The Journal of Navigation (2023), **76**:1 103–112
doi:10.1017/S037346332200056X

FORUM

Analysis and Discussion on Ellipsoidal and virtual sphere models used for transverse arrangement INS

Zhe Wen, Hongwei Bian, Heng Ma, and Rongying Wang

College of Electrical Engineering, Naval University of Engineering, Wuhan, China.
(E-mail: mkggrr2004@aliyun.com)

Received: 15 May 2022; **Accepted:** 11 October 2022

Keywords: polar navigation; inertial navigation; transverse navigation; ellipsoidal earth model

Abstract
Transverse arrangement is one of the main methods used in the polar inertial navigation system (INS). In the traditional algorithm, the calculation of using the earth ellipsoid model is complex, while using the earth sphere model cannot satisfy a high-accuracy application. Therefore, an approach based on the virtual sphere model is proposed, which has been proved in simulation experiments to reduce the computational complexity and maintain the same accuracy as the ellipsoid algorithm, but its accuracy has not yet been proved in theory. Starting from the basic principles of the ellipsoid and virtual sphere model algorithm, this paper compares the key formulations of the two. Finally, it is proved that the two arrangements are actually the same.

1. Introduction

Polar navigation has been a hot topic in recent years, owing to the Arctic routes development, the exploitation of resources and the military forces in the polar region. The inertial navigation system (INS) is a passive autonomous navigation system that is not disturbed by the external environment and has become the most important navigation technology in the polar region because of the complex electromagnetic environment (Bian et al., 2020). Due to special factors, such as the convergence of longitude and the close collinearity of the vector between the earth rotation angular rate and gravity, the INS under the traditional arrangement faces problems such as the sharp increase of heading error and calculation overflow. Therefore, polar navigation methods such as grid navigation and transverse navigation need to be used.

The application of the transverse method has a long history. In 1958, the US 'Nautilus' nuclear submarine equipped with the N6A INS successfully crossed the Arctic (Curtis and Slater, 1959), which used the transverse navigation method. Initially, the earth sphere model was used in the INS arrangement, which led to an approximation error, so it was considered not suitable for long period navigation (Lin et al., 2019). After several decades of technology development, the transverse navigation method based on a proposed ellipsoid model solved this problem (Xu and Dou, 2014; Yao et al., 2016). It is worth noting that the ellipsoid transverse algorithm is much more complex than the traditional algorithm, but the complexity of these algorithms in the spherical model is basically the same, which will be discussed in detail in Section 2.

To optimise the algorithm and reduce the complexity of the solution, Qin et al. (2018) proposed the transverse algorithm with the virtual sphere model. In simulations, the solving accuracy is commensurate with the transverse ellipsoid arrangement, and the computational complexity is much lower than the latter. Many scholars have also carried out subsequent research, such as constructing integrated navigation

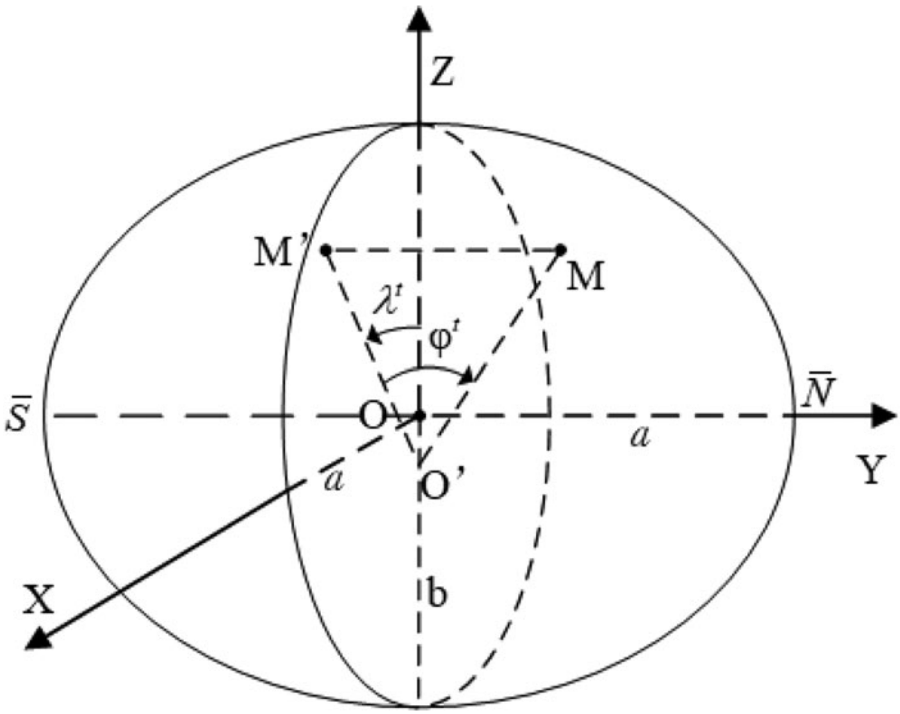

Figure 1. *Definition of transverse latitude and transverse longitude..*

with the algorithm (Ge et al., 2021) or proposing new navigation methods inspired by the algorithm (Liu et al., 2020).

It is considered that the simulation results are inadequate to support the confirmation and promotion of the virtual sphere model algorithm. Therefore, after analysing the essence of the ellipsoid algorithm and the virtual sphere model algorithm, this paper compares the calculation formulae of key variables in the two arrangements and proves that the two arrangements are the same. This finding is expected to promote the application and development of inertial navigation methods in the future.

2. Analysis of ellipsodial transverse navigation algorithms

2.1. Transverse latitude and transverse longitude

The latitude and longitude of the Earth used in navigation are measured from geodesy. The latitude of any point on the ellipsoid surface is defined as the angle between the normal vector on the ellipsoid and the equatorial plane, and the longitude is defined as the angle between the projection of the normal vector on the ellipsoid in the equatorial plane and the axis, which connects the geocentre and the prime meridian in the equatorial plane.

Similarly, the definition of transverse longitude and latitude is shown in Figure 1.

The Earth ellipsoid model equation can be expressed as

$$F(x, y, z) = \frac{x^2}{a^2} + \frac{y^2}{a^2} + \frac{z^2}{b^2} - 1 = 0 \tag{1}$$

and the normal vector on M (x_0, y_0, z_0), i.e. O'M, is

$$\boldsymbol{n} = (F_x, F_y, F_z)|_M = \left(x_0, y_0, \frac{z_0}{1-e^2}\right) \tag{2}$$

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

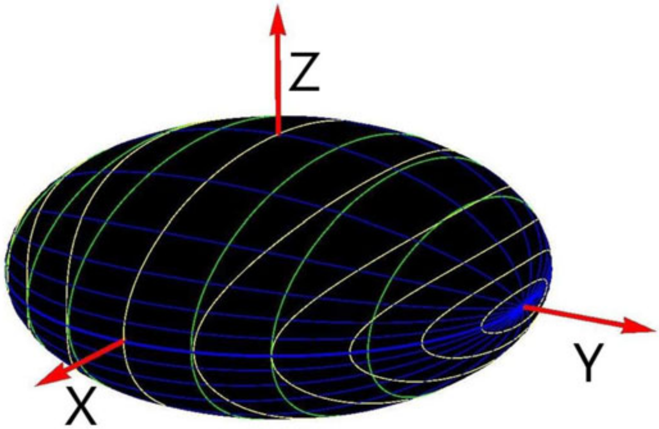

Figure 2. *Schematic diagram of ellipsoid transverse meridians and parallels..*

where e denotes the eccentricity of the ellipsoid, $e = \sqrt{a^2 - b^2}/a$.

The transverse latitude at M is the angle between the normal vector $\boldsymbol{n}$ and the X-O-Z plane, so the point with the same transverse latitude φ^t satisfies Equation (3), which is an elliptical cone with an opening to the Y-axis:

$$F_1(x, y, z) = \sin\varphi^t - \frac{y}{\sqrt{x^2 + y^2 + \left(\frac{z}{1-e^2}\right)^2}} = 0 \tag{3}$$

Another way to define the transverse latitude is called auxiliary transverse latitude $\bar{\varphi}^t$:

$$F_1'(x, y, z) = \sin\bar{\varphi}^t - \frac{y}{a} = 0 \tag{4}$$

It is shown that the auxiliary transverse latitudes are the elliptical intersection formed by planes parallel to X-O-Z and the ellipsoid, which refers to the definition method of the grid heading, and has the advantage of simple calculation and representation, but it is not the latitude measured in geodesy.

The transverse longitude at M is the included angle between O'M' and the Z-axis, so the point with the same transverse longitude λ^t satisfies Equation (5), which is the plane passing through the Y-axis:

$$F_2(x, y, z) = \tan\lambda^t - \frac{x(1-e^2)}{z} = 0 \tag{5}$$

The transverse meridian and parallel, that is, the curve formed by the points of the same transverse longitude (latitude) on the ellipsoid surface, are derived by simultaneously solving F and F_2 (or F_1). Figure 2 shows some transverse meridians, parallels and auxiliary transverse parallels on the surface of the ellipsoid, where the blue line represents the transverse meridians, the yellow line represents the transverse parallels and the green line represents the auxiliary transverse parallels. The complexity and irregularity can be seen intuitively.

2.2. Difficulties in constructing transverse geographic coordinate system

The tangent direction of transverse meridians, parallels and auxiliary transverse parallels passing through M(φ^t,λ^t) are to be calculated first.

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

The relationship between the Earth Centred Earth Fixed and the transverse longitude and latitude is

$$\begin{cases} x = R_N \cos\varphi^t \sin\lambda^t \\ y = R_N \sin\varphi^t \\ z = R_N (1-e^2)\cos\varphi^t \cos\lambda^t \end{cases} \tag{6}$$

Suppose the surface normal vector determined by Equation (3) is $\boldsymbol{n}_1$, and the direction (tangent) vector of the transverse parallel is $\boldsymbol{T}_1 = \boldsymbol{n} \times \boldsymbol{n}_1$; similarly, suppose the surface normal vector determined by Equation (5) is $\boldsymbol{n}_2$, then the direction (tangent) vector of the transverse meridian is $\boldsymbol{T}_2 = \boldsymbol{n} \times \boldsymbol{n}_2$. In combination with Equation (6), the transverse longitude and latitude can be used to represent the direction (tangent) vectors $\boldsymbol{T}_1$, $\boldsymbol{T}_2$ and $\boldsymbol{T}_3$, which respectively represent the direction (tangent) vector of transverse parallels, transverse meridians and auxiliary transverse parallels:

$$\begin{cases} \boldsymbol{T}_1 = [\cos\lambda^t(1-e^2\cos^2\varphi^t), -e^2\sin\varphi^t\cos\varphi^t\sin\lambda^t\cos\lambda^t, -(1-e^2)\sin\lambda^t] \\ \boldsymbol{T}_2 = [-\sin\varphi^t\sin\lambda^t, \cos\varphi^t(1-e^2\cos^2\lambda^t), -(1-e^2)\sin\varphi^t\cos\lambda^t] \\ \boldsymbol{T}_3 = [\cos\lambda^t, 0, -\sin\lambda^t] \end{cases} \tag{7}$$

It is difficult to define an orthogonal transverse geographic coordinate system with reference to traditional methods because $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are not perpendicular. In application, the tangent direction $\boldsymbol{T}_3$ of the auxiliary transverse parallels is generally defined as the east direction of the transverse geographic system. The north direction of the transverse geographic system, which is not the direction determined by $\boldsymbol{T}_2$, is determined by the right-hand rule with the up direction (normal vector on the ellipsoid).

Therefore, the transverse eastward velocity not only leads to the change of transverse longitude, but also changes the transverse latitude; the transverse north velocity not only leads to the change of transverse latitude, but also changes the transverse longitude, which is the direct reason for the complexity of the arrangement in a transverse ellipsoid INS. The deeper reason concluded is that the earth ellipsoid model is a rotating ellipsoid around the Z-axis, not around the X-axis or Y-axis, so the transverse meridians and parallels are not as regular as the traditional ones.

2.3. Key algorithms of ellipsoid transverse arrangement

Given the attitude and velocity update of an ellipsoid transverse arrangement are mostly similar to the traditional algorithms, this section mainly focuses on the position update algorithm and the calculation of the angular velocity of the transverse geographic system (t system) relative to the earth system (ECEF system) caused by the carrier motion, $\boldsymbol{\omega}_{et}^t$. Since predecessors have already studied the algorithm, most of the derivation details will be omitted here.

According to the definitions of ellipsoidal transverse longitude, latitude and the ellipsoidal transverse geographic coordinate system (t system), $\boldsymbol{C}_t^e$ is calculated as

$$\boldsymbol{C}_t^e = \begin{pmatrix} \cos\lambda^t & -\sin\varphi^t\sin\lambda^t & \cos\varphi^t\sin\lambda^t \\ 0 & \cos\varphi^t & \sin\varphi^t \\ -\sin\lambda^t & -\sin\varphi^t\cos\lambda^t & \cos\varphi^t\cos\lambda^t \end{pmatrix} \tag{8}$$

According to the matrix differential formula,

$$\dot{\boldsymbol{C}}_t^e = \boldsymbol{C}_t^e(\boldsymbol{\omega}_{et}^t \times) \tag{9}$$

then

$$\boldsymbol{\omega}_{et}^t = (-\dot{\varphi}^t, \dot{\lambda}^t\cos\varphi^t, \dot{\lambda}^t\sin\varphi^t)^{\mathrm{T}} \tag{10}$$

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

For the calculation of change rate of transverse longitude $\dot{\lambda}^{\mathrm{t}}$ and latitude $\dot{\varphi}^{\mathrm{t}}$, a decomposition formula with Earth Cartesian position vector $\boldsymbol{r}^e$ is to be used:

$$\dot{\boldsymbol{r}}^e = \boldsymbol{v}^e = \boldsymbol{C}_t^e \boldsymbol{v}^t = \frac{\mathrm{d}\boldsymbol{r}^e}{\mathrm{d}\varphi^{\mathrm{t}}}\dot{\varphi}^t + \frac{\mathrm{d}\boldsymbol{r}^e}{\mathrm{d}\lambda^{\mathrm{t}}}\dot{\lambda}^t \tag{11}$$

Equation (11) can be divided into two parts for the convenience of calculation:

$$\begin{cases} \boldsymbol{C}_t^e (v_{tE}^t, 0, 0)^{\mathrm{T}} = \dfrac{d\boldsymbol{r}^e}{d\varphi^t}\dot{\varphi}_{tE}^t + \dfrac{d\boldsymbol{r}^e}{d\lambda^t}\dot{\lambda}_{tE}^t \\ \boldsymbol{C}_t^e (0, v_{tN}^t, 0)^{\mathrm{T}} = \dfrac{d\boldsymbol{r}^e}{d\varphi^t}\dot{\varphi}_{tN}^t + \dfrac{d\boldsymbol{r}^e}{d\lambda^t}\dot{\lambda}_{tN}^t \end{cases} \tag{12}$$

where $\dot{\varphi}_{tE}^t$, $\dot{\lambda}_{tE}^t$ denotes the change rate of transverse latitude and longitude caused by eastward transverse velocity, respectively, and $\dot{\varphi}_{tN}^t$, $\dot{\lambda}_{\mathrm{tN}}^{\mathrm{t}}$ denotes the change rate of transverse latitude and longitude caused by northward transverse velocity, respectively. Substituting in Equations (6) and (8) and combining the expression of prime vertical circle R_N denoted by transverse longitude and latitude:

$$R_N = \frac{a}{\sqrt{1 - e^2\cos^2\varphi^t\cos^2\lambda^t}} \tag{13}$$

The position updating formula of ellipsoid transverse arrangement can be derived, which can also be used in the calculation of $\boldsymbol{\omega}_{et}^t$:

$$\begin{cases} \dot{\varphi}^t = \dot{\varphi}_{tE}^t + \dot{\varphi}_{tN}^t = \dfrac{V_{tN}^t(1 - e^2\sin^2\lambda^t - e^2\cos^2\varphi^t\cos^2\lambda^t) + e^2 V_{tE}^t \sin\varphi^t \sin\lambda^t \cos\lambda^t}{R_N(1 - e^2)} \\ \dot{\lambda}^t = \dot{\lambda}_{tE}^t + \dot{\lambda}_{tN}^t = \dfrac{V_{tE}^t \sec\varphi^t(1 - e^2\cos^2\lambda^t) + e^2 V_{tN}^t \tan\varphi^t \sin\lambda^t \cos\lambda^t}{R_N(1 - e^2)} \end{cases} \tag{14}$$

3. Construction of virtual sphere model and its application in transverse navigation

3.1. Virtual sphere model in traditional coordinate system

According to the literature (Qin et al., 2018), the idea of virtual sphere model design can be extracted: (a) based on the fact that the radius of the earth is closely related to its velocity in inertial navigation calculation; (b) a 'virtual sphere' model can be regarded as an artificial ball with radius of curvature in Prime Vertical. It can be concluded that the transition of the velocity and the unified construction of radius are of key importance.

This can be further explained in the traditional coordinate system. In the movement of the surface carrier, the curvature radius related to the eastward velocity in the INS position update is the Prime Vertical radius R_N, and the curvature radius R_M of the Prime Meridian circle is related to the northward velocity. If the northward velocity is multiplied by the ratio of R_N to R_M, which is defined as the virtual northward velocity, and the virtual northward velocity in the position update equation is to be divided equally with the original eastward velocity by R_N, then the constructed virtual resultant velocity is equal to the combination of the virtual northward velocity and the original eastward velocity, as if the carrier moves on the sphere with radius R_N at the virtual resultant velocity. See Figure 3.

Figure 3(a) shows the motion in the real situation, with the geographical eastward speed $\boldsymbol{v}_E$ and northward speed $\boldsymbol{v}_N$.

Figure 3(b) shows the virtual sphere model corresponding to Figure 3(a), with the position (i.e. the longitude and latitude) of the carrier unchanged. The radius of the virtual sphere is R_N, and the carrier

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

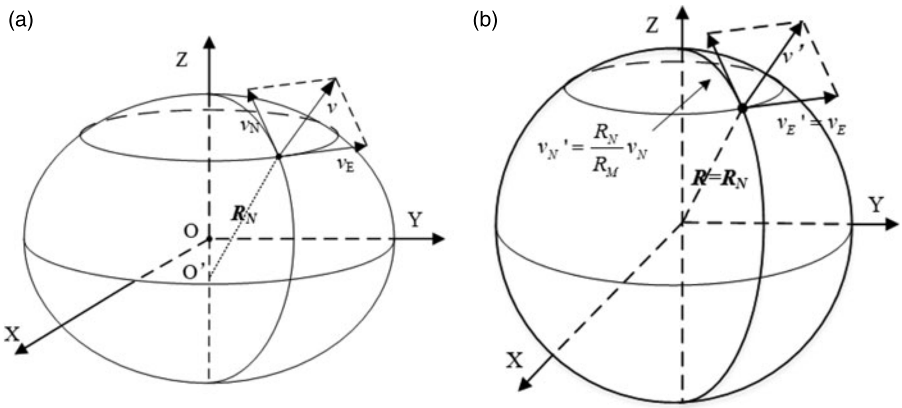

Figure 3. *Transformation from ellipsoid to virtual sphere..*

speed is replaced by the virtual speeds $\boldsymbol{v}_E{}'$ and $\boldsymbol{v}_N{}'$:

$$\begin{cases} v_E{}' = v_E \\ v_N{}' = \dfrac{R_N}{R_M} v_N \end{cases} \tag{15}$$

The position update formula is

$$(\dot{\varphi}, \dot{\lambda}) = \left(\frac{v_N}{R_M}, \frac{v_E}{R_N \cos \varphi} \right) = \frac{1}{R_N} (v_N{}', v_E{}' \sec \varphi) \tag{16}$$

In the traditional arrangement, there is obviously no difference between the virtual sphere method and the general algorithm, which can only be regarded as a new way to understand the problem. However, in the transverse arrangement of the ellipsoid, the advantages of applying the virtual sphere model are revealed.

3.2. Key algorithms of INS transverse arrangement based on virtual sphere model

The core idea of the arrangement is to transfer the transverse velocity to the geographical system for changing the northward velocity $\boldsymbol{v}_N$, and then the processed resultant velocity is transferred back to the transverse system (t system) but substitutes into the virtual sphere with R_N as the radius for transverse position update. The process avoids complex calculation of many related curvature radii.

Therefore, the relationship between the virtual transverse velocity $\boldsymbol{v}^{t\prime} = (v^{t\prime}_{tE}, v^{t\prime}_{tN}, 0)^T$ and the real transverse velocity $\boldsymbol{v}^t = (v^t_{tE}, v^t_{tN}, 0)^T$ is

$$\boldsymbol{v}^{t\prime} = \boldsymbol{C}^t_g \begin{pmatrix} 1 & & \\ & \dfrac{R_N}{R_M} & \\ & & 1 \end{pmatrix} \boldsymbol{C}^g_t \boldsymbol{v}^t \tag{17}$$

where $\boldsymbol{C}^t_g$ is the transformation matrix from transverse geographic coordinate system to geographic coordinate system (g system):

$$\boldsymbol{C}^t_g = \begin{pmatrix} \cos\sigma & -\sin\sigma & 0 \\ \sin\sigma & \cos\sigma & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{18}$$

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

and

$$\begin{cases} \cos\sigma = \dfrac{-\sin\varphi^t\cos\lambda^t}{\sqrt{1-\cos^2\varphi^t\cos^2\lambda^t}} \\ \sin\sigma = \dfrac{\sin\lambda^t}{\sqrt{1-\cos^2\varphi^t\cos^2\lambda^t}} \end{cases} \tag{19}$$

Using the virtual transverse velocity and the radius R_N, the transverse position updating equation and the calculation formula of ω^t_{et} can be written as

$$(\dot{\varphi}^t, \dot{\lambda}^t) = \frac{1}{R_N}(v^{t\prime}_{tN}, v^{t\prime}_{tE}\sec\varphi^t) \tag{20}$$

$$\omega^t_{et} = \frac{1}{R_N}(-v^{t\prime}_{tN}, v^{t\prime}_{tE}, v^{t\prime}_{tE}\tan\varphi^t)^T \tag{21}$$

3.3. *Equivalence proof of virtual sphere model algorithm and ellipsoid algorithm on the ellipsoid surface*

It is enough to prove that the calculation formulae for position updating between two arrangements are the same.

For the algorithm based on the virtual sphere model, substituting Equations (13), (18) and (19) into Equation (17) and combining Equation (22) yields

$$R_M = \frac{R_N(1-e^2)}{1-e^2\cos^2\varphi^t\cos^2\lambda^t} \tag{22}$$

The expression of virtual transverse velocity is calculated by

$$\begin{cases} v^{t\prime}_{tE} = \dfrac{V^t_{tE}[-2+e^2+e^2\cos(2\lambda^t)] - e^2V^t_{tN}\sin(2\lambda^t)\sin\varphi^t}{2(-1+e^2)} \\ v^{t\prime}_{tN} = \dfrac{(-4+3e^2)V^t_{tN} - e^2[V^t_{tN}\cos(2\lambda^t) - 2V^t_{tN}\cos^2\lambda^t\cos(2\varphi^t) + 2V^t_{tE}\sin(2\lambda^t)\sin\varphi^t]}{4(-1+e^2)} \end{cases} \tag{23}$$

Substituting Equation (23) into Equation (20) and making a simplification, the change rate of transverse longitude and latitude is shown to be completely equal to Equation (14).

3.4. *Equivalence proof of virtual sphere model algorithm and ellipsoid algorithm with height change*

The cases discussed above are all on the ellipsoidal surface. To expand the application scope of the algorithm, it is necessary to deduce a formula for when the height changes.

With altitude h, Equation (6) should be changed to

$$\begin{cases} x = (R_N+h)\cos\varphi^t\sin\lambda^t \\ y = (R_N+h)\sin\varphi^t \\ z = [R_N(1-e^2)+h]\cos\varphi^t\cos\lambda^t \end{cases} \tag{24}$$

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

and Equation (17) should be changed to

$$\boldsymbol{v}^{t\prime} = \boldsymbol{C}_g^t \begin{pmatrix} 1 & & \\ & \dfrac{R_N + h}{R_M + h} & \\ & & 1 \end{pmatrix} \boldsymbol{C}_t^g \boldsymbol{v}^t \tag{25}$$

In addition, calculation formulae such as Equation (16) should change R_N into $R_N + h$, while other calculation formulae will not be changed. Through software derivation, the transverse longitude and latitude change rate of the virtual sphere model algorithm and the ellipsoid model algorithm are derived to be equal.

So far, it is proved that the virtual sphere model algorithm is equivalent to the ellipsoid algorithm in transverse arrangement INS.

4. Conclusion and further discussion

Through analysing the principle of the INS ellipsoid transverse arrangement, this paper points out the root causes of the complexity of the ellipsoid transverse arrangement. Aiming at the newly proposed virtual sphere model algorithm, this paper briefly expounds the core idea and reveals the principle that it can avoid complex operation. The difference between the two algorithms is only the calculation involved in a position update, so the equivalence of the two algorithms is proved, which provides a sufficient theoretical foundation for the application of the virtual sphere model algorithm.

The following are points for further research.

1. The proof of the equivalence may broaden the method of solving the problems on INS update with the unconventional coordinate system, which is the path from a special geographic coordinate system (e.g. oblique coordinate systems and other pseudo coordinate systems) to a conventional geographic coordinate system, to a virtual sphere system, and finally back to the special coordinate system itself, and this is a certain contribution to the development of the inertial navigation algorithm if proved effective.
2. Some implicit relations between ellipsoid and sphere in mathematical analysis may be revealed by the proof, and potential practical value in other fields may be found.

Funding statement. This work was supported in part by the P.R.C. National Natural Science Foundation of China under Grant 41876222.

Competing interests. None.

References

Bian, H., Liu, W., Wen, C. and Wang, R. (2020). *Polar Navigation*. Beijing: Science Press.

Curtis, T. E. and Slater, J. M. (1959). Inertial navigation in submarine polar operation of 1958. *Navigation*, **6**(5), 275–283.

Ge, H., Xu, X., Huang, L. and Zhao, H. (2021). Grid SINS/GNSS integrated navigation algorithm based on the virtual sphere model in polar region. *Navigation Positioning and Timing*, **8**(06), 81–87.

Lin, X., Bian, H., Ma, H. and Wang, R. (2019). Applicability analysis of the approximate model of the earth with the arrangement of INS in polar region. *Acta Geodaetica et Cartographica Sinica*, **48**(3), 306–312.

Liu, C., Wu, W., Feng, G. and Wang, M. (2020). Polar navigation algorithm for INS based on virtual sphere n-vector. *Journal of Chinese Inertial Technology*, **28**(04), 421–428.

Qin, F., Chang, L., Tong, L., Wang, Z., Huang, C. and Zha, F. (2018). Transverse polar navigation method based on virtual sphere model. *Journal of Chinese Inertial Technology*, **26**(05), 571–578.

Xu, X. and Dou, M. (2014). Inertial navigation algorithm in polar regions based on transverse geographic coordinate system. *Journal of Huazhong University of Science and Technology (Natural Science Edition)*, **42**(12), 116–121.

Yao, Y., Xu, X., Li, Y., Liu, Y., Sun, J. and Tong, J. (2016). Transverse navigation under the ellipsoidal earth model and its performance in both polar and Non-polar areas. *The Journal of Navigation*, **69**, 335–352.

Appendix

A: Original code from Wolfram Mathematica

```
(*Equivalence proof of virtual sphere model algorithm and ellipsoid algorithm*)

(*Surface condition*)

(*Ellipsoid algorithm*)
Vt = {Vte, Vtn, VtU}; (*transverse velocity*)
re = {Rn Sin[λ] Cos[ϕ], Rn Sin[ϕ], Rn (1 - e^2) Cos[ϕ] Cos[λ]}; (*formula(6)in the paper*)
Cte = ( Cos[λ]    -Sin[λ] Sin[ϕ]    Cos[ϕ] Sin[λ]
        0          Cos[ϕ]            Sin[ϕ]
        -Sin[λ]   -Sin[ϕ] Cos[λ]    Cos[ϕ] Cos[λ] ); (*formula(8)in the paper*)
Rn = a / Sqrt[1 - e^2 Cos[ϕ]^2 Cos[λ]^2]; (*formula(13)in the paper*)
veE = Cte.{Vte, 0, 0};
veN = Cte.{0, Vtn, 0};
Edϕλ =
  FullSimplify[Solve[{veE[[1]] == D[re[[1]], ϕ] × x + D[re[[1]], λ] × y, veE[[2]] == D[re[[2]], ϕ] × x + D[re[[2]], λ] × y, veE[[3]] == D[re[[3]], ϕ] × x + D[re[[3]], λ] × y},
    {x, y}]]; (*solving formula(12.1)in the paper*)
Ndϕλ = FullSimplify[Solve[{veN[[1]] == D[re[[1]], ϕ] × x + D[re[[1]], λ] × y, veN[[2]] == D[re[[2]], ϕ] × x + D[re[[2]], λ] × y, veN[[3]] == D[re[[3]], ϕ] × x + D[re[[3]], λ] × y},
    {x, y}]]; (*solving formula(12.2)in the paper*)
Edϕ = x /. Edϕλ[[1]];
Edλ = y /. Edϕλ[[1]];
Ndϕ = x /. Ndϕλ[[1]];
Ndλ = y /. Ndϕλ[[1]];
dϕ = FullSimplify[Edϕ + Ndϕ] (*solving position updating formula of ellipsoid*) (*formula(14)derived*)
dλ = FullSimplify[Edλ + Ndλ] (*solving position updating formula of ellipsoid*) (*formula(14)derived*)

1/(2 a (-1 + e^2)) Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] (Vtn (-2 + e^2 + e^2 (Cos[λ]^2 Cos[2 ϕ] + Sin[λ]^2)) - 2 e^2 Vte Cos[λ] Sin[λ] Sin[ϕ])

(Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] (Vte (-2 + e^2 + e^2 Cos[2 λ]) Sec[ϕ] - 2 e^2 Vtn Cos[λ] Sin[λ] Tan[ϕ])) / (2 a (-1 + e^2))

(*Virtual sphere model algorithm*)
Sinσ = Sin[λ] / Sqrt[1 - Cos[ϕ]^2 Cos[λ]^2]; (*formula(19)in the paper*)
Cosσ = -Sin[ϕ] Cos[λ] / Sqrt[1 - Cos[ϕ]^2 Cos[λ]^2]; (*formula(19)in the paper*)
Cgt = ( Cosσ   -Sinσ   0
        Sinσ    Cosσ   0
        0       0      1 ); (*formula(18)in the paper*)
Rm = Rn (1 - e^2) / (1 - e^2 Cos[ϕ]^2 Cos[λ]^2); (*formula(22)in the paper*)
IVt = Simplify[Cgt.( 1  0      0
                     0  Rn/Rm  0
                     0  0      1 ).Transpose[Cgt].Vt] (*formula(17)in the paper*) (*formula(23)derived*)
Idϕ = FullSimplify[Part[IVt, 2] / Rn] (*formula(20)in the paper*)
Idλ = FullSimplify[Part[IVt, 1] Sec[ϕ] / Rn] (*formula(20)in the paper*)

{(e^2 Vte Cos[λ]^2 + Vte (-2 + e^2 - e^2 Sin[λ]^2) - 2 e^2 Vtn Cos[λ] Sin[λ] Sin[ϕ]) / (2 (-1 + e^2)),
 (-(-1 + e^2) Vtn Sin[λ]^2 - e^2 Vte Cos[λ] (1 - Cos[λ]^2 Cos[ϕ]^2) Sin[λ] Sin[ϕ] + Vtn Cos[λ]^2 (1 - e^2 Cos[λ]^2 Cos[ϕ]^2) Sin[ϕ]^2) / ((-1 + e^2) (-1 + Cos[λ]^2 Cos[ϕ]^2)), VtU}

-1/(a (-1 + e^2) (-1 + Cos[λ]^2 Cos[ϕ]^2))
  Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] ((-1 + e^2) Vtn Sin[λ]^2 + e^2 Vte Cos[λ] (-1 + Cos[λ]^2 Cos[ϕ]^2) Sin[λ] Sin[ϕ] + Vtn Cos[λ]^2 (-1 + e^2 Cos[λ]^2 Cos[ϕ]^2) Sin[ϕ]^2)

(Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] Sec[ϕ] (Vte (-2 + e^2 + e^2 Cos[2 λ]) - e^2 Vtn Sin[2 λ] Sin[ϕ])) / (2 a (-1 + e^2))

(*Equivalence proof of position update formula of two algorithms*)
FullSimplify[Idϕ == dϕ]
FullSimplify[Idλ == dλ]

True

True
```

https://doi.org/10.1017/S037346332200056X Published online by Cambridge University Press

```
(*Conditions with altitude change*)

(*Ellipsoid algorithm with altitude change considered*)
reh = {(Rn + h) Sin[λ] Cos[ϕ], (Rn + h) Sin[ϕ], (Rn (1 - e^2) + h) Cos[ϕ] Cos[λ]};(*formula(24)in the paper*)
Edϕλh =
  FullSimplify[Solve[{veE[[1]] == D[reh[[1]], ϕ] × x + D[reh[[1]], λ] × y, veE[[2]] == D[reh[[2]], ϕ] × x + D[reh[[2]], λ] × y,
     veE[[3]] == D[reh[[3]], ϕ] × x + D[reh[[3]], λ] × y}, {x, y}]];
Ndϕλh =
  FullSimplify[Solve[{veN[[1]] == D[reh[[1]], ϕ] × x + D[reh[[1]], λ] × y, veN[[2]] == D[reh[[2]], ϕ] × x + D[reh[[2]], λ] × y,
     veN[[3]] == D[reh[[3]], ϕ] × x + D[reh[[3]], λ] × y}, {x, y}]];
Edϕh = x /. Edϕλh[[1]];
Edλh = y /. Edϕλh[[1]];
Ndϕh = x /. Ndϕλh[[1]];
Ndλh = y /. Ndϕλh[[1]];
dϕh = FullSimplify[Edϕh + Ndϕh](*position update formula, transverse latitude*)
dλh = FullSimplify[Edλh + Ndλh](*position update formula, transverse longitude*)

(-2 e^4 h Vtn Cos[λ]^4 Cos[ϕ]^4 - 2 Vtn (h + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2]) +
   2 e^2 Vtn Cos[λ]^2 Cos[ϕ]^2 (2 h + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2]) + 2 a e^2 Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] Sin[λ] (Vtn Sin[λ] - Vte Cos[λ] Sin[ϕ])) /
 (2 a^2 (-1 + e^2) - 2 h^2 + h (-2 e^4 h Cos[λ]^4 Cos[ϕ]^4 + a (-4 + e^2 - e^2 Cos[2 λ]) Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] + e^2 Cos[λ]^2 (4 h Cos[ϕ]^2 + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] (3 + Cos[2 ϕ]))))

-(2 e^4 h Vte Cos[λ]^4 Cos[ϕ]^3 + 2 Vte (h + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2]) Sec[ϕ] -
   2 e^2 Vte Cos[λ]^2 (2 h Cos[ϕ] + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] Sec[ϕ]) + a e^2 Vtn Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] Sin[2 λ] Tan[ϕ]) /
 (2 a^2 (-1 + e^2) - 2 h^2 + h (-2 e^4 h Cos[λ]^4 Cos[ϕ]^4 + a (-4 + e^2 - e^2 Cos[2 λ]) Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] + e^2 Cos[λ]^2 (4 h Cos[ϕ]^2 + a Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2] (3 + Cos[2 ϕ]))))

(*Virtual sphere model algorithm with altitude change considered*)
IVth = Simplify[Cgt. {{1, 0, 0}, {0, (Rn+h)/(Rm+h), 0}, {0, 0, 1}}. Transpose[Cgt]. Vt];(*formula(25)in the paper*)
Idϕh = FullSimplify[Part[IVth, 2] / (Rn + h)](*position update formula, transverse latitude*)
Idλh = FullSimplify[Part[IVth, 1] Sec[ϕ] / (Rn + h)](*position update formula, transverse longitude*)

-(a Vtn Sin[λ]^2 - a e^2 Vtn Sin[λ]^2 + h Vtn (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2) Sin[λ]^2 + a e^2 Vte Cos[λ] Sin[λ] Sin[ϕ] -
   a e^2 Vte Cos[λ]^3 Cos[ϕ]^2 Sin[λ] Sin[ϕ] + a Vtn Cos[λ]^2 Sin[ϕ]^2 - a e^2 Vtn Cos[λ]^4 Cos[ϕ]^2 Sin[ϕ]^2 + h Vtn Cos[λ]^2 (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2) Sin[ϕ]^2) /
 ((-1 + Cos[λ]^2 Cos[ϕ]^2) (h + a / Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2]) (a - a e^2 + h (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2)))

-(Sec[ϕ] (a Vte Sin[λ]^2 - a e^2 Vte Cos[λ]^2 Cos[ϕ]^2 Sin[λ]^2 + h Vte (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2) Sin[λ]^2 + a e^2 Vtn Cos[λ] Sin[λ] Sin[ϕ] -
   a e^2 Vtn Cos[λ]^3 Cos[ϕ]^2 Sin[λ] Sin[ϕ] + a Vte Cos[λ]^2 Sin[ϕ]^2 - a e^2 Vte Cos[λ]^2 Sin[ϕ]^2 + h Vte Cos[λ]^2 (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2) Sin[ϕ]^2)) /
 ((-1 + Cos[λ]^2 Cos[ϕ]^2) (h + a / Sqrt[1 - e^2 Cos[λ]^2 Cos[ϕ]^2]) (a - a e^2 + h (1 - e^2 Cos[λ]^2 Cos[ϕ]^2)^(3/2)))

(*Equivalence proof of position update formula of two algorithms with altitude change considered*)
FullSimplify[Idϕh == dϕh]
FullSimplify[Idλh == dλh]

True

True
```

Created with Wolfram Mathematica 7.0

Cite this article: Wen Z, Bian H, Ma H, Wang R (2023). Analysis and Discussion on Ellipsoidal and virtual sphere models used for transverse arrangement INS. *The Journal of Navigation* **76**: 1, 103–112. https://doi.org/10.1017/S037346332200056X