Coordinate Inverse and Automatically Mapping of Roadside Stakes for Complex Curve Route

HUAI PING FENG[1], ZHIPENG WANG[1],CHUANG JIANG[2],JIANMEI CHANG[1]*
[1]School of Civil Engineering Shijiazhuang Tiedao University, 17 Northeast, Second Inner Ring , Shijiazhuang, Hebei, P.R.C.
[2] The Third Railway Survey and Design Institute Group Corporation, Tianjing 300251, P.R.C.
E-mail: Fenghuaiping@stdu.edu.cn

Abstract: - With construction of high-speed railway, coordinate inverse problem, especially adopted in more complex types of transition curves becomes time consuming and error prone. The traditional methods in practice have shorts of only applying with limited types of curves, complexity, and tedious. In this paper, a new inverse algorithm is put forward, which is suit for various types of curves and the first iteration point is not necessary to be input but can be random. The iteration can be ensured to be convergence. In this article, using of this algorithm, software has been developed for lofting any types of transition curves by language C# and ACCESS, which makes the laying work more accurate and quick compared with that by hand calculator. The location of the roadside stake is showed by the use of graphic controller Teigha.net API. Tests with existed works show that this method is feasible and it can provide more accurate results quickly and automatically.

Key-Words: - Coordinate inverse; Automatically mapping; Roadside stake; Complex transition curve

1 Introduction

In a long route line, various kinds of route types are required, such as straight line, circular line and types of transition curves to provide a smooth and comfortable transport. Transition curve, also known as a spiral easement, is used to reduce the effects of centrifugal force where a straight section changes into a curve [1-3]. On the other hand, without transition curve the lateral acceleration of a rail vehicle would change abruptly where the straight track meets the curve. The selection of a suitable transition curve is importance towards a proper alignment design in road and railway projects. The Clothoid is exclusively used in road alignments. The cubic parabola is used, for historical reasons only, in railway alignments. With the rapid increase of design speed for highway and railroad, different types of transition curve occurred, such as cubic parabola curve [4], a single C-Bézier curve[5], half-sine wave[6], a fair PH quintic curve for G^2[7], and cubic Bézier curves[8]. A method for the spiral transitions from a straight line to a circle is investigated in [9]. However, as pointed out in [8], transition curves may connect not only a line with a circle but also any two lines with different topology. An S-shaped or a broken back C-shaped spiral transition using a Pythagorean hodograph quintic curve is introduced in [9].

In the construction of roadway and railway with transition curve, coordinate inverse calculation is wildly used. By the coordinate inverse calculation the construction deviation, such as the cross-section of embankment in curve can be quickly calculated. Roadside stake is one of the elements used to control the width of the road. For the roadside stake of the roadway, determine the mileage of the stake by the coordination of arbitrary point is just one kind of the coordinate inverse calculation work. The data source of roadside stake can be the actual measured data or the points selected on the design drawings.

The inverse calculation of roadside stake in straight line and circular curve is simple[10]. It is difficult in transition curves. At present, arch simulation method [11], iterative methods [11] are used in calculation of transition curves. Study on the relationship between transition curve and neighbor point was made in[12]. algorithms and mapping techniques of roadside piles are presented in [13-15]. But these methods can only deal with a few type of curve such as Spiral curve, and have the shorts of hard work and tedious. However, curve which mainly follows the type of Spiral curve now cannot meet the requirements of high-speed railway projects and new types of transition curves have been proposed. Lofting and coordination inverse calculation for those transition curves by hand calculator became a very heavy work and it is also easy to have errors. So it is necessary and meaningful to research and develop software which suit to any type of transition curves more accurately and quickly.

In this paper, a new coordinate inverse algorithm is put forward, which is suitable for any random type of transition curves. According to this algorithm, the start point for iteration is not need to be input. It is not need to distribute the section on the type of curves. The convergence of the algorithm can be guaranteed. It is very suitable for calculation using computer program.

Under this model, computer program is work out, and by comparing the calculate result with others, the accuracy and goods of it are shown. Compared to the existing survey software[16-19], the program have benefits on calculating quickly. In the same time, the calculation results are visible which is easy to operate.

2 Algorithm of Inverse Calculation

As shown in Fig.1, the location of point P to the curve AB can be determined by two amounts, l_p and D_p exclusively. PP' is vertical to curve AB and D_p is the vertical distance for point P to the curve AB. l_p is the length of the curve from start point A to P' on the curve.

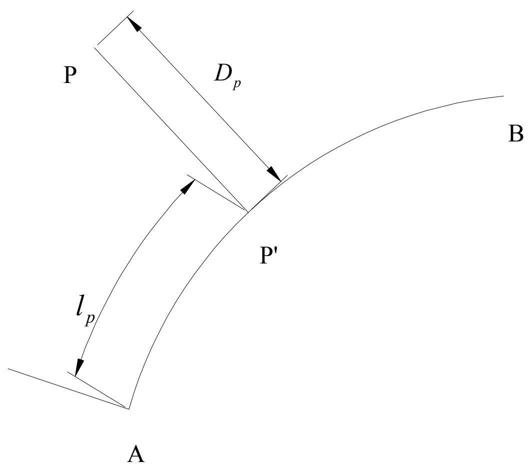

Figure 1 *Coordination elements of roadside stake*

As Fig.2 shows, PP' is the normal line of the curve that goes through the point P. For normal lines through every point on the curve, if the distance from P to the normal line is 0, then, that point is P'. P' can be determined by this method. Following is the iterative processes to finish the inverse calculation:
The method for inverse calculation of transition curve is described by the following algorithm.

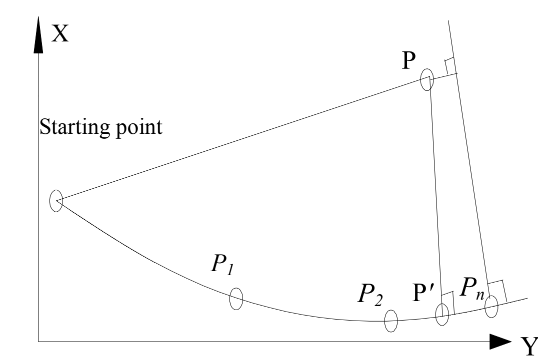

Figure 2 *Inverse calculation processes*

Algorithm 1 *Iterative calculating process*

1. **Input** : coordinates of any start point p_0 (arbitrary)on line and the existing point P .
2. Calculate the distance l_0 that from P to P_0 the (arbitrary point) of curve and let $l = l_0$. Let the curve length from start point P_0 to P_1 on the curve be equal to l and calculate the coordinate of point P_1.
3. Calculate the distance l_1 from P to P_1 and let $l = l_0 + l_1$. Let the curve length from the start point to P_2 be equal to l and calculate the coordinate of point P_2 on the curve.
4. Iterative as step 2, and so on.
5. **If** $l_{n-1} < l_{n-2}$, which means iterative point is left to point P', let the calculate distance l_n >0
6. **If** $l_{n-1} > l_{n-2}$, which means iterative point is right to point P', change l_n to be the distance that from P to the normal line of P_n and let l_n <0.
7. **When** $l_n \cong 0$, iterative point P_n is the foot point P'. This algorithm can make the iteration convergence.

In final step of above algorithm, the azimuth of normal line on P_n must be calculated. Let the azimuth $\alpha_{P_n}^{v}$ of the normal line on P_n and azimuth $\alpha_{P_n}^{\tau}$ of the tangent line on point P_n respectively, we can get the following relationship:

$$\alpha_{P_n}^{v} = \begin{cases} \alpha_{P_n}^{\tau} - \pi/2, & if \quad \alpha_{P_n}^{\tau} < 0 \\ \alpha_{P_n}^{\tau} + 2\pi, & if \quad \alpha_{P_n}^{\tau} > 0 \end{cases} \tag{1}$$

Thus can make the direction of normal line on iterative point is to the left of the route. The distance from the point P to the normal line can be got from the following equation:

$$D_P = (Y_P - Y_{P_n})\cos\alpha_{P_n}^{v} - (X_P - X_{P_n})\sin\alpha_{P_n}^{v} \tag{2}$$

While l_n=0, point P' can be decided when P' is known. Through the compare between the azimuth angle $\alpha_{P'P}$ of line PP' and azimuth angle $\alpha_{P'}^{v}$ of normal line on P', the position of point P in the route line can be made certain. Because the normal line is point to the left side of the route, if $\sin\alpha_{P'P} = \sin\alpha_{P'}^{v}$ the point P is located on the left side of the route; if $\sin\alpha_{P'P} = -\sin\alpha_{P'}^{v}$, point P is on the right side of the route.

3 Computer program

In order to confirm that the algorithm described above can be used properly, a computer program was developed with C# programming language. The program is capable of forward and inverse calculation. To improve the calculating speed, ACCESS is used to store all of the coordination data of the line. Teigha.net API is used for visual mapping. For any random type of curve, following basic items are defined: starting azimuth angle α_A, the coordinates of the start point (X_A, Y_A), starting point curvature $1/R_A$, ending point curvature $1/R_B$, curve length L and deflection coefficient. Firstly, a data structure, such as XianYuan(line segment, such as straight line, circular curve, transition curve) is defined to store the data for further calculation. The structure of it is as following:

Public struct XianYuan{ *double x0; double y0; string quxianleixing; double qslicheng; double qsqulu; double qsfangweijiao; double jslicheng; double jsqulu; double pianzhuancs;*}

After these factors of the route are input, the data will be stored in the database in the form of *XianYuan*. Then a whole view of the route line can be drawn by use of drafting function.

For any roadside stakes, by computing the distance from the points on the line to the given roadside stake, the line segment that it belongs to can be determined. Then using the basic factors of the line segment in which it belongs to and taking the starting point of this segment as the initial point, the coordinate inverse computation can be conducted following the algorithm as mentioned above. Then the mileage of P', vertical distance D_p, the coordinate of the foot point P' and the azimuth angle $\alpha_{P'P}$ of line P'P can be obtained.

The basic procedure used to conduct the coordinate inverse computation is shown as below:

- Using the coordinates of two points to compute the distance:
  double *getJuli(PointF p1, PointF p2)*
- Using the mileage of point P_n to conduct azimuth angle of the normal line:
  double *getFaXianfwj(XianYuan xy, double licheng)*
- Computing the distance between given point to line
  double *getFaXianJuli(PointF p1，double faxianfwj, PointF p2)*
- Using the coordinates of two points to compute the azimuth angle:
  double *getZuoBiaofwj(PointF p1, PointF p2)*
- Using the mileage of a point to get its coordinates:
  PointF getPoint(XianYuan xy, double licheng)

The procedures of calculation (using the starting point as initial computing point) is as following：

```
PointF p1;
double ln = getJuli(p, p1);
double ln2 = ln;
double l = ln;
while (ln2 < ln)
{
PointF pn = getPoint(getXianyuan(l)，  l);
ln2 = getJuli(p, pn);
l += ln2;
}
double fxjuli = 1;
double jingdu=0.00001;
while (Math.Abs(fxjuli) > jingdu)
{
 Fxjuli=
getFaXianJuli(p,getFaXianfwj(getXianyuan(l),l),
getPoint(getXianyuan(l),l));
l += fxjuli;
}
```

If *Math.Abs(fxjuli)*, the absolute value of the distance between point P and the normal line on P_n is smaller than the given very small value(define as *jingdu*), the loop is over. *jingdu* works as a controlling parameter for precision. The mileage of foot point is known after the loop is over and then D_p and the coordinates of the roadside stake and so on can be calculated.

4 Test of the Method and Analysis of Obtained Results

In order to examine whether the method and the program work properly, project mentioned in literature [7] is adopted. This project has more than one converge point on the curve. After inputting the line segment information of the engineering project, the whole line landscape is shown as Fig.3. The results of the project's inverse coordinate are shown in Table 1. In the Table 1 the coordination of roadside stake is known and the value of mileage l_p, azimuth angle $\alpha_{P'P}$, coordination of point P'(foot point) are calculated.

Inputting the given coordinates of roadside stake points in Table1 and using above mentioned algorithm, inverse computation is carried out. The results are shown in Table2.

As we can see from the contrast between Table1 and Table2, the algorithm used in this article is highly precise in its calculation.

Table 1 Inverse coordinate calculated

Roadside stake		Mileage of foot point l_p /m	Azimuth $\alpha_{P'P}$ /(° ′ ″)	Foot point P'		D_p /m
x_j /m	y_j /m			$x_{jp'}$ /m	$y_{jp'}$ /m	
2538188.153	504306.837	477.832	214 10 54.75	2538190.120	504303.941	3.501
2538072.386	504340.868	611.653	137 1 52.89	2538075.779	504344.511	4.978
2538026.350	504391.798	678.904	149 3 38.18	2538022.375	504385.167	7.731
2538186.156	504619.943	144.592	285 5 36.18	2538194.031	504622.067	8.157

Table 2 Inverse computing results from the model

Roadside stake		Mileage of roadside stake l_p /m	Azimuth $\alpha_{P'P}$ /(° ′ ″)	Foot point P'		D_p /m
x_j /m	y_j /m			$x_{jp'}$ /m	$y_{jp'}$ /m	
2538188.153	504306.837	477.8319	214 10 5461	2538190.1200	504303.9407	3.5011
2538072.386	504340.868	611.6525	137 1 5271	2538075.7792	504344.5107	-4.9783
2538026.350	504391.798	678.9038	149 3 3820	2538022.3753	504385.1670	7.7310
2538186.156	504619.943	144.5924	285 5 3625	2538194.0315	504622.0670	8.1569

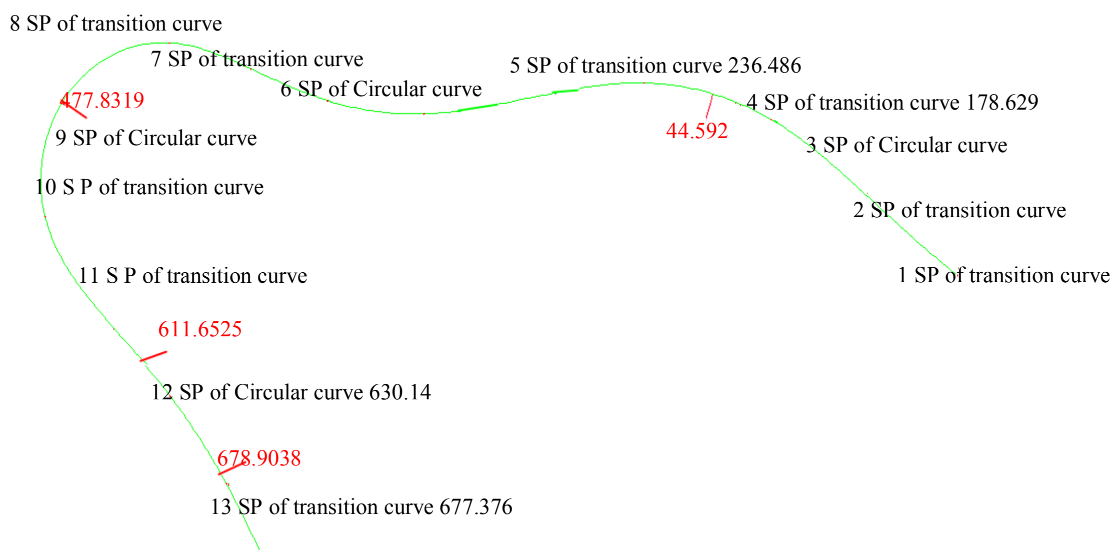

Figure 3 View for roadside stake

Besides, the calculation can start without the inputting of initial point. Mileage of roadside stake l_p .etc. can be automatically gotten by computer programs. The algorithm can perfectly meet the need of highway or railway engineering construction. In addition, using the view function, the relationship between the roadside stake point and the route line is shown as Fig.3. It can be found that all of the points can be inverse calculated freely.

Table 3 shows ordinates of the clothoid endpoints and the ordinates of point

Compared with other methods[11], this method have the benefit not only on time cost, but also on accurate. As we know that by method [11], before carry out a inverse calculation, point P must be approximately located to a segment zone, such as strait line, transient curve or a circle. Otherwise, a long calculating time is needed to get convergence. Because any existing point on curve can be regard as start point in the proposed method, it can also be found that computation time significantly decreased compared to method [11]. As shown in step6, l_n is changed to be the distance that from P to the normal line, which makes the calculation more quickly and accurate.

Our computations were compared with reference [11], and performed on a Intel(R) Core(TM) 2 Duo CPU 2.26 GHz computer. To test the calculation time for inverse problem, six cases were carried out. For each case, a series of given points were calculated, and the calculating time were recorded. Numbers of the point for each case were 4, 75, 1350, 2700, 4050 and 6000 respectively, which are arbitrarily distributed around the line as shown in Fig. 3. It can be simply seen from Tables 3 that computation times for proposed method were up to almost 1.7 times faster than using the method [11], when points were more than 1000.

Table 3 Comparison of computation time (in seconds) .

Case No.	Point No.	Ref. 11	Proposed
1	4	0.001	0.001
2	75	0.071	0.053
3	1350	1.132	0.782
4	2700	1.915	1.130
5	4050	3.023	1.788
6	6000	5.214	3.134

5 Conclusion and Future Work

In this article a new coordinate inverse algorithm is put out, which can be applied to types of curves and no inputting initial iterative point and no section distribute is needed. The following can be drawn from this study.

(1) The procedure proposed at step 6 in algorithm compares well with the existing coordinate inverse algorithm. This can significantly reduce the time required to meet convergence with reasonable accuracy.

(2) The mathematic types of transition curve are not taken into account in proposed algorithm. As a result, this method is suitable to any types of curves.

(3)With this new algorithm, programmed in language of C# and store the route line information of the project in ACCESS database, we compared the calculations with the existing engineering projects. The results show that fast and visual inverse coordinate computation can be achieved with high exactness under this algorithm.

Despite using a relatively inefficient runtime system in our experiments, the results are carried out in computer, which lead to some inconvenience in some cases. Our future work includes converting the program to Java, which can operate in cell phone. By this way, the coordination inverse problem can solved quickly in situ.

Acknowledgement

This project is supported by the Natural Science Foundation of China (51478279) and the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry (China). These financial supports are gratefully acknowledged.

References:

[1] ERGIN T, ORHAN B. A new transition curve with enhanced properties. *Canadian Journal of Civil Engineering,* 2005, 32 (5): 913-923.

[2] Jiang,C., Feng, H. A new layout software for transition curves. *Geotechnical investigation and surveying*, Vol.42(1), 2014,pp.83-86.

[3] S., Lian, J., Liu, X., Li,. Test Verification of Rationality of Transition Curve Parameters of Dedicated Passenger Traffic Railway Lines. Journal of the China Railway Society, Vol.28 (6), 2014, pp.88-91.

[4] Eva Blažková, Zbyněk Šír. Identifying and approximating monotonous segments of algebraic curves using support function representation. *Computer Aided Geometric Design,* Vol. 31(7), 2014, pp.358-372.

[5] R., Winkel. On a generalization of Bernstein polynomials and Bézier curves based on umbral

calculus. *Computer Aided Geometric Design*, Vol.31, 2014, pp227-244.

[6] Ziatdinov, R., Yoshida,N., Kim,T. Fitting G2 multispiral transition curve joining two straight lines. *Computer-aided design*, Vol. 44,2012, pp.591-596.

[7] G., Xu, X.,Yang. Construction of several second- and fourth-order geometric partial differential equations for space curves. *Computer Aided Geometric Design*, Vol. 31(2), 2014, pp.63-80.

[8] Yoshida N, Saito T. Interactive aesthetic curve segments. The Visual Computer, Vol.22(9), 2006,pp.896-905.

[9] Habib Z, Sakai M. G2 Pythagorean hodograph quintic transition between two circles with shape control. Computer Aided Geometric Design, Vol.24(5),2007,pp.252-66.

[10] Du Ning, a fast method for determination of relationship between ground points and road; Bulletin of surveying and mapping,1999,11,20-21

[11] Qin Hui, Direct arithmetic of coordinate inverse on a spiral curve, Site Investigation Science and Technology, Vol.5,2010,pp.18-23.

[12] Xiao Guang Wang et al., Study on the Relationship between Transition Curve and Neighbor Point, 2011, Advanced Materials Research, 243-249, 5858

[13] Liu De-er et al., an algorithm and use for automatically mapping roadside piles, 2011, science of surveying and mapping,36(5),211-213

[14] Chen Rui, wang jinsong, further discussion of subgrade side pile lofting,2013, 2, Beijing surveying and mapping,62-65

[15] J.,Wang.i, The method, program and application of route arbitrary mileage centerline and side pegs inverse coordinate calculation, Urban Geotechnical Investigation & Surveying, Vol.26,No.1,2009,pp.114-116.

[16] Z., Zhang, Road survey and design, Science Press, Beijing, 2005.

[17] Li,Q., Discussion on monitoring method for center curve of highway, Bulletin of Surveying and Mapping, vol(5),2006,PP:21-23.

[18] Gao,S., Shi,D., A quick solution to a nonlinear equation of spiral curve, Bulletin of Surveying and Mapping, Vol. 3, 2006, pp:28-30.

[19] Tan Hui, CASIO fx-5800P Program calculator: highway and railway construction surveying program, Shanghai; Tongji University Press. 2009.