文件路径
    workbook = load_workbook(excel_file)
    worksheet = workbook['111']  # 选择要读取的工作表
    for row in worksheet.iter_rows(min_row=2, max_col=1, values_only=True):
        self.comboBox.addItem(str(row[0]))  # 将Excel中的选项转换为字符串类型后添加到下拉框中

1 usage
def v_select(self):
    """初始化速度选择下拉框"""
    for value in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]:
        self.comboBox_2.addItem(str(value))
```

Figure 5.17 Temperature and speed selection

As shown in Figure 5.18As shown in, different temperatures can be selected through the drop-down box, and the contents in the corresponding text box can be updated in real time.

```python
    def updateValue(self, text):
        """根据选择项更新文本框内容"""
        excel_file = 'your_excel_file.xlsx'  # Excel文件路径
        workbook = load_workbook(excel_file)
        worksheet = workbook['111']  # 选择要读取的工作表

        row_num = self.comboBox.currentIndex() + 2  # 获取选择的选项对应的行号，需要+2是因为下拉框第一项对应的行号是3
        cell_value = worksheet.cell(row=row_num, column=2).value  # 读取对应行的第2列数值
        formatted_f_x = "{}m".format(cell_value)
        self.lineEdit_biaozhunzhi.setText(str(formatted_f_x))
```

Figure 5.18 Update text box contents

（2) point with animation

As shown in Figure 5.19As shown in, by drawing points on the initial picture and updating the position of the current points at certain intervals, the route of the line mobile robot moving on the high-voltage transmission line is simulated.

Add _ point: Adds points to the points list.

Next _ point: Update the current point index and call drawpoint to draw the point.

drawpoint: Draws the current point on the initial picture and updates the display.

```python
    def add_point(self, x, y):
        """添加点到列表中"""
        point = QPoint(x, y)  # 创建一个QPoint对象
        self.points.append(point)  # 在点列表中添加QPoint对象
    2 usages
    def next_point(self):
        """绘制下一个点"""
        self.current_point = (self.current_point + 1) % len(self.points)  # 循环到下一个点
        print(self.current_point)
        self.drawpoint()
        self.update()
        self.repaint()
    1 usage
    def drawpoint(self):
        """绘制当前点"""
        pixmap = QPixmap(self.img_path)
        painter = QPainter(pixmap)
        painter.setRenderHint(QPainter.Antialiasing)
        pen = QPen(Qt.black)
        pen.setWidth(8)
        painter.setPen(pen)
        current_point_index = self.current_point % len(self.points)
        pen.setColor(Qt.red)
        painter.setPen(pen)
        pen.setWidth(20)
        point = self.points[current_point_index]  # 获取当前点位置
        diameter = 8  # 设置绘制圆点的直径
        painter.drawEllipse(point, diameter // 2, diameter // 2)  # 绘制圆点
```

Figure 5.19 Draw Points and Animations

（3) Button Event Handling and Timer Control

As shown in Figure 5.20As shown in, click the start button to start two timers, and click the stop button to stop the timer. Wherein, the timer 1 controls the display of the real-time sag value and the running horizontal distance in the main window, and is updated every 1000ms; Timer 2 controls the display of animation on the initial picture, and simulates that the line mobile robot updates its position every interval according to the parameters such as running distance, running speed and the number of drawn points.

```python
    def start_pushButton(self):
        """开始测量并进行计时，计算相关参数"""
        self.timer.timeout.connect(self.updateRunningTime)
        self.timer2.timeout.connect(self.next_point)
        select_text = self.comboBox_2.currentText()

        lat1 = float(self.lineEdit_latitude1.text())
        lon1 = float(self.lineEdit_longitude1.text())
        lat2 = float(self.lineEdit_latitude2.text())
        lon2 = float(self.lineEdit_longitude2.text())
        l = self.vincenty_distance(lat1, lon1, lat2, lon2)
        # w为导线单位长度重量，H为导线张力，Q为集中荷载，x为机器人运动的横向距离，h为两挂点的高程差（h = h2 - h1），l为档距。
        w = float(self.lineEdit_weightPerUnitLength.text())
        H = float(self.lineEdit_runningTension.text())
        Q = float(self.lineEdit_concentratedLoad.text())
        x = float(self.x)
        h = float(self.lineEdit_height2.text()) - float(self.lineEdit_height1.text())
        la = l / 2 - H / w * (math.asinh(w * h / (2 * H * (math.sinh(w * l * 0.5 / H)))))
        l1 = la + H / w * math.asinh(h / l)

        self.timer.start(1000)
        self.timer2.start(l1 / (len(self.points)-1) / float(select_text) * 1000)
```

```python
    def stop_pushButton(self):
        """停止计时和测量"""
        self.timer.stop()
        self.timer2.stop()
        self.timer.timeout.disconnect(self.updateRunningTime)
        self.timer2.timeout.disconnect(self.next_point)
```

Figure 5.20 Button Event Handling and Timer Control

(4) Data calculation and display

As shown in Figure 5.21As shown, by reading the text in the box content, and then based on sag algorithms, that is realize the calculation and display of gear pitch, real-time sag, real-time horizontal distance, sag point horizontal distance and other results.

```python
    def updateRunningTime(self):
        """实时更新运行时间和相关参数"""
        select_text = self.comboBox_2.currentText()
        lat1 = float(self.lineEdit_latitude1.text())
        lon1 = float(self.lineEdit_longitude1.text())
        lat2 = float(self.lineEdit_latitude2.text())
        lon2 = float(self.lineEdit_longitude2.text())
        l = self.vincenty_distance(lat1, lon1, lat2, lon2)
        # w为导线单位长度重量，H为导线张力，Q为集中荷载，x为机器人运动的横向距离，h为两挂点的高程差（h = h2 - h1），l为档距。
        w = float(self.lineEdit_weightPerUnitLength.text())
        H = float(self.lineEdit_runningTension.text())
        Q = float(self.lineEdit_concentratedLoad.text())
        x = float(self.x)
        h = float(self.lineEdit_height2.text()) - float(self.lineEdit_height1.text())
        la = l / 2 - H / w * (math.asinh(w * h / (2 * H * (math.sinh(w * l * 0.5 / H)))))
        l1 = la + H / w * math.asinh(h / l)

        if self.x < l1:
            formatted_time = "{}m".format(self.x)  #实时水平距离
            self.x += float(select_text)
            self.lineEdit_running.setText(formatted_time)
            f_x = self.calculate_fx(w, H, Q, x, h, l)
            formatted_f_x = "{}m".format(f_x)  #实时弧垂
            self.lineEdit_realTimeSagValue.setText(formatted_f_x)
            formatted_f_shui = "{}m".format(l1)  #弧垂点水平距离
            self.lineEdit_shuiping.setText(formatted_f_shui)
            formatted_f_l = "{}m".format(l)  # 档距
```

Figure 5.21 Data calculation and display

（5) pop-up window display

As shown in Figure 5.22As shown in, when the line mobile robot runs to the maximum sag point, a pop-up window will pop up in the window, prompting the user that "it has run to the maximum sag point", and the current sag value will be displayed synchronously.

```python
# 弹窗提示
f_x = self.calculate_fx(w, H, Q, x - float(select_text), h, l)  # 再次计算可用最新的f_x值
msg_box = QMessageBox()
msg_box.setWindowTitle("提示")
msg_box.setText("已运行到弧垂最大点！\n当前弧垂值为{:.3f}m".format(f_x))
msg_box.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)
# 设置按钮的中文文本
msg_box.button(QMessageBox.Ok).setText("确认")
msg_box.button(QMessageBox.Cancel).setText("取消")
# 显示弹窗并获取用户的响应
response = msg_box.exec_()
if response == QMessageBox.Ok:
    print("用户点击了确认按钮")
else:
    print("用户点击了取消按钮")
```

Figure 5.22 Pop-up window Hint

As shown in Figure 5.23As shown in, a close event is defined to handle the window closing event, and a dialog box with the title "Confirm Exit" and the content "Are you sure you want to exit?" Pops up, providing two button options of "Yes" and "No" to let the user choose whether to close the window or not. If Yes is selected, the closing event is accepted and the window closes; If you select No, the closing event is ignored and the window does not close.

```python
def closeEvent(self, event):
    """窗口关闭事件，弹出确认对话框"""
    # 定义消息框的标题和内容
    title = '确认退出'
    message = '    您确定要退出吗？ '
    # 创建自定义的消息框
    reply = QMessageBox()
    reply.setWindowTitle(title)
    reply.setText(message)
    # 添加自定义按钮
    yes_button = reply.addButton('是', QMessageBox.YesRole)
    no_button = reply.addButton('否', QMessageBox.NoRole)
    # 弹出确认对话框
    reply.exec_()
    # 根据用户的选择决定是否接受关闭事件
    if reply.clickedButton() == yes_button:
        event.accept()  # 接受关闭事件
    else:
        event.ignore()  # 忽略关闭事件
```

Figure 5.23 Window Close

5.2. 2 Wire parameter sub-window program design

（1) Confirm button slot function implementation

As shown in Figure 5.24As shown, by clicking“Confirm”Button, emits the values signal signal, passing three parameter values. If the operation is successful, the Success reg signal is issued, and if an error occurs, the exception information, the file name of the error, and the line number are output.

```python
    def confirm_pushButton(self):
        try:
            # 发出信号，将读取的值传递出去
            self.ValuesSignal.emit(self.unit_length_weight, self.average_tension, self.concentrated_load)
            self.SuccessReg.emit()
        except Exception as e:
            # 打印异常信息及错误文件和行数
            print("---串口线程异常--： ", str(e))
            print(f'error file: {e.__traceback__.tb_frame.f_globals["__file__"]}')
            print(f'error line:{e.__traceback__.tb_lineno}')
```

Figure 5.24 Confirm Button Slot Function Implementation

（2) Import button slot function implementation

As shown in Figure 5.25As shown, by clicking“To lead”Button, opens the dialog box of file selection, the user can select an Excel file and read the contents of that Excel file with pandas. Confirm the column names you want to look for (gravity per length, horizontal tension, concentrated load) and look for these names in the Excel file. After finding the matching name, get its corresponding numeric value, assign it to the variable, and display the read value in the corresponding text box.

```python
def import_pushButton(self):
    try:
        # 打开文件选择对话框
        file_name, _ = QFileDialog.getOpenFileName(self, "选择Excel文件", "../", "Excel Files (*.xlsx *.xls)")
        if file_name:
            # 使用pandas读取Excel文件
            df = pd.read_excel(file_name)
            names_to_find = ['单位长度重力', '水平张力', '集中荷载']
            values = {}
            first_column_index = 0  # '630导线参数'是第一列，其索引为0
            # 遍历第一列（'630导线参数'）以找到匹配的名称
            for index, name in df.iloc[:, first_column_index].items():
                if name in names_to_find:
                    values[name] = df.iloc[index, 2]  # 从第三列获取对应的值
            # 提取单位长度重量、平均运行张力和集中荷载的值
            self.unit_length_weight = values.get('单位长度重力')
            self.average_tension = values.get('水平张力')
            self.concentrated_load = values.get('集中荷载')
            # 设置文本框的值
            self.lineEdit_weightPerUnitLength.setText(str(self.unit_length_weight))
            self.lineEdit_runningTension.setText(str(self.average_tension))
            self.lineEdit_concentratedLoad.setText(str(self.concentrated_load))
    except Exception as e:
        # 打印异常信息及错误文件和行数
        print("---串口线程异常--：", str(e))
        print(f'error file: {e.__traceback__.tb_frame.f_globals["__file__"]}')
        print(f'error line:{e.__traceback__.tb_lineno}')
```

Figure 5.25 Import button slot function implementation

5.2. 3 Tower data sub-window programming

As shown in Figure 5.26As shown, the input text is obtained from six text boxes (lineEdit _ latitude1, lineEdit _ longitude1, lineEdit _ height1, lineEdit _ latitude2, lineEdit _ longitude2, and lineEdit _ height2) by clicking the "Confirm" button. Issues the values signal signal, which contains six string parameters (data fetched from the text box). If the operation is successful, a success reg signal is emitted, and if an error occurs, a signal that an exception has occurred is output.

```python
    def confirm_pushButton(self):
        try:
            # 从文本输入框中获取值
            self.latitude1 = self.lineEdit_latitude1.text()
            self.longitude1 = self.lineEdit_longitude1.text()
            self.height1 = self.lineEdit_height1.text()
            self.latitude2 = self.lineEdit_latitude2.text()
            self.longitude2 = self.lineEdit_longitude2.text()
            self.height2 = self.lineEdit_height2.text()
            # 发出信号，将获取的值传递出去
            self.ValuesSignal.emit(self.latitude1, self.longitude1, self.height1,
                                   self.latitude2, self.longitude2, self.height2)
            # 发出注册成功信号
            self.SuccessReg.emit()
        except Exception as e:
            # 打印异常信息及错误文件和行数
            print("---串口线程异常--：", str(e))
            print(f'error file: {e.__traceback__.tb_frame.f_globals["__file__"]}')
            print(f'error line:{e.__traceback__.tb_lineno}')
```

Figure 5.26 Button slot function implementation

5.3 Summary

By designing user interface and programming, this chapter realizes the real-time display of the design sag value, the actual sag value and the horizontal position of the line mobile robot of high-voltage transmission line, so that users can simply and clearly observe it.

When the robot moves to the maximum sag point, it stops moving immediately. By comparing the measured sag value and the design value on the interface, the current state of the high-voltage transmission line can be easily judged, which is helpful for subsequent tightening or loosening of the transmission line to ensure line safety.

The main window code is shown in Appendix 2,Wire parameters see Appendix 3 for the sub-window code, tower data see Appendix 4 for the sub-window codes.

References

[1] Jiao Chengyi. Observation of sag (relaxation) of overhead lines of transmission lines [J]. Science and Technology Information, 2012 (27): 389-390 +416.

[2] Xu Qingsong, Ji Hongxian, Wang Menglong. Real-time monitoring of transmission line sag [J]. High Voltage Technology, 2007 (07): 206-209.

[3] Cloet, E. & Lilien, J.: Uprating transmission lines through the use of an innovative real-time monitoring system. 2011 IEEE PES 12th International Conference on Transmission and Distribution Construction, Operation and Live-Line Maintenance (ESMO), Providence, RI, 1-6 (2011)

[4] Zhimin, H., Yadong, L.: The field application analysis of dynamic line rating system based on tension monitoring. 2011 IEEE Power Engineering and Automation Conference, Wuhan, 284-288 (2011)

[5] Nazare, F.V.B.d., Werneck, M.M.: Hybrid optoelectronic sensor for current and temperature monitoring in overhead transmission lines. IEEE Sens. J. 12 (5), 1193-1194 (2012)

[6] Hsu, T.S., et al.: An IoT-based sag monitoring system for overhead transmission lines. IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia), Bangkok, Thailand, 515-519 (2019)

[7] Tong Weiguo, Li Baoshu, Yuan Jinsha, et al. Transmission line sag measurement method based on aerial sequence images [J]. Chinese Journal of Electrical Engineering, 2011, 31 (16): 115-120.

[8] Mahajan, S.M., Singareddy, U.M.: A real-time conductor sag measurement system using a differential GPS. IEEE Trans. Power Delivery 27 (2), 475480 (2012)

[9] Mo Zhiyue, Qin Liwen, Ye Lei, et al. Research on sag calculation method using image recognition technology [J]. Science and Technology Vision, 2015 (23): 18-19 +276.

[10] Ma Weifeng, Wang Jinliang, Wang Cheng, et al. Airborne lidar point cloud data transmission line sag simulation [J]. Journal of Surveying and Mapping Science and Technology, 2019, 36 (04): 394-399 +405.

[11] Xu, Q., et al.: Magnetic-field-sensing-based approach for current reconstruction, sag detection, and inclination detection for overhead transmission system. IEEE Trans. Magn. 55(7), 1–7 (2019)

[12] Hao Bin, Cai Yuan, Zhu Chunmao.Research on a method based on RTK UAV photogrammetry wire sag [J]. Northeast Electric Power Technology, 2022, 43 (04): 28-31.

[13] Qian Jianguo, Song Jiang, Zhou Jiahui, et al. Research on transmission line sag measurement

by UAV aerial photography [J]. Surveying and Mapping Science, 2023, 48 (06): 189-197 +238.
[14] Abraham, A.P., Ashok, S.: Gyro-accelerometric SAG analysis and online-monitoring of transmission lines using line recon robot. 2012 Annual IEEE India Conference (INDICON), Kochi, 1036-1040 (2012)
[15] ZENGIN A T, ERDEMIR G, AKINCI T C, et al., 2020. Measurement of Power Line Sagging Using Sensor Data of a Power Line Inspection Robot [J/OL]. IEEE Access, 8: 99198-99204.
[16] Li Kai, Liu Yunfei, Han Qiyun, et al. Research on Beidou positioning sag observation of overhead transmission lines [J]. Science and Technology Innovation and Application, 2022, 12 (10): 67-70.
[17] Han Qiyun, Jiao Jian, Wang Junlong, et al. Construction method and application of transmission line sag observation based on satellite positioning technology [J]. Electrical Technology, 2022 (16): 123-126.
[18] Cheng Yansheng, Yang Hemeng, Chen Yanfang. Design of sag monitoring system for overhead transmission lines based on Beidou [J]. Automation Technology and Application, 2023, 42 (02): 98-100 +154.
[19] Wang Jiemin, Wang Wenyi, Liang Huiyong, et al. Design and application of combined positioning system for power line sag measurement [J]. Hebei Electric Power Technology, 2023, 42 (04): 73-78.
[20] Zhao Jiufeng. Application of catenary theory in the design and calculation of zip line with large chord inclination angle [J]. Journal of Xian Institute of Aeronautics and Aeronautics, 2023, 41 (01): 69-73.

Appendix 1 Vincenty Formula Code

```
from math import atan, atan2, cos, sin, sqrt, tan, radians

Def vincenty _ distance (lat1, lon1, lat2, lon2, a = 6378137.0, b = 6356752.314245, f
= 1/298. 257223563):

"""
Calculate the distance between two points on the earth using the Vincenty formula.
: lat1: the latitude of the first point in degrees
: lon1: The longitude of the first point in degrees
    : lat2: the latitude of the second point in degrees
: lon2: The longitude of the second point in degrees
: a: Major half axis of the Earth (WGS-84 standard is used by default)
: b: Earths minor half axis (WGS-84 standard is used by default)
: f: Oblateness of the Earth (WGS-84 standard is used by default)
: return: The distance between two points in meters
    """

# Convert degrees to radians
lat1, lon1, lat2, lon2 = map (radians, [lat1, lon1, lat2, lon2])

# Difference in longitude
    L = lon2-lon1

# naturalized latitude
U1 = atan ((1-f) * tan (lat1))
U2 = atan ((1-f) * tan (lat2))

# iteration variables
sinU1, cosU1 = sin (U1), cos (U1)
sinU2, cosU2 = sin (U2), cos (U2)

Lambda = L
```

```
# Perform iterative calculations
for iteration in range (1000):
sinLambda = sin (Lambda)
cosLambda = cos (Lambda)
sinSigma = sqrt ((cosU2 * sinLambda) * * 2 + (cosU1 * sinU2-sinU1 * cosU2 *
cosLambda) * * 2)
if sinSigma = = 0:
return 0 # If two points coincide, the distance is 0
cosSigma = sinU1 * sinU2 + cosU1 * cosU2 * cosLambda
sigma = atan2 (sinSigma, cosSigma)
sinAlpha = cosU1 * cosU2 * sinLambda/sinSigma
cos2Alpha = 1-sinAlpha * * 2
if cos2Alpha = = 0:
        cos2SigmaM = 0 # on the equator line
else:
cos2SigmaM = cosSigma-2 * sinU1 * sinU2/cos2Alpha
C = f/16 * cos2Alpha * (4 + f * (4-3 * cos2Alpha))
LambdaPrev = Lambda
Lambda = L + (1-C) * f * sinAlpha * (sigma + C * sinSigma * (cos2SigmaM + C *
cosSigma * (-1 + 2 * cos2SigmaM * * 2)))

# Check convergence
if abs (Lambda-LambdaPrev) < 1e-12:
break
else:
# If the iteration does not converge
return None

# Calculate u2, A and B parameters
u2 = cos2Alpha * (a * * 2-b * * 2)/(b * * 2)
A = 1 + u2/16384 * (4096 + u2 * (-768 + u2 * (320-175 * u2)))
B = u2/1024 * (256 + u2 * (-128 + u2 * (74-47 * u2)))

# compute deltaSigma
```

deltaSigma = B * sinSigma * (cos2SigmaM + B/4 * (cosSigma * (-1 + 2 * cos2SigmaM * * 2)-B/6 * cos2SigmaM * (-3 + 4 * sinSigma * * 2) * (-3 + 4 * cos2SigmaM * * 2))

Calculate final distance

s = b * A * (sigma-deltaSigma)

return s

Appendix 2 Main Window Code

```
import math
import time

From PyQt5. QtCore import QTimer, QDateTime, Qt, QPoint
From PyQt5. QtGui import QPixmap, QPen, QPainter, QBrush, QColor
From PyQt5. QtWidgets import QMessageBox, QApplication, QFileDialog,
QTableWidgetItem, QHeaderView
From Ui _ test _ 1 import * # Import the user interface of Ui _ test _ 1
From Ui _ test _ 2 import * # Import the user interface of Ui _ test _ 2
From project _ test import Ui _ MainWindow # Import the main window interface
from math import atan, atan2, cos, sin, sqrt, tan, radians, pi
From openpyxl import load _ workbook # Import libraries that read Excel files


class MainWindow (QtWidgets.QMainWindow, Ui _ MainWindow):

Def _ _ init _ _ (self, parent = None):
super (MainWindow, self). _ _ init _ _ (parent)
self.setupUi (self) # Set UI interface
self.showMaximized () # Maximize the display window
self.towerData = towerData _ window () # Create a tower data window instance
self.wireParameters = wireParameters _ window () # Create a wire parameter window
instance
self.points = [] # List of coordinates of storage points
# Add sample points that are used to draw the graph
Self.add _ example _ points ()
Self.current _ point = 0 # Current point index to be drawn
self.init () # initialize
Self.img _ path = "C:/Users/lenovo/Desktop/fifi terminal/fifi-copy/high voltage
line.png" # picture path
        Self.label _ image.setPixmap (QPixmap (self.img _ path)) # Set pictures to
tags
```

```
Self.temperature _ select () # Initialize the temperature selection drop-down box
Self.v _ select () # initialize speed selection drop-down box

Def add _ example _ points (self):
"" "Add sample point" ""
Self.add _ point (143, 110)
Self.add _ point (148, 112)
Self.add _ point (153, 114)
Self.add _ point (158, 115.5)
Self.add _ point (163, 117)
Self.add _ point (168, 118.5)
Self.add _ point (173, 120)
Self.add _ point (178, 121.5)
Self.add _ point (183, 123)
Self.add _ point (188, 124.25)
Self.add _ point (193, 125.5)
Self.add _ point (198, 126.75)
Self.add _ point (203, 128)
Self.add _ point (208, 129.25)
Self.add _ point (213, 130.5)
Self.add _ point (218, 131.75)
Self.add _ point (223, 133)
Self.add _ point (228, 134.25)
Self.add _ point (233, 135.5)
Self.add _ point (238, 136.75)
Self.add _ point (243, 138)
Self.add _ point (248, 139.25)
Self.add _ point (253, 140.5)
Self.add _ point (258, 141.75)
Self.add _ point (263, 143)
Self.add _ point (268, 144.25)
          Self.add _ point (273, 145.5)
Self.add _ point (278, 146.75)
Self.add _ point (283, 148)
```

Self.add _ point (288, 149)
Self.add _ point (293, 150)
Self.add _ point (298, 151)
Self.add _ point (303, 152)
Self.add _ point (308, 153)
Self.add _ point (313, 154)
Self.add _ point (318, 155)
Self.add _ point (323, 156)
Self.add _ point (328, 157)
Self.add _ point (333, 158)
Self.add _ point (338, 159)
Self.add _ point (343, 160)
Self.add _ point (348, 161)
Self.add _ point (353, 162)
Self.add _ point (358, 163)
Self.add _ point (363, 164)
Self.add _ point (368, 164.75)
Self.add _ point (373, 165.5)
Self.add _ point (378, 166.25)
Self.add _ point (383, 167)
Self.add _ point (388, 167.75)
Self.add _ point (393, 168.5)
Self.add _ point (398, 169.25)
Self.add _ point (403, 170)
Self.add _ point (408, 170.75)
Self.add _ point (413, 171.5)
Self.add _ point (418, 172.25)
Self.add _ point (423, 173)
Self.add _ point (428, 173.75)
Self.add _ point (433, 174.5)
Self.add _ point (438, 175.25)
Self.add _ point (443, 176)
Self.add _ point (448, 176.5)
Self.add _ point (453, 177)

```
Self.add _ point (458, 177.5)
Self.add _ point (463, 178)
Self.add _ point (468, 178.5)
Self.add _ point (473, 179)
Self.add _ point (478, 179.5)
Self.add _ point (483, 180)
Self.add _ point (488, 180.5)
Self.add _ point (493, 181)
Self.add _ point (498, 181.5)
Self.add _ point (503, 182)
Self.add _ point (508, 182.5)
Self.add _ point (513, 183)
Self.add _ point (518, 183.5)
Self.add _ point (523, 184)
Self.add _ point (528, 184.5)
Self.add _ point (533, 185)
Self.add _ point (538, 185.5)
Self.add _ point (543, 186)
Self.add _ point (548, 186.25)
Self.add _ point (553, 186.5)
Self.add _ point (558, 186.75)
Self.add _ point (563, 187)
Self.add _ point (568, 187.25)
Self.add _ point (573, 187.5)
Self.add _ point (578, 187.75)
Self.add _ point (583, 188)
Self.add _ point (588, 188.25)
Self.add _ point (593, 188.5)
Self.add _ point (598, 188.75)
Self.add _ point (603, 189)
Self.add _ point (608, 189.25)
Self.add _ point (613, 189.5)
        Self.add _ point (618, 189.75)
Self.add _ point (623, 189.85)
```

```
Self.add _ point (628, 189.90)
Self.add _ point (633, 189.95)
Self.add _ point (638, 190)


def init (self):
"" "Initialize various parameters and binding events" ""
self.timer = QTimer ()
self.timer2 = QTimer ()
self.x = 0
Self.pushButton _ wireParameters.clicked.connect (self.wireParameters _
pushButton)
Self.pushButton _ setTowerData.clicked.connect (self.towerData _ pushButton)
Self.pushButton _ start.clicked.connect (self.start _ pushButton)
Self.pushButton _ stop.clicked.connect (self.stop _ pushButton)
self.wireParameters.SuccessReg.connect (self.success _ wireParameters) # Register or
cancel to jump back
self.wireParameters.ValuesSignal.connect (self.valuesReceived _ wireParameters)

self.towerData.SuccessReg.connect (self.success _ towerData) # Register or cancel to
jump back
self.towerData.ValuesSignal.connect (self.valuesReceived _ towerData)
self.comboBox.currentTextChanged.connect (self.updateValue)

Def valuesReceived _ wireParameters (self, unit _ length _ weight, average _ tension,
concentrated _ load):
        "" "Update Wire Parameters" ""
Self.lineEdit _ weightPerUnitLength.setText (str (unit _ length _ weight))
Self.lineEdit _ runningTension.setText (str (average _ tension))
Self.lineEdit _ concentratedLoad.setText (str (concentrated _ load))

Def valuesReceived _ towerData (self, latitude1, longitude1, height1, latitude2,
longitude2, height2):
"" "Update the data information of the tower" ""
```

Self.lineEdit _ latitude1. setText (latitude1)
Self.lineEdit _ longitude1. setText (longitude1)
Self.lineEdit _ height1. setText (height1)
Self.lineEdit _ latitude2. setText (latitude2)
Self.lineEdit _ longitude2. setText (longitude2)
Self.lineEdit _ height2. setText (height2)

Def temperature _ select (self):
"" "Initialization Temperature Selection Drop-down Box" ""
Excel _ file = C:/Users/lenovo/Desktop/fifi final/fifi-copy/UI/your _ Excel _ file.xlsx
# Excel file path
workbook = load _ workbook (excel _ file)
worksheet = workbook [111] # Select the worksheet to read
For row in worksheet.iter _ rows (min _ row = 2, max _ col = 1, values _ only = True):
self.comboBox.addItem (str (row [0])) # Add options in Excel to drop-down boxes after converting them to string types

Def v _ select (self):
"" "Initialization Speed Selection Drop-down Box" ""
for value in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]:
Self.comboBox _ 2. addItem (str (value))

def updateValue (self, text):
"" "Update text box contents according to selection" ""
Excel _ file = your _ Excel _ file.xlsx # Excel File Path
workbook = load _ workbook (excel _ file)
worksheet = workbook [111] # Select the worksheet to read

Row _ num = self.comboBox.currentIndex () + 2 # Get the row number corresponding to the selected option. +2 is required because the row number corresponding to the first item of the drop-down box is 3

```
Cell _ value = worksheet.cell (row = row _ num, column = 2). value # Reads the second column value of the corresponding row
Formatted _ f _ x = "{} m". format (cell _ value)
Self.lineEdit _ biaozhunzhi.setText (str (formatted _ f _ x))

Def add _ point (self, x, y):
"" "Add point to list" ""
point = QPoint (x, y) # Create a QPoint object
self.points.append (point) # Add a QPoint object to the point list

Def next _ point (self):
"" "Draw the next point" ""
Self.current _ point = (self.current _ point + 1)% len (self.points) # Loop to the next point
print (self.current _ point)
        self.drawpoint ()
self.update ()
self.repaint ()

def drawpoint (self):
"" "Draw Current Point" ""
pixmap = QPixmap (self.img _ path)
painter = QPainter (pixmap)
painter.setRenderHint (QPainter.Antialiasing)
pen = QPen (Qt.black)
pen.setWidth (8)
painter.setPen (pen)
Current _ point _ index = self.current _ point% len (self.points)
pen.setColor (Qt.red)
painter.setPen (pen)
pen.setWidth (20)
# Get the current point position
point = self.points [current _ point _ index]
# Set the diameter of the drawn dot
```

```
diameter = 8
# Draw dots
painter.drawEllipse (point, diameter//2, diameter//2)
print (point)
painter.end ()
Self.label _ image.setPixmap (pixmap)

Def wireParameters _ pushButton (self):
"" "Display Wire Parameters Window" ""
self.wireParameters.setWindowTitle ("Wire Parameters")
self.wireParameters.show ()

Def towerData _ pushButton (self):
"" "Display Tower Data Window" ""
self.towerData.setWindowTitle ("tower data")
self.towerData.show ()

Def success _ towerData (self):
"" "Close Tower Data Window" ""
        self.towerData.close ()

Def success _ wireParameters (self):
"" "Close Wire Parameters Window" ""
self.wireParameters.close ()

Def start _ pushButton (self):
"" "Start measurement and timing, calculate relevant parameters" ""
self.timer.timeout.connect (self.updateRunningTime)
Self.timer2. timeout.connect (self.next _ point)
Select _ text = self.comboBox _ 2. currentText ()

lat1 = float (self.lineEdit _ latitude1. text ())
lon1 = float (self.lineEdit _ longitude1. text ())
lat2 = float (self.lineEdit _ latitude2. text ())
```

lon2 = float (self.lineEdit _ longitude2. text ())
l = self.vincenty _ distance (lat1, lon1, lat2, lon2)
# w is the weight per unit length of the conductor, h is the tension of the conductor, Q is the concentrated load, x is the lateral distance of the robot movement, h is the elevation difference between the two hanging points (h = h2-h1), and l is the gear pitch.
w = float (self.lineEdit _ weightPerUnitLength.text ())
H = float (self.lineEdit _ runningTension.text ())
Q = float (self.lineEdit _ concentratedLoad.text ())
x = float (self.x)
h = float (self.lineEdit _ height2. text ())-float (self.lineEdit _ height1. text ())
la = l/2-H/w * (math.asinh (w * H/(2 * H * (math.sinh (w * l * 0.5/H)))))
l1 = la + h/w * math.asinh (h/l)

self.timer.start (1000)
Self.timer2. start (l1/(len (self.points)-1)/float (select _ text) * 1000)

def updateRunningTime (self):
"" "Real-time update of runtime and related parameters" ""
Select _ text = self.comboBox _ 2. currentText ()
lat1 = float (self.lineEdit _ latitude1. text ())
lon1 = float (self.lineEdit _ longitude1. text ())
lat2 = float (self.lineEdit _ latitude2. text ())
lon2 = float (self.lineEdit _ longitude2. text ())
l = self.vincenty _ distance (lat1, lon1, lat2, lon2)
# w is the weight per unit length of the conductor, h is the tension of the conductor, Q is the concentrated load, x is the lateral distance of the robot movement, h is the elevation difference between the two hanging points (h = h2-h1), and l is the gear pitch.
w = float (self.lineEdit _ weightPerUnitLength.text ())
H = float (self.lineEdit _ runningTension.text ())
Q = float (self.lineEdit _ concentratedLoad.text ())
x = float (self.x)

```
h = float (self.lineEdit _ height2. text ())-float (self.lineEdit _ height1. text ())
            la = l/2-H/w * (math.asinh (w * H/(2 * H * (math.sinh (w * l * 0.5/H)))))
l1 = la + h/w * math.asinh (h/l)

if self.x < l1:
Formatted _ time = "{} m". format (self.x) # real-time horizontal distance
self.x + = float (select _ text)
Self.lineEdit _ running.setText (formatted _ time)
F _ x = self.calculate _ fx (w, h, Q, x, h, l)
Formatted _ f _ x = "{} m". format (f _ x) # real-time sag
Self.lineEdit _ realTimeSagValue.setText (formatted _ f _ x)
Formatted _ f _ shui = "{} m". format (l1) # sag point horizontal distance
Self.lineEdit _ shuiping.setText (formatted _ f _ shui)
Formatted _ f _ l = "{} m". format (l) # pitch
Self.lineEdit _ dangju.setText (formatted _ f _ l)

else:
self.timer.stop ()
Self.timer2. stop ()

# pop-up prompt
F _ x = self.calculate _ fx (w, h, Q, x-float (select _ text), h, l) # calculates the latest
available f _ x value again
Msg _ box = QMessageBox ()
Msg _ box.setWindowTitle ("hint")
Msg _ box.setText ("Run to maximum sag point!\ nCurrent sag value is {:. 3f} m".
format (f _ x))
                  Msg _ box.setStandardButtons (QMessageBox.Ok
QMessageBox.Cancel) true
# Set the Chinese text of the button
Msg _ box.button (QMessageBox.Ok). setText ("Confirm")
Msg _ box.button (QMessageBox.Cancel). setText ("Cancel")
# Display the pop-up window and get the users response
response = msg _ box.exec _ ()
```

```
if response = = QMessageBox. Ok:
print ("The user clicked the confirmation button")
else:
print ("The user clicked the cancel button")

Def stop _ pushButton (self):
"" "Stop Timing and Measurement" ""
self.timer.stop ()
Self.timer2. stop ()
self.timer.timeout.disconnect (self.updateRunningTime)
Self.timer2. timeout.disconnect (self.next _ point)

def closeEvent (self, event):
"" "Window closing event, confirmation dialog box pops up" ""
# Define the title and content of the message box
title = confirm exit
message = Are you sure you want to quit?
# Create a custom message box
reply = QMessageBox ()
reply.setWindowTitle (title)
reply.setText (message)
# Add custom buttons
Yes _ button = reply.addButton ( Yes, QMessageBox.YesRole)
            No _ button = reply.addButton ( No , QMessageBox.NoRole)
# Confirmation dialog box pops up
reply.exec _ ()
# Decide whether to accept the shutdown event according to the users selection
if reply.clickedButton () = = yes _ button:
event.accept () # Accepts the closing event
else:
event.ignore () # Ignore closing events
```

```
Def vincenty _ distance (self, lat1, lon1, lat2, lon2, a = 6378137.0, b =
6356752.314245, f = 1/298.257223563):
"""
Calculate the horizontal distance between two points on the earth using the Vincenty
formula.
: param lat1: the latitude of the first point in degrees
: param lon1: longitude of the first point in degrees
: param lat2: the latitude of the second point in degrees
: param lon2: longitude of the second point in degrees
: param a: major half axis of the earth (WGS-84 standard is used by default)
: param b: Earths minor half-axis (WGS-84 standard is used by default)
: param f: Oblateness of the Earth (WGS-84 standard is used by default)
: return: The distance between two points in meters
"""

# Convert degrees to radians
lat1, lon1, lat2, lon2 = map (radians, [lat1, lon1, lat2, lon2])

                    # Difference in longitude
L = lon2-lon1

# naturalized latitude
U1 = atan ((1-f) * tan (lat1))
U2 = atan ((1-f) * tan (lat2))

# iteration variables
sinU1, cosU1 = sin (U1), cos (U1)
sinU2, cosU2 = sin (U2), cos (U2)

Lambda = L
# Perform iterative calculations
for iteration in range (1000):
sinLambda = sin (Lambda)
cosLambda = cos (Lambda)
```

```
sinSigma = sqrt ((cosU2 * sinLambda) * * 2 + (cosU1 * sinU2-sinU1 * cosU2 *
cosLambda) * * 2)
if sinSigma = = 0:
return 0 # If two points coincide, the distance is 0
cosSigma = sinU1 * sinU2 + cosU1 * cosU2 * cosLambda
sigma = atan2 (sinSigma, cosSigma)
sinAlpha = cosU1 * cosU2 * sinLambda/sinSigma
cos2Alpha = 1-sinAlpha * * 2
if cos2Alpha = = 0:
cos2SigmaM = 0 # on the equator line
else:
cos2SigmaM = cosSigma-2 * sinU1 * sinU2/cos2Alpha
C = f/16 * cos2Alpha * (4 + f * (4-3 * cos2Alpha))
LambdaPrev = Lambda
                    Lambda = L + (1-C) * f * sinAlpha * (
sigma + C * sinSigma * (cos2SigmaM + C * cosSigma * (-1 + 2 * cos2SigmaM * *
2)))

# Check convergence
if abs (Lambda-LambdaPrev) < 1e-12:
break
else:
# If the iteration does not converge
return None

# Calculate u2, A and B parameters
u2 = cos2Alpha * (a * * 2-b * * 2)/(b * * 2)
A = 1 + u2/16384 * (4096 + u2 * (-768 + u2 * (320-175 * u2)))
B = u2/1024 * (256 + u2 * (-128 + u2 * (74-47 * u2)))

# compute deltaSigma
deltaSigma = B * sinSigma * (cos2SigmaM + B/4 * (
cosSigma * (-1 + 2 * cos2SigmaM * * 2)-B/6 * cos2SigmaM * (-3 + 4 * sinSigma * *
2) * (
```

```
-3 + 4 * cos2SigmaM * * 2)))

# Calculate final distance
s = b * A * (sigma-deltaSigma)
return s

Def calculate _ fx (self, w, h, Q, x, h, l):
"" "Calculate real-time sag" ""
F _ x = ((x * (l-x)/(2 * h)) * ((w/(l/((l * * 2 + h * * 2) * * 0.5))) + 2 * Q/l))
Return f _ x



If _ _ name _ _ = = _ _ main _ _:
    from PyQt5 import QtCore

QtCore.QCoreApplication.setAttribute (QtCore.Qt.AA _ EnableHighDpiScaling) #
Adaptive Resolution
app = QtWidgets.QApplication (sys.argv)
ui = MainWindow ()
ui.setWindowTitle ("High Voltage Transmission Line Sag Measurement System")
ui.show ()
sys.exit (app.exec _ ())
```

Appendix 3 Wire Parameter Subwindow Code

import sys
import pandas as pd
From PyQt5. QtCore import pyqtSignal
From PyQt5. QtWidgets import QFileDialog
From test _ 1 import UI _ wireParameters # Import classes in UI files
from PyQt5 import QtCore, QtGui, QtWidgets

Create a class that inherits from QMainWindow and Ui _ wireParameters
Class wireParameters _ window (QtWidgets.QMainWindow, Ui _ wireParameters):
SuccessReg = pyqtSignal () # Defines a registration success signal
ValuesSignal = pyqtSignal (float, float, float) # Defines a new signal with three parameters

Def _ _ init _ _ (self):
super (wireParameters _ window, self). _ _ init _ _ ()
self.setupUi (self) # Initialize the UI
self.init () # Calls a custom initialization function
Initialize three parameters
Self.unit _ length _ weight = None
Self.average _ tension = None
Self.concentrated _ load = None

def init (self):
# Connect confirmation button click event to corresponding slot function
Self.pushButton _ confirm.clicked.connect (self.confirm _ pushButton)
Connect the import button click event to the corresponding slot function
Self.pushButton _ import.clicked.connect (self.import _ pushButton)

Def confirm _ pushButton (self):
try:

```
# Signals to pass the read value out
self.ValuesSignal.emit (self.unit _ length _ weight, self.average _ tension, self.concentrated _ load)
self.SuccessReg.emit ()
except Exception as e:
# Print exception information and error file and number of lines
print ("---serial thread exception--:", str (e))
print (ferror file: {e. _ _ traceback _ _. tb _ frame.f _ globals ["_ _ file _ _"]} )
print (ferror line: {e. _ _ traceback _ _. tb _ lineno} )

Def import _ pushButton (self):
try:
# Open the file selection dialog box
File _ name, _ = QFileDialog.getOpenFileName (self, "Select Excel file", "../", "Excel Files (*. xlsx *. xls)")
If file _ name:
# Use pandas to read an Excel file
df = pd.read _ excel (file _ name)
                    Names _ to _ find = [gravity per unit length , horizontal tension, concentrated load ]
values = {}
First _ column _ index = 0 # 630 Wire parameter is the first column with an index of 0
# Traverse the first column (630 wire parameters) to find matching names
for index, name in df.iloc [:, first _ column _ index]. items ():
If name in names _ to _ find:
values [name] = df.iloc [index, 2] # Get the corresponding value from the third column
# Extract the values of weight per unit length, average running tension and concentrated load
Self.unit _ length _ weight = values.get (gravity per unit length )
Self.average _ tension = values.get ( Horizontal tension )
Self.concentrated _ load = values.get ( Concentrated load )
# Set the value of the text box
Self.lineEdit _ weightPerUnitLength.setText (str (self.unit _ length _ weight))
```

```
Self.lineEdit _ runningTension.setText (str (self.average _ tension))
Self.lineEdit _ concentratedLoad.setText (str (self.concentrated _ load))
except Exception as e:
# Print exception information and error file and number of lines
                    print ("---serial thread exception--:", str (e))
print (ferror file: {e. _ _ traceback _ _. tb _ frame.f _ globals ["_ _ file _ _"]} )
print (ferror line: {e. _ _ traceback _ _. tb _ lineno} )


If _ _ name _ _ = = _ _ main _ _:
# Enable high DPI scaling
from PyQt5 import QtCore
QtCore.QCoreApplication.setAttribute (QtCore.Qt.AA _ EnableHighDpiScaling)
app = QtWidgets.QApplication (sys.argv)
# Create a main window object
wireParameters = wireParameters _ window ()
# Display window
wireParameters.show ()
# Turn on the application main loop
sys.exit (app.exec _ ())
```

Appendix 4 Tower Data Child Window Code

From PyQt5. QtCore import pyqtSignal
From project _ test import UI _ MainWindow # Import the UI class of the main window
From test _ 2 import UI _ towerData # UI class for importing tower data window
from PyQt5 import QtCore, QtGui, QtWidgets

Create a class that inherits from QMainWindow and Ui _ towerData to manage the tower data window
Class towerData _ window (QtWidgets.QMainWindow, Ui _ towerData):
SuccessReg = pyqtSignal () # Defines a registration success signal
ValuesSignal = pyqtSignal (str, str, str, str, str, str) # Defines a new signal with six string arguments

Def _ _ init _ _ (self):
    super (towerData _ window, self). _ _ init _ _ ()
self.setupUi (self) # Initialize the UI
self.init () # Calls a custom initialization function
# Initialize six parameters
self.latitude1 = None
self.longitude1 = None
self.height1 = None
self.latitude2 = None
self.longitude2 = None
self.height2 = None

def init (self):
# Connect confirmation button click event to corresponding slot function
Self.pushButton _ confirm.clicked.connect (self.confirm _ pushButton)

Def confirm _ pushButton (self):
try:

```
# Get value from text input box
self.latitude1 = self.lineEdit _ latitude1. text ()
self.longitude1 = self.lineEdit _ longitude1. text ()
self.height1 = self.lineEdit _ height1. text ()
self.latitude2 = self.lineEdit _ latitude2. text ()
self.longitude2 = self.lineEdit _ longitude2. text ()
self.height2 = self.lineEdit _ height2. text ()
# Signals to pass the acquired value out
self.ValuesSignal.emit (self.latitude1, self.longitude1, self.height1,
self.latitude2, self.longitude2, self.height2)
                    # Signal registration success
self.SuccessReg.emit ()
except Exception as e:
# Print exception information and error file and number of lines
print ("---serial thread exception--:", str (e))
print (ferror file: {e. _ _ traceback _ _. tb _ frame.f _ globals ["_ _ file _ _"]} )
print (ferror line: {e. _ _ traceback _ _. tb _ lineno} )


# Main method, from which the program launches the form designed by PyQt
If _ _ name _ _ = = _ _ main _ _:
import sys
from PyQt5 import QtCore, QtWidgets

# Enable high DPI scaling
QtCore.QCoreApplication.setAttribute (QtCore.Qt.AA _ EnableHighDpiScaling)
app = QtWidgets.QApplication (sys.argv)
# Create a tower data window object
towerData = towerData _ window ()
# Display window
towerData.show ()
# Turn on the application main loop
sys.exit (app.exec _ ())
```