Use of n-vector for Radar Applications

Nina Ødegaard, Kenneth Gade

Norwegian Defence Research Establishment
Kjeller, NORWAY
email: Nina.Odegaard@ffi.no
Kenneth.Gade@ffi.no

Abstract: *This paper aims to alert the radar community about the benefits of using n-vector for representing global horizontal position. Unlike other alternatives, n-vector position calculations are simple, vectorized and exact at all global positions. We show some examples of how to solve radar specific calculations using n-vector, and stress the necessity of unambiguous notation.*

1. Introduction

Practical applications in the radar field usually involve some form of position and/or orientation calculations. This can be e.g.

- calculation of the position of a radar detection based on range, azimuth and elevation to the target.
- sharing of target tracks between radars at different locations.
- calculation of the distance between two objects (both Euclidian and great circle distance).
- calculation of the intersection point of two target tracks.
- calculation of the relative displacements between positions along a synthetic aperture.
- determination of where a radar beam intersects a scene as the antenna position and orientation change.
- interpolation of position measurements.

The base position and orientation data can often come from an Inertial Navigation System (INS), which can consist of a Global Navigation Satellite System (GNSS), an Inertial Measurement Unit (IMU), and perhaps other sensors. The position and orientation data for different objects (like radar and target) often come from different sources and in different formats, and have to be combined somehow to calculate the desired quantities. These transformed position and orientation data are often used in the processing of the radar data. The quality of the radar

978-3-7369-9545-1 ©2018 DGON

product and the subsequent exploitation of the radar data thus depend on the accuracy of the position and orientation data.

Global horizontal position can be represented in several ways, like latitude and longitude, a local Cartesian coordinate frame or various map projections. Global position can also be represented by the vector from the Earth center, decomposed in the Earth Centered Earth Fixed coordinate frame (ECEF). Each of these formats has their own limitations. With the ECEF vector, it is not possible to treat the horizontal and vertical positions independently, as is often desired [1]. A local Cartesian coordinate frame uses a flat Earth approximation, and is only accurate in a small area. Map projections have distortions due to the projection chosen, and calculations are very complex when crossing zone boundaries, as in UTM [2]. Latitude and longitude have singularities at the poles and a discontinuity at the $\pm 180^{\text{o}}$ longitude meridian. An error in latitude leads to an error in longitude, as these errors are coupled. Code that uses these formats in calculations can often be complicated, with if statements (because different positions need to be treated differently), contains approximations in the equations, and sometimes uses iterations to arrive at a solution. Several of the errors mentioned above increase when getting close to the poles. As marine traffic increase in the Arctic region [3], it can be expected that a lot of code that was never intended for use in the polar regions will not have been replaced on various navigation systems, radars and other sensors on board ships. This can lead to significant errors in calculations, as we will see in this paper.

n-vector was introduced to the navigation community by [4] in 2010. By using n-vector instead of other alternatives to represent global horizontal position, the code is often exact and simple, and it works equally well for all Earth positions. The aim of this paper is to inform the radar community of this elegant and exact method for global position calculations. We will show some examples of how n-vector can be used for radar specific calculations.

A problem often encountered in practical applications of radar data is that providers of position and orientation data do not give good enough documentation on the format of the data, or which transformations have been applied to them. When position and orientation data, or computer code for position and orientation calculations, are to be shared with other users, we will stress the importance of precise and unambiguous notation. The notation from Chapter 2 in [5] will be used here.

2. n-vector

n-vector is the normal vector to a reference ellipsoid, as shown in Fig. 1. It is used to replace latitude and longitude as a representation of global horizontal position. As the ellipsoid is a strictly convex and differentiable surface, any 2D position on it can be uniquely represented by the outward pointing normal vector at that point, [4]. By using the height/depth over/under the nearest point on the ellipsoid, 3D positions can be represented. n-vector is non-singular and it also allows many position calculations to be solved with simple vector algebra. More information can be found in [6], where also a code library can be downloaded.

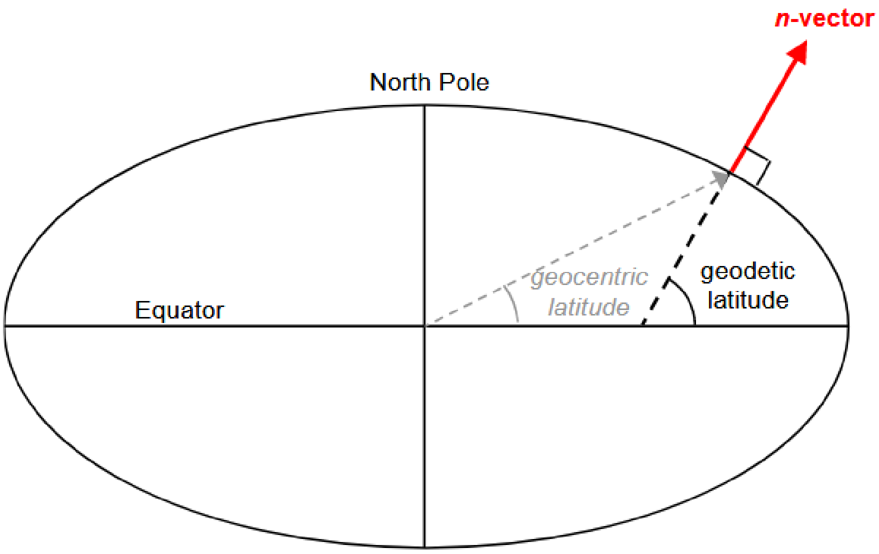

Figure 1: n-vector is the normal vector to the reference ellipsoid, and it corresponds to geodetic latitude

3. Notation

Precise and unambiguous notation is very important in all calculations involving several coordinate frames. It makes it easier to read and implement code, it will also most likely prevent many errors that would otherwise not be spotted. A subset of the notation from Chapter 2 in [5] will be presented here (only the parts that are being used in this paper).

A general position vector $\boldsymbol{p}_{AB}^{C}$ is a vector that goes from the origin of frame A to the origin of frame B. It is decomposed in the coordinate frame C, i.e. its three components are along the axes of C.

A rotation matrix $\boldsymbol{R}_{AB}$ is a direction cosine matrix (DCM) describing the orientation of frame B relative to frame A. For a rotation matrix we have that $\boldsymbol{R}_{AB} = (\boldsymbol{R}_{BA})^T$ where T means matrix transpose.

A couple of rules from Chapter 2 in [5] will be used in this paper. The “rule of closest frames” is used when transforming a vector:

$$\boldsymbol{p}^A = \boldsymbol{R}_{AB}\, \boldsymbol{p}^B \tag{1}$$

The rule determines the order of subscripts to use on $\boldsymbol{R}$ (either $\boldsymbol{R}_{AB}$ or $\boldsymbol{R}_{BA}$), and it says that B should be closest to B (hence $\boldsymbol{R}_{AB}$ is correct here).

The second rule we will use is the “cancellation of intermediate frames” as

$$\boldsymbol{R}_{AC} = \boldsymbol{R}_{AB}\, \boldsymbol{R}_{BC} \tag{2}$$

where frame B is canceled, and we are left with the two outermost frames.

The frames mentioned so far are arbitrary, while two frames will always have a specific meaning in this paper. The E frame is the ECEF frame with origin in the center of the Earth. The N frame is north-east-down and has origin directly above or below a body/object, at Earth's surface (surface of ellipsoid model).

4. Examples

This chapter will show some examples of how n-vector can be used in common radar applications. We will see how the n-vector solution is often simple and exact at all Earth locations, and that good notation makes the procedures more intuitive.

4.1. Example 1

A common step in many radar applications is to calculate Euclidian distances in north and east (and possibly down) directions from positions given in latitude, longitude and height (located within a limited area). Two positions, A and B (i.e. the origins of coordinate frames A and B), are given as lat_{EA}, lon_{EA} and lat_{EB}, lon_{EB} where E means relative to Earth. The horizontal displacements between them can be found approximately in the classical way as:

$$\begin{aligned} \Delta_{north} &= (lat_{EA} - lat_{EB})\, r_{roc,meridian} \\ \Delta_{east} &= (lon_{EA} - lon_{EB})\ \cos(lat_{mean})\, r_{roc,transverse} \end{aligned} \tag{3}$$

where $r_{roc,meridian}$ and $r_{roc,transverse}$ are the radii of curvature of the Earth, either modelled as spherical Earth or an ellipsoid. For a spherical Earth the two values are the same (approximately 6371 km). Reference [7] gives the formulas of $r_{roc,meridian}$ and $r_{roc,transverse}$ as

$$\begin{aligned} r_{roc,meridian} &= \frac{a(1-e^2)}{(1-e^2\ \sin^2(lat_{mean}))^{3/2}} \\ r_{roc,transverse} &= \frac{a}{\sqrt{1-e^2\ \sin^2(lat_{mean})}} \end{aligned} \tag{4}$$

where a is the semi-major axis of the ellipsoid, b is the semi-minor axis and $e = \sqrt{1 - \frac{b^2}{a^2}}$.

The displacements in (3) are often used to calculate the Euclidian distance between the positions as the square root of the sum of the components squared, possibly with the height component included. The distance found with this method is not exact, and the error increases with latitude. We will compare this solution to the exact distance found using n-vector.

We will now find the exact displacements and the exact distance. The positions first have to be converted from latitude/longitude to n-vectors (the text in Courier font in the following equations is MatLab code, which can be downloaded at the n-vector website [6]):

$$\begin{aligned} \boldsymbol{n}_{EA}^{E} &= \texttt{lat\_lon2n\_E}(lat_{EA}, lon_{EA}) \\ \boldsymbol{n}_{EB}^{E} &= \texttt{lat\_lon2n\_E}(lat_{EB}, lon_{EB}) \end{aligned} \tag{5}$$

Then the delta vector decomposed in E is computed as:

$$\boldsymbol{p}_{AB}^{E} = \texttt{n\_EA\_E\_and\_n\_EB\_E2p\_AB\_E}(\boldsymbol{n}_{EA}^{E}, \boldsymbol{n}_{EB}^{E}, z_{EA}, z_{EB}) \tag{6}$$

where z_{EA} and z_{EB} are the depths (negative heights) of the two positions.

The Euclidian distance between the two positions can be found as

$$d_{AB} = \texttt{norm}(\boldsymbol{p}_{AB}^{E}) \tag{7}$$

To find the north and east components we need to decompose the vector in N instead of E. A rotation matrix will apply the transformation. The rotation matrix can be found from the n-vector (at position A) as:

$$\boldsymbol{R}_{EN} = \texttt{n\_E2R\_EN}(\boldsymbol{n}_{EA}^{E}) \tag{8}$$

The delta vector can now be decomposed in N as follows:

$$\boldsymbol{p}_{AB}^{N} = \boldsymbol{R}_{NE}\, \boldsymbol{p}_{AB}^{E} \tag{9}$$

Note how the “rule of closest frames” specifies the order of the subscripts on $\boldsymbol{R}$ (E is closest to E).

The three components of $\boldsymbol{p}_{AB}^{N}$ are now the north, east and down displacements between positions A and B, in the N system defined at position A. Note that the N system is singular at the poles, since the north and east directions are undefined there. Therefore the north, east and down displacements are singular. However, the Euclidian distance found in (7) is non-singular, since the N system is not involved in the calculations.

To illustrate the difference between the classical method and the method in (7), pairs of positions are placed 200 m apart in the north direction and 100 m apart in the east direction, on different latitudes. The Euclidian distance for the classical approach is calculated by taking the square root of the sum of the components squared (north-east-down). Fig. 2 shows the differences in Euclidian distance between using the classical approach and the exact method based on n-vector. Fig. 3 shows the same, but zoomed in to show what happens when we are further away from the pole. For n-vector the WGS84 ellipsoid is used in both plots, and for latitude/longitude WGS84 is used in the left plot and spherical Earth is used in the right plot. The difference crosses zero at around 48° latitude in the right plots, because the radius of curvature of the spherical approximation and the radius of curvature of the ellipsoid are equal at that latitude.

4.2. Example 2

Assuming spherical Earth, we sometimes wish to find the great circle distance between two positions instead of the Euclidian distance. The great circle distance is the shortest path following

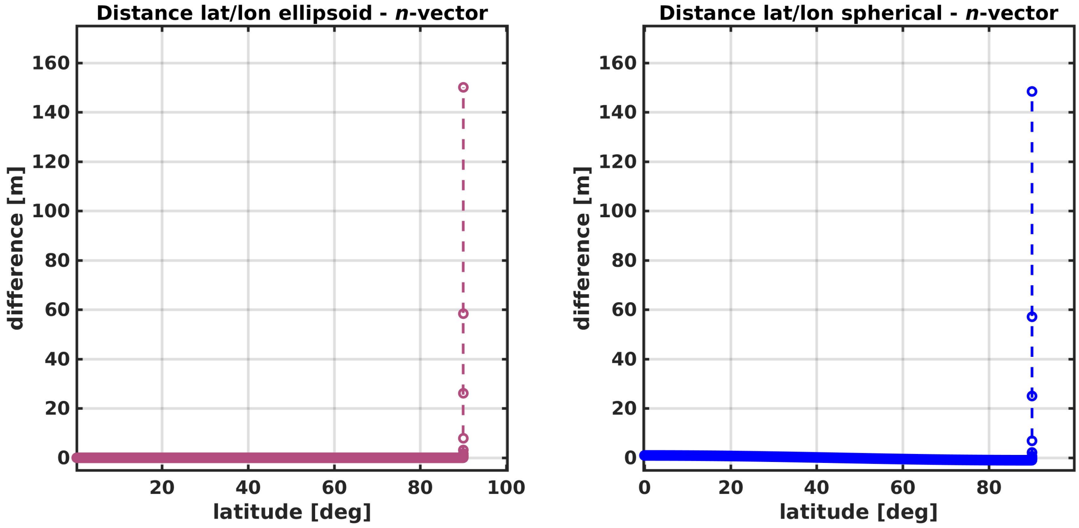

Figure 2: Differences in calculated Euclidian distance between using the classical approach and the exact method based on n-vector. For n-vector the WGS84 ellipsoid is used in both plots. For the classical approach the WGS84 ellipsoid is used in the plot to the left, and spherical Earth in the plot to the right. Pairs of positions are placed 200 m apart in the north direction and 100 m apart in the east direction, on different latitudes (with a step size of 0.001°).

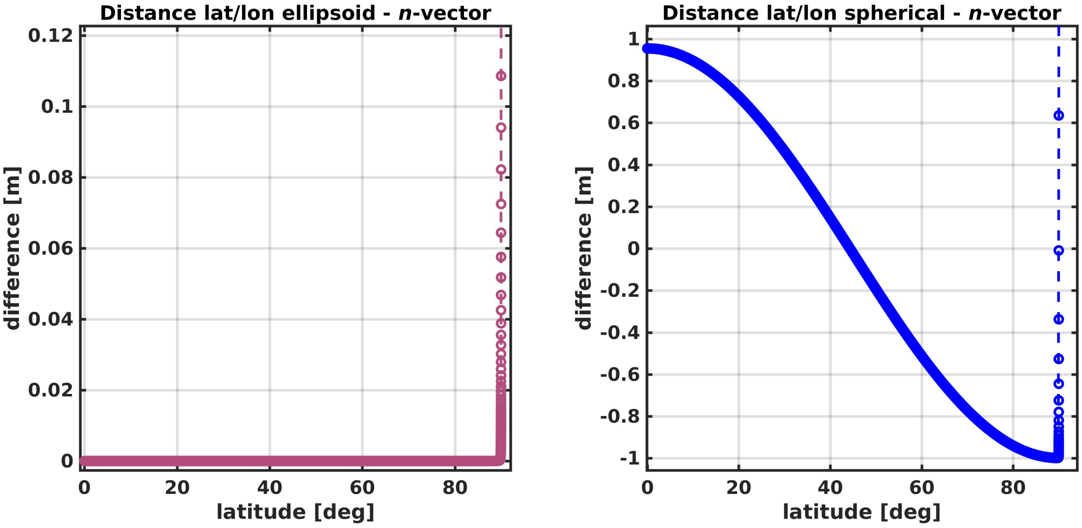

Figure 3: Zoomed version of Fig. 2. Here we can see that for the classical approach using spherical Earth we have considerable errors in the calculated distance also on lower latitudes.

the surface of the sphere. This can be when the two objects of interest are ships, or for example in conditions forming evaporation ducts. The great circle distance can be found easily using n-vector:

$$s_{AB} = \texttt{atan2}\left(\left|\boldsymbol{n}_{EA}^{E} \times \boldsymbol{n}_{EB}^{E}\right|, \boldsymbol{n}_{EA}^{E} \cdot \boldsymbol{n}_{EB}^{E}\right) \cdot r_{Earth} \tag{10}$$

where r_{Earth} is the radius of the Earth.

4.3. Example 3

Assume a ship based radar with an antenna that can scan in both azimuth and elevation, either mechanically or electronically. The position of the radar antenna A is given as $\boldsymbol{n}_{EA}^{E}$ and z_{EA}, and the orientation as $\boldsymbol{R}_{NA}$. $\boldsymbol{R}_{NA}$ can be found from roll, pitch and yaw using the MatLab function `zyx2R.m`, available at [6]. Assume that a target detection is done (B) and we have a range, an azimuth and an elevation angle. These can be combined to form the vector $\boldsymbol{p}_{AB}^{A}$ (the delta vector going from A to B, decomposed in A). We wish to find the exact position of the target detection using n-vector.

The delta vector, $\boldsymbol{p}_{AB}^{A}$, must first be decomposed in E. For that we need to find $\boldsymbol{R}_{EA}$:

$$\boldsymbol{R}_{EA} = \boldsymbol{R}_{EN}\,\boldsymbol{R}_{NA} \tag{11}$$

$\boldsymbol{R}_{EN}$ can be found as in (8).

The delta vector is now decomposed in E:

$$\boldsymbol{p}_{AB}^{E} = \boldsymbol{R}_{EA}\,\boldsymbol{p}_{AB}^{A} \tag{12}$$

It is now easy to find the exact position of target detection B as:

$$[\boldsymbol{n}_{EB}^{E}, z_{EB}] = \texttt{n\_EA\_E\_and\_p\_AB\_E2n\_EB\_E}(\boldsymbol{n}_{EA}^{E}, \boldsymbol{p}_{AB}^{E}, z_{EA}) \tag{13}$$

The above calculations are exact at all Earth locations, as well as easy to implement using the code library. The WGS84 ellipsoid is default in (13), but other ellipsoids or spherical Earth can be chosen by including additional input parameters.

Imagine now that detections made on several radars are to be combined into a situation picture. It is important to be aware of the potential problems with using displacements in north and east for reporting the detections/tracks. If the radars are far apart, or if they are close to the poles, the N frames of the radars will have a significant angular difference, as illustrated in Fig. 4. The north and east displacements of a delta vector will be highly position dependent, and each radar will have its own north and east directions.

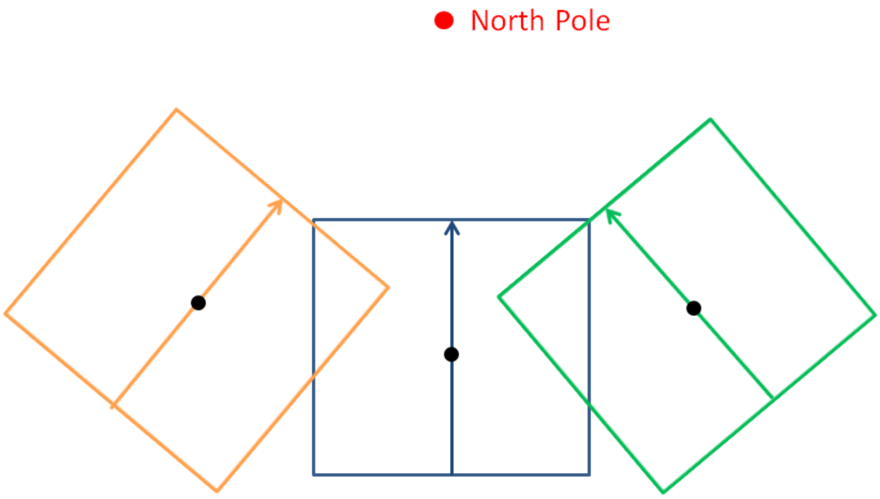

Figure 4: Illustration of how the north direction can vary when close to the poles. The figure shows the north - east planes of three positions. The vector pointing north varies from position to position, and displacements found in one position will not correspond to the same target position when viewed from another position.

4.4. Example 4

Assume that tracks from two detected ships have been established. We want to find out where the two great circle paths intersect (spherical Earth assumed). This is a very complicated process using latitude and longitude [8]. However, when using n-vector, the task can be done in two lines of code.

We represent each great circle by its normal vector. The normal vector is found by taking the cross product of two n-vectors defining the great circle. Any two n-vectors lying on the path can be used. Having the normal vectors to both paths, the intersection is now simply found by taking the cross product of the two normal vectors:

$$\boldsymbol{n}_{EC}^{E} = \pm\texttt{unit}\Big(\big(\boldsymbol{n}_{EA,1}^{E} \times \boldsymbol{n}_{EA,2}^{E}\big) \times \big(\boldsymbol{n}_{EB,1}^{E} \times \boldsymbol{n}_{EB,2}^{E}\big)\Big) \tag{14}$$

Here C is the intersection point, $\boldsymbol{n}_{EA,1}$ and $\boldsymbol{n}_{EA,2}$ are two n-vectors defining path A and $\boldsymbol{n}_{EB,1}$ and $\boldsymbol{n}_{EB,2}$ are two n-vectors defining path B. Note that the great circles will intersect at opposite sides of the Earth (antipodal positions), and thus two solutions are found. Selecting the solution that is closest to e.g. $\boldsymbol{n}_{EA,1}^{E}$ can be achieved by selecting the solution that has a positive dot product with $\boldsymbol{n}_{EA,1}^{E}$. In program code, this can be implemented by selecting any of the two solutions above (+ or -), called C', and calculating:

$$\boldsymbol{n}_{EC}^{E} = \texttt{sign}\big(\boldsymbol{n}_{EC'}^{E} \cdot \boldsymbol{n}_{EA,1}^{E}\big)\boldsymbol{n}_{EC'}^{E} \tag{15}$$

4.5. Example 5

Difficulties often arise when position and orientation data are to be shared among several users, and also when data from several sensors are to be combined in some way. One issue is that the data often come with different sampling rates, e.g. a GNSS and a radar will normally not have the same rate. Interpolation is usually necessary. If standard interpolation is used directly on latitude/longitude the interpolated positions will lie along a curved line. The curvature will increase when getting closer to the poles, and the interpolation will not work across the $\pm 180^{\mathrm{o}}$ longitude meridian. However, when using n-vector, standard interpolation can be used directly. We have the positions of B at times t_0 and t_1, given as $\boldsymbol{n}^E_{EB}(t_0)$ and $\boldsymbol{n}^E_{EB}(t_1)$. The interpolated position at time t_i can be found exactly as

$$\boldsymbol{n}^E_{EB}(t_i) = \texttt{unit}\left(\boldsymbol{n}^E_{EB}(t_0) + (t_i - t_0)\frac{\boldsymbol{n}^E_{EB}(t_1) - \boldsymbol{n}^E_{EB}(t_0)}{t_1 - t_0}\right) \tag{16}$$

5. Conclusion

This paper has showed some of the benefits of using n-vector for calculations involving global horizontal position. To introduce the concept of n-vector to the radar community, some examples of common calculations in radar applications have been presented. We have seen that using n-vector often makes the code simple and vectorized, and the results are valid at all Earth positions, without exceptions or limited accuracy, for both short and long distances. Unambiguous notation is also essential in order to apply the correct transformations to the correct quantities.

References

[1] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[2] J. P. Snyder, “Map projections – a working manual,” U.S. Geological survey professional paper 13950, United States Government Printing Office, 1987.

[3] “Arctic Marine Shipping Assessment 2009 Report,” AMSA Report, Arctic Council, 2009.

[4] K. Gade, “A non-singular horizontal pozition representation,” *The Journal of Navigation*, vol. 63, pp. 395–417, 2010.

[5] K. Gade, *Inertial Navigation – Theory and Applications*. PhD thesis, Norwegian University of Science and Technology (NTNU), January 2018.

[6] K. Gade. The n-vector website. `www.navlab.net/nvector`, Accessed 21 March 2018.

[7] J. R. Clynch, “Radius of the Earth - Radii used in geodesy,” tech. rep., Naval Postgraduate School, Monterey, California, USA, 2002.

[8] The Movable Type Ltd website. `www.movable-type.co.uk`, Accessed 21 March 2018.