Real-time data visualization method for oil pipeline monitoring based on Internet of Things

Qian Wu¹, Xiaolong Chen¹,*, Haoyu Yu¹, Qiang Liu² and Yanhong Yang³

¹School of Mechano-Electronic Engineering, Xidian University, Xi’an, China
²State Key Laboratory for Performance and Structural Safety of Petroleum Tubular Goods and Equipment Materials, CNPC Tubular Goods Research Institute, Xi’an, China
³School of Foreign Languages, Xidian University, Xi’an, China

*Corresponding author e-mail: xlchen@mail.xidian.edu.cn

Abstract. With the increase in the number and length of pipelines in China, pipeline damage accidents are increasing as well. Traditional methods do not allow real-time monitoring of pipelines. The development of the Internet of Things has promoted the monitoring techniques and the development of pipelines. But a problem arises as to how to deal with the huge data generated by monitoring systems. This paper proposes a real-time data visualization method for oil pipeline monitoring based on the Internet of Things. In this method, the measurement nodes of the sensing layer collect the pipeline data and transmit them to the server of the application layer through the network layer. A web application is built based on Spring MVC in the application layer of the Internet of Things to process the data through the interaction between the front and the back. The data visualization algorithm based on Echarts and AJAX is used to position the pipelines and the real-time situation of pipeline anomalies. The results show that with this method, the program runs faster and has higher monitoring efficiency, and the poor readability problem of the oil pipeline data is solved.

1. Introduction

Monitoring of oil pipelines has become increasingly difficult since the extension and increase of oil pipelines. With the development of technology, monitoring the oil pipeline in combination using the Internet of Things(IoT) techniques has become the main trend. However, the amount of data generated by the monitoring process is very large and the readability of huge data is poor[1]. Given this, it is a key research issue that how pipeline maintenance personnel can accurately and quickly acquire pipeline information.

Traditional oil pipeline fault monitoring methods use conventional monitoring, manual monitoring, etc... They are inefficient and cannot monitor long distance pipelines. It is difficult to manage monitoring data, and also cannot be obtained in real time. In order to solve the problems, researchers are constantly exploring new methods. Literature [2] proposed the design of oil pipeline security integrated platform based on Internet of Things technology. In the literature [3], key technologies such as measurement and communication were studied to solve the problem that the pipelines could not be too long. In order to further solve the problem of poor readability of monitoring data, this paper proposes a real-time data visualization method based on Internet of Things.
Using IoT technology, measurement nodes can collect data, which will then be transferred to the aggregation node through the 2.4G module. Data is finally aggregated to the server. And then data will be processed using JavaScript and Echarts. Finally a visual image will be produced[4]. When an exception occurs, the abnormal location can be found quickly. The authors developed a web application based on Spring MVC, connecting the complete IoT pipeline monitoring system to the experiment equipment. The results show that with this method, the oil pipeline can be monitored dynamically in real time.

2. Real-time data visualization algorithm for oil pipeline monitoring

The measuring node in the sensing layer of the oil pipeline monitoring system is equipped with temperature sensors, pressure sensors, pull-out sensors, bending sensors and GPS positioning modules. The measured data is sent to the application layer through the network layer. In order to convert the one-dimensional data in the database into real-time alarm visualization data and a map, it is necessary to obtain real-time measurement data and the online map coordinates converted by GPS coordinates.

2.1. Pipeline positioning and coordinate conversion algorithm

Due to the huge amount of data in the pipeline monitoring system, the chart library selected by this method is Echarts. The coordinate system on the canvas is different from the actual geographic system. If the measured coordinates are not processed before marking the position on the canvas, the result will deviate from the real position. GPS measured coordinates need to be converted to coordinates on canvas. No Baidu map coordinate conversion interface can be directly used in Echarts3.0. This requires us to manually convert the coordinates.

The required coordinate system instance needs to be obtained, and then the dataToPoint and pointToData methods are used to convert the latitude and longitude data and the pixel position. This algorithm is as follows. GeoCoordSys represents the acquired geographic coordinate instance, and current.lng and current.lat are the longitude and latitude of the current pipeline. Then the Baidu Map API is called to output the location.[5]

Procedure

```javascript
var model = myChart.getModel().getSeriesByIndex(opts.series.length - 1);
var geoCoordSys = model.coordinateSystem;
var point = geoCoordSys.dataToPoint([current.lng, current.lat]);
var coord = geoCoordSys.pointToData(point);
End Procedure
```

2.2. Real-time monitoring and alarm algorithm

The data of all nodes are read in real time. Then the server establishes pipeline trend models, and determines whether each node is abnormal. The abnormal nodes will be marked in red to alarm. AJAX asynchronous loading technology is used to request data for 1 second. The abnormal node is obtained by traversing all the data and comparing with the normal range. Figure 1 shows the block diagram of the data judgment alarm algorithm. Then the pipeline trend models will be built based on Echarts. The model building algorithm and alarm method are as follows.

![Figure 1. Data judgment alarm algorithm block diagram.](image-url)
Procedure

```
var dangerIdList = res.exceptionList;
var IsInArray = function(arr,val){
    var testStr=','+arr.join(',')+',';
    return testStr.indexOf(', '+val+',' )!=-1;
};
var nodes = [
    {name: 'name1',
     x: x1,
     y: y1,
     value : value1,
    },
    ......
    {name: 'namen',
     x: xn,
     y: yn,
     value : valuen,
    },
];

nodes.forEach(function(item) {
    item.itemStyle = {
        normal: {
            color: dangerIdList.indexOf(item.value) > -1 ? 'red':'#3398AB'
        }
    }
});
```

End Procedure

Among them, res.exceptionList is an array of abnormal nodes obtained after data security judgment. The nodes contain information such as the name, model coordinates, and value of the measurement. The encapsulated IsInArray method is used to compare and find the nodes in the abnormal node. Finally, the itemStyle of Echarts is used to mark the abnormal ones with a red alarm[6].

3. Algorithm implementation and analysis

3.1. Overall design of the experiment

Several measurement nodes and several aggregation nodes are installed on the experimental pipeline, forming the sensing layer and the network layer. A SpringMVC-based web application develops into the application layer[7]. As shown in Figure 2, the temperature, pressure, pull-off and deformation data obtained from measurement are received by the server and stored in the database. The server backend traverse the database and do data processing according to the front-end request, and return the data through the front-end interface. The data is visualized by the method finally.

![Figure 2. Implementation process diagram.](image-url)
Table 1 shows the environment in which the experimental program is set up. Mysql database is used to store data. The background framework is Spring MVC. The front-end framework is vue.js.

### Table 1. program environment.

| Project                              | value                          |
|--------------------------------------|-------------------------------|
| Operating environment               | Windows7 system or above      |
| Web container                       | Tomcat 9.0                    |
| Database                             | Mysql                         |
| Front frame                          | vue.js                        |
| Background framework                 | Spring MVC                    |
| Browser                              | Chrome                        |

3.2. Experiments and results

As shown in Figure 3, 17 measurement nodes and 3 aggregation nodes were installed on the experimental pipeline. Adding a jitter signal during data acquisition can improve the quality of the acquisition[8].

![Figure 3. Experimental site.](image)

The data was transferred to the database in real time and stored. Part of the data is shown in Figure 4. It is difficult to obtain useful information only from the database. We can not get the information such as the location of a pipeline, and the abnormal state of the node.

![Figure 4. Database part data.](image)

The coordinates of the latitude and longitude of the pipeline measured by the experiment were (108.9138889,34.2363889). We converted the geographic coordinates of the pipeline into screen coordinates and marked the pipeline location by using Baidu map API. The result is shown in Figure 5. We can get the pipeline location from the map near the main building of Xidian University, Yanta District, Xi'an City, Shanxi Province, in line with the experimental site.

![Figure 5. Pipeline location visualization.](image)
We created a pipeline trend model on the visualization page. Then we configured the name, coordinates, and value of each point by using the relationship diagram module in Echarts. We used the myChart.setOption(option) method to render the page. As shown in Table 2, it is a configuration item of each node in the measurement node to the model, where x and y are the coordinates of the node on the canvas in the model, and the value is the node flag.

| Node number | x   | y   | value |
|-------------|-----|-----|-------|
| C20         | 30  | 50  | 20    |
| B19         | 80  | 50  | 19    |
| H3          | 30  | 100 | H3    |
| A15         | -30 | 105 | 15    |
| B14         | -80 | 125 | 14    |
| A13         | -60 | 145 | 13    |
| A11         | -40 | 165 | 11    |
| A12         | -40 | 165 | 12    |
| H2          | -40 | 245 | H2    |
| A10         | -30 | 265 | 10    |
| B8          | 80  | 285 | 8     |
| A4          | 150 | 305 | 4     |
| A3          | 170 | 325 | 3     |
| A7          | 210 | 355 | 7     |
| H1          | 210 | 385 | H1    |
| B2          | 210 | 405 | 2     |

The pipeline trend model can be created after we enter the configuration items. Figure 6 shows the distribution of the pipeline node.

Real-time monitoring algorithm is used to monitor the data. The specified pipeline temperature and pressure safety range are shown in Table 3. AJAX is used to request the back-end timing every 1 second. The server traversed the database and determines whether the current node is abnormal. All abnormal node numbers were encapsulated into the res.exceptionList array. Then the server compared the node model with the IsInArray method, and finally marked the abnormal nodes in red.

| Parameter | Safety range |
|-----------|--------------|
| Temperature | -30~45      |
| pressure    | 0.2~0.8      |
As shown in Figure 7, the normal nodes do not change in color, but the abnormal nodes turn red. The staff can quickly locate the abnormal pipeline. As AJAX technology is applied, the request for data does not need to refresh the entire page. When the measurement data reaches several hundred thousand or more, the traditional full page refresh method will cause the system to be stuck, and our experimental page can quickly present the data visualization graph. This method enhances the readability of the pipeline monitoring data and enables real-time monitoring.

![Pipeline trend schematic](image)

**Figure 7.** Real-time monitoring pipeline trend model.

### 4. Conclusion

This paper combines AJAX technology, with Baidu map API, and uses JavaScript programming language to locate the pipeline and measure data visualization. A real-time visualization method based on the IoT is proposed for oil pipeline monitoring. A complete IoT monitoring system is built to experiment with and verify this method. The result shows that this method can solve the problem of poor readability of huge data and creates the rapid positioning and the real-time monitoring of pipeline. There are some shortcomings in this paper. For example, the rendering effect of the pipeline model is too simple. 3D visualization of the pipeline model is the next research focus.

### 5. Acknowledgments

This work was financially supported by the National Natural Science Foundation of China, Grant No. 61727804, 61471282, 61801358 and 61805187.

### References

[1] Huang W H, Zheng H L, Li M F, Development history and prospects of China's oil and gas storage and operation industry[J]. Oil & Gas Storage and Transportation, 2019(01).

[2] Gao S J, Research on Integrated Platform of Oil and Gas Pipeline Security Based on Internet of Things Technology[A], Technology Analysis, 2018, 1672-7274 (2018) 12-0077-02.

[3] Meng L S, Design and Implementation of Key Technologies for Oil and Gas Pipeline Monitoring and Early Warning Based on Internet of Things[D], Northeastern University, 2015.

[4] Ren X J, Visualization and simulation technology applied in oil&gas pipeline transportation[J], Science Technology and Industry, 2014, 14(1):123-126.

[5] Li Y Z, Sun Y, Song B, Yu F X, Map positioning implementation based on SpringMVC and MyBatis[J], Image Processing&Multimedia Technology, 2019(04).

[6] Wang T P, Wang J S, Wang J, Liu Y C, Design and implementation of ship data visualization analysis system in the south China sea[A], Computer Application and Software,2019(08).

[7] Lu T, Chen X L, Bai W H, Research on environmental monitoring and control technology based on intelligent Internet of Things perception, 2018 International Symposium on Test Automation & Instrumentation, Dalian, China, 2018.7.18-2018.7.20.

[8] Yin T, Chen X L, Research on broadband dither technique in signal acquisition, 2018 International Symposium on Test Automation & Instrumentation, Dalian, China, 2018.7.18-2018.7.20.