1090-WP29-07
July 21-24, 2009

<u>RTCA Special Committee 186, Working Group 3</u>

<u>ADS-B 1090 MOPS, Revision B</u>

<u>Meeting #29</u>

An Expanded Description of the CPR Algorithm

Prepared by Al Marshall
Sensis Corporation

SUMMARY
This paper is intended to become an informative Appendix on the CPR encoding and decoding algorithms. The algorithms are described with diagrams, text and equations in a manner that is easier to understand than the necessarily terse description in Appendix A. No new requirements or test procedures are proposed.

1 Purpose

The purpose of this appendix is to provide explanations and derivations of the Compact Position Reporting (CPR) equations given in Appendix A. CPR encoding and decoding can be implemented without using any of the information in this appendix but, if a CPR implementation has problems, the information in this appendix may be helpful in resolving the issue.

2 Motivation for Compact Position Reporting

CPR) was developed for ADS-B messages broadcast on the 1090 MHz Extended Squitter (ES) datalink to reduce the number of bits required to convey participant latitude and longitude. Position resolution for ES messages is approximately 5.1 meters for airborne participants and 1.3 meters for surface participants (see Section 3). The circumference of the earth is approximately 40 000 kilometers so 40 000 000 m/5.1 m = ~7 800 000 discrete position values. 7 800 000 position values would require 23 bits in a message. Longitude is expressed over a range of 360° so longitude would require the full 23 bits. Latitude is expressed over a range of 180° so only 3 900 000 discrete position values or 22 bits would be required. Following similar reasoning, surface position would require 25 bits for longitude and 24 bits for latitude. CPR conveys position with 17 bits each for latitude and longitude plus 1 “CPR format” bit.

Table 2-1 Message bits required for position encoding with and without CPR

		Without CPR	With CPR	Bits Saved with CPR
Airborne Position	Latitude	22	17	
	Longitude	23	17	
	CPR Fmt	0	1	
	Total	45	35	10
Surface Position	Latitude	24	17	
	Longitude	25	17	
	CPR Fmt	0	1	
	Total	49	35	14

CPR saves 10 bits per position message for airborne participants and 14 bits per position message for surface participants. Position messages are broadcast twice per second under most conditions so CPR saves 20 bits/second for airborne participants and 28 bits/second for surface participants. The maximum message transmission rate allowed for each 1090ES ADS-B participant is 6.2 messages/sec. Of the 112 bits in each ES message, only 56 are available for the ADS-B payload. The first 5 ADS-B payload bits are reserved for the message type so 51 bits from each message are available for data. 51 bits/message multiplied by 6.2 messages/sec is 316 bits/second. CPR saves 6% - 9% of the available bits for other uses.

3 CPR Coordinate System

CPR can be thought of as a coordinate system and a set of coordinate transformation algorithms. The coordinate system is spherical and comparable to the latitude/longitude reference system commonly used for navigation. The coordinate transformation algorithms convert between latitude/longitude and CPR coordinates. The coordinate transformation algorithms are commonly called “CPR encoding” and “CPR decoding.”

In the CPR coordinate system, the globe is divided into zones. “Latitude zones” start at the equator and go to both poles. “Longitude zones” start at the Prime Meridian and proceed eastward around the globe. Latitude zones are approximately 360 nautical miles (nmi) in height measured in the north-south direction. Longitude zones are approximately 360 nmi in width measured in the east-west direction. The number of longitude zones is reduced as one moves from the equator to the poles to maintain approximately constant zone width.

There are two sets of slightly different sized zones in both latitude and longitude. One set of zones is called “even” and the other is called “odd.” Figure 3-1 and Figure 3-2 depict latitude and longitude zones on a global scale where the red lines represent even zone boundaries and the blue lines represent odd.

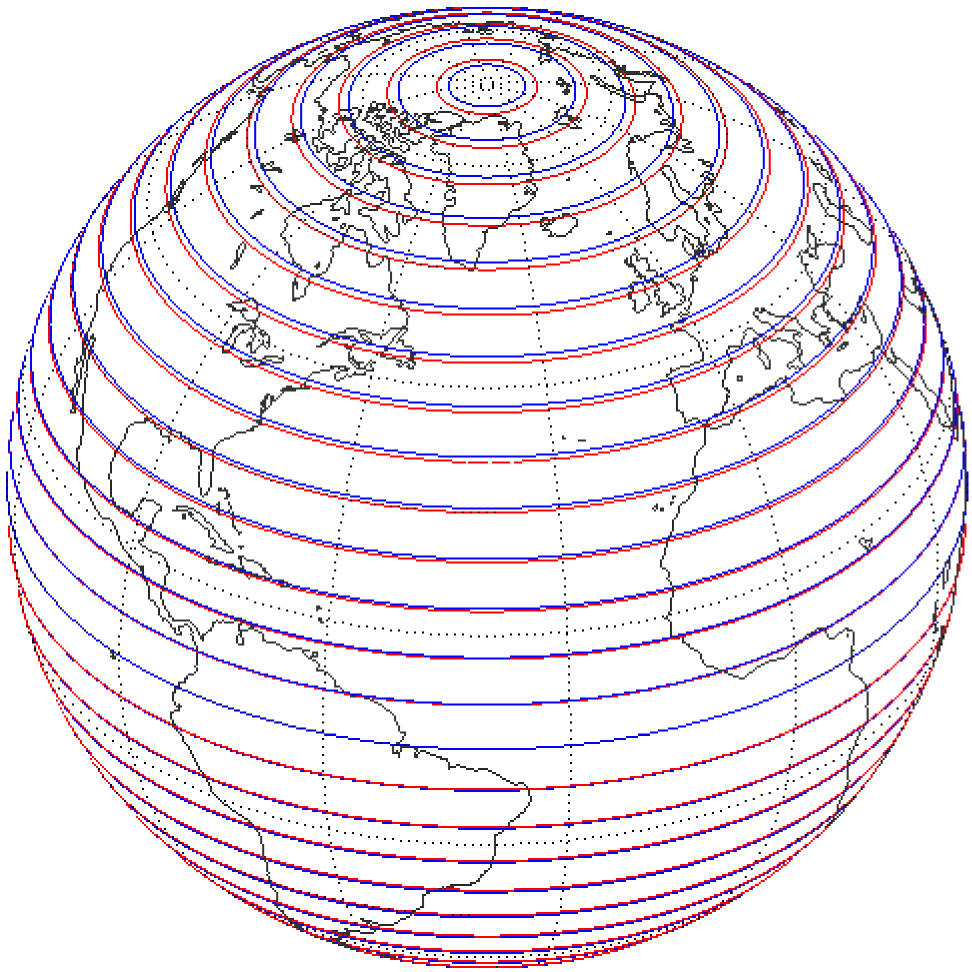


Figure 3-1 Latitude Zone Boundaries (Red is Even; Blue is Odd)

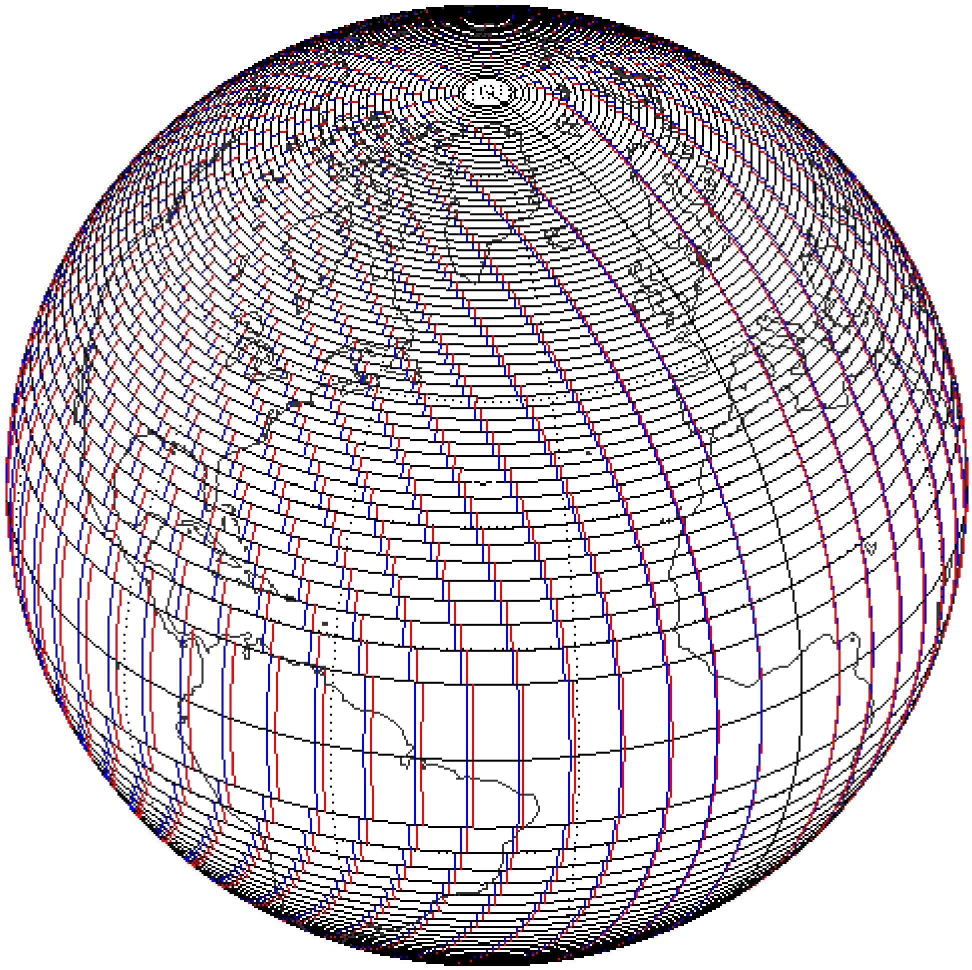


Figure 3-2 Longitude Zone Boundaries (Red is Even; Blue is Odd)

Each zone is identified with a zone index number. Imagine a tailor's tape measure marked with even and odd latitude zones as shown in the left side of Figure 3-3. The tape measure has 4*NZ* = 60 even zones and 4*NZ*-1 = 59 odd zones numbered beginning with zone index 0. The zone boundaries will line up at the beginning and end of the tape measure but nowhere else. The tape measure is wrapped around the globe starting with zone index 0 extending north from the equator. The tape passes over the north and south poles and returns to the equator as shown on the right side of Figure 3-3.

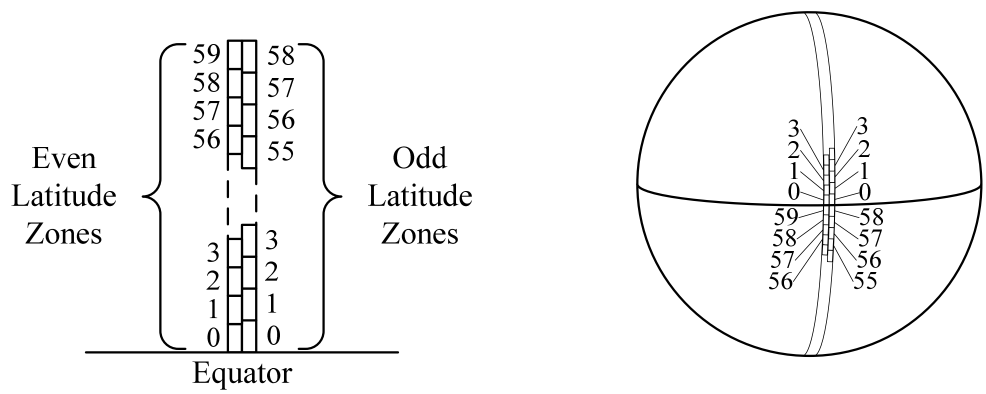


Figure 3-3 Latitude Zone Indexing

Longitude zone indexing is the same as latitude zone indexing except zone index 0 is just east of the Prime Meridian and the zone numbers increase in the eastward direction. As mentioned above, the number of longitude zones around the globe varies with latitude and this is reflected in the zone index numbers.

Latitude and longitude zones are divided into bins. Each zone contains 2^{Nb} equal sized bins. The value of *Nb* varies with the type of position being encoded. Bins are numbered starting with 0 at the southern edge of latitude zones and the western edge of longitude zones. More details on sizing are provided in the following sections.

Every point on the globe is identified in the CPR coordinate system with a latitude zone index, latitude bin number, longitude zone index, longitude bin number and CPR format (even or odd zone size).

3.1 Latitude Zones

Latitude zones are symmetrical about the equator. The height of a latitude zone is called $Dlat_i$. The subscript i has a value of 0 to indicate the height of an even zone or a value of 1 to indicate the height of an odd zone. $Dlat_i$ is defined in terms of the constant NZ. NZ is the number of even latitude zones in one quadrant of the globe and has a value of 15. The latitude zone size, $Dlat_i$, is determined with Eq 1.

$$Dlat_i = \frac{360°}{4NZ - i} \qquad \text{Eq 1}$$

$NZ = 15$ [A.1.7.3 a]
$i = 0$ for even encoding
$i = 1$ for odd encoding

Since there are 4 quadrants around the globe, there are 60 even latitude zones and 59 odd zones. Even latitude zones are approximately 360 nautical miles (nmi) high and odd latitude zones are approximately 366 nmi high measured in the north-south direction.

The definition of $Dlat_i$ is depicted in Figure 3-4.

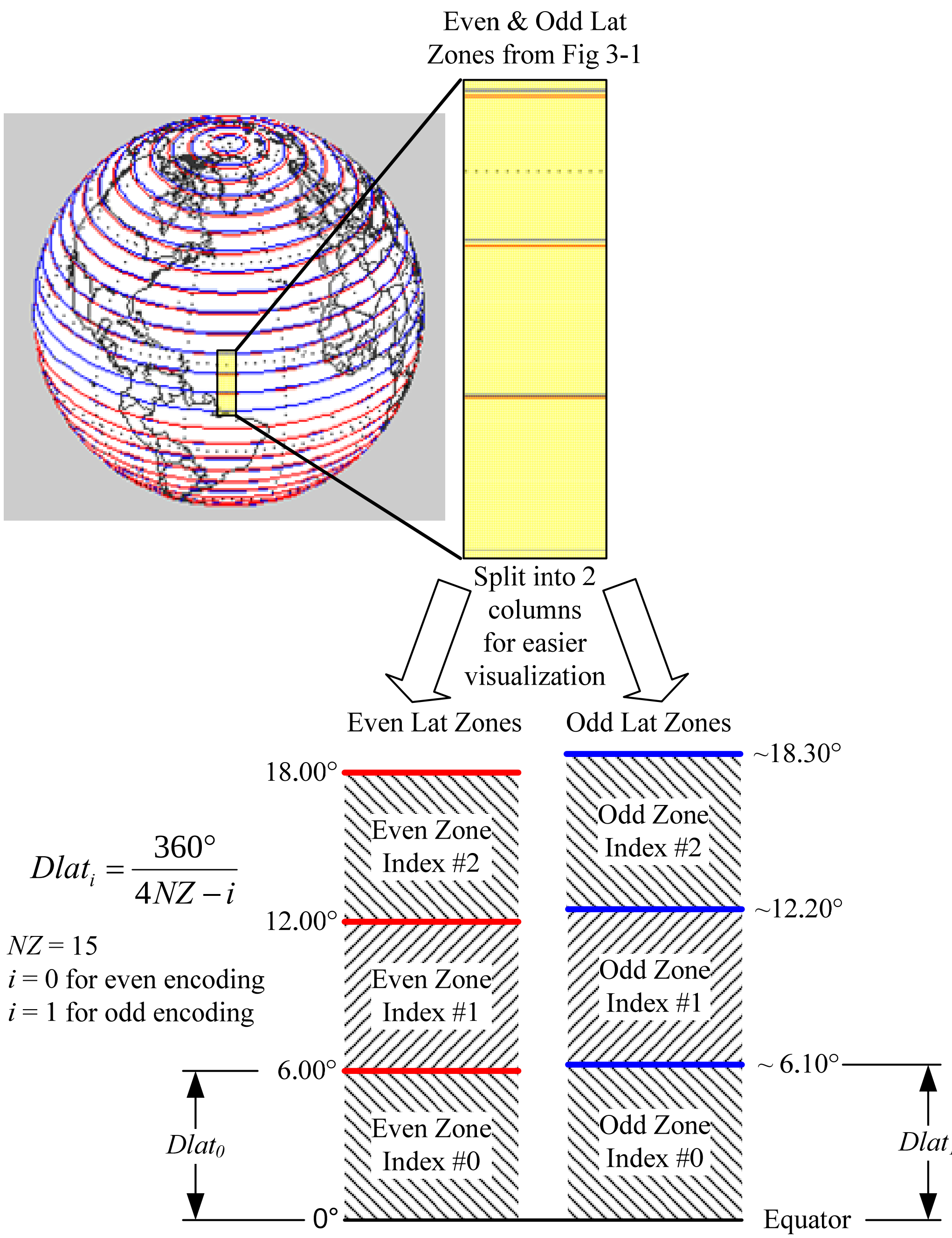


Figure 3-4 Definition of Latitude Zone Sizes

Latitude zones are divided into bins. There are 2^{Nb} bins in each latitude zone. Each latitude bin is identified by the integer bin number YZ_i. YZ_i is the bin number relative to the southern edge of the latitude zone (in both the northern and southern hemispheres). The subscript i indicates whether the value of YZ is relative to an even zone (i=0) or an odd zone (i=1). The CPR decoding algorithms produce a latitude value (a real number) from YZ_i which is the latitude at the centerline of the bin. This value is called the bin centerline latitude.

Figure 3-5 shows the relationship between latitude bins and zones.

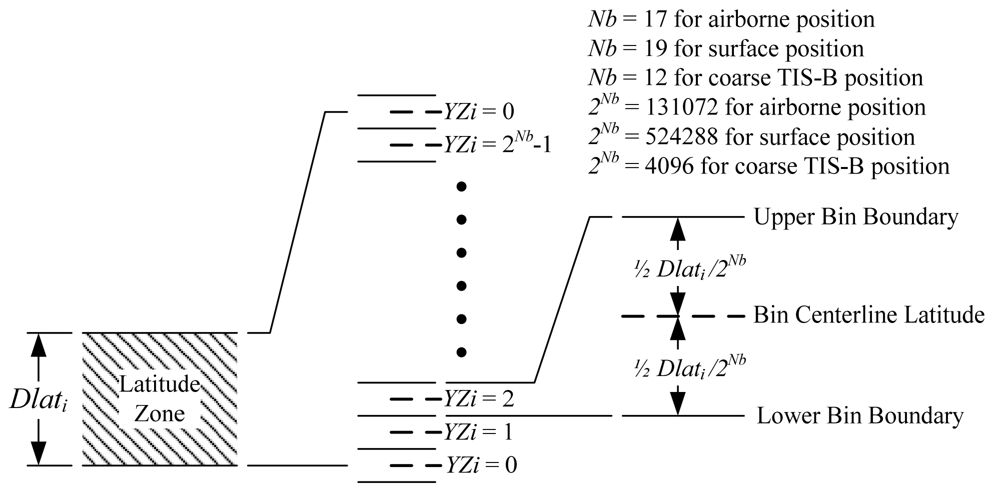


Figure 3-5 Latitude Zones and Bins

The bin height, which is equal to the encoding resolution, is the latitude zone height divided by the number of bins per zone. Section §A.1.7.2 defines the number of bits used for encoding to be 17 for airborne position, 19 for surface position and 12 for Coarse TIS-B position. The resulting bin sizes are shown in Table 3-1.

Table 3-1 - CPR Latitude Encoding Resolution

Position Type	Nb	bins/zone	Zone size (degrees)		Bin size (degrees)		Bin Size (meters)	
			Even	Odd	Even	Odd	Even	Odd
Airborne	17	131072	6.000	6.102	4.6E-05	4.7E-05	5.10	5.18
Surface	19	524288	6.000	6.102	1.1E-05	1.2E-05	1.27	1.30
Coarse TIS-B	12	4096	6.000	6.102	0.00146	0.00149	163.07	165.83

The CPR encoding algorithm transforms latitude into even and odd bin numbers. Figure 3-6 is an example showing even and odd bin numbers for airborne position at 43.054°N. Red lines delimit the even zone and blue lines delimit the odd zone. The point of interest, colored magenta, is in even bin number 23035 and odd bin number 7349.

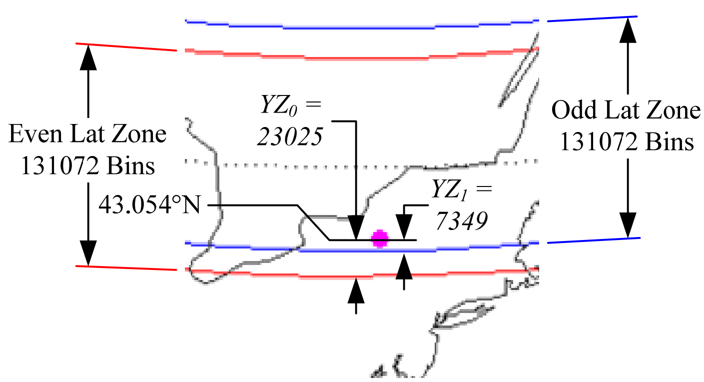


Figure 3-6 YZ for Even and Odd Latitude Zones at 43.054N

Although not produced by the encoding algorithm, the zone index number for both the even and odd zones at the latitude in Figure 3-6 is 7.

3.2 Longitude Zones and Bins

Longitude zones and bins are similar to latitude zones and bins. For a given position type (i.e. airborne, surface, coarse TIS-B), the number of bins per longitude zone is equal to the number of bins per latitude zone. The main difference between latitude and longitude zones is that the number of longitude zones circling the globe is a function of latitude. The CPR coordinate system keeps the encoding resolution (i.e. bin width in meters) as close to a constant as possible everywhere on the globe. Since the number of bins per zone is fixed, the zone width (in nmi) must also remain close to constant. The distance around the globe along lines of constant latitude is smaller near the poles than near the equator. The CPR algorithm keeps the longitude bin and zone width approximately constant by reducing the number of zones as latitude increases.

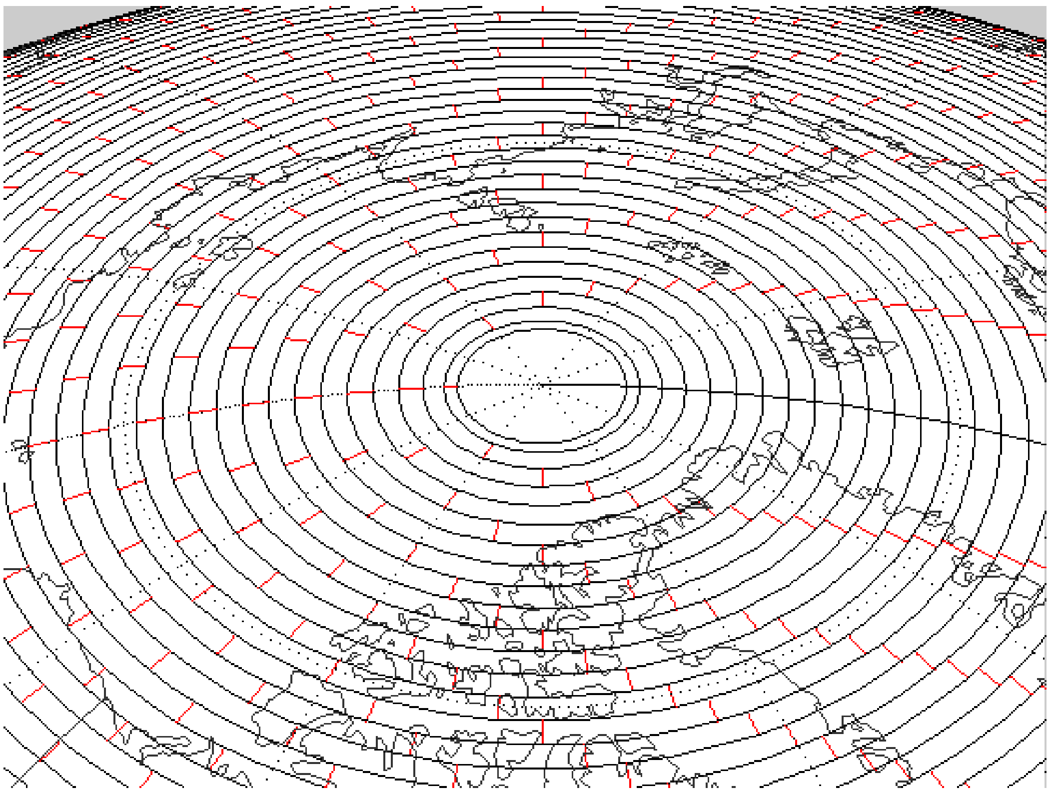


Figure 3-7 Reduction in Number of Longitude Zones at High Latitudes

Note that the linear (not angular) widths of the longitude zones in Figure 3-7 are all approximately equal and that the number of zones around the globe decreases as the magnitude of latitude increases. The number of even longitude zones at a latitude is called *NL*. The latitudes at which the number of longitude zones changes are called "NL Transition Latitudes." These are represented by black lines of constant latitude in Figure 3-2 and Figure 3-7. NL Transition Latitudes generally do not coincide with latitude zone boundaries.

For all CPR computations, NL is calculated using the "recovered latitude" $Rlat_i$. $Rlat_i$ is obtained by converting a bin number to a bin centerline latitude. The rationale for this is discussed in Section 4. The equation for the number of longitude zones (*NL*) as a function of latitude is specified in Eq 2:

For $Rlat_i = 0°$ (the Equator), $NL = 59$
For $Rlat_i = +87°$ or $lat = -87°$, $NL = 2$
For $Rlat_i > +87°$ or $lat < -87°$, $NL = 1$
For all other latitudes ($-87° < Rlat_i < 0°$ and $0° < Rlat_i < +87°$):

$$NL(Rlat_i) = floor\left(2\pi \left[\arccos\left(1 - \frac{1-\cos\left(\frac{\pi}{2NZ}\right)}{\cos^2\left(\frac{\pi}{180°} \bullet \left|Rlat_i\right|\right)} \right) \right]^{-1} \right)$$

Eq 2

[A.1.7.2.d]

Note in Eq 2 that NL is inversely proportional to the magnitude of the recovered latitude and that the result of the computation on the right hand side is quantized by the floor function. As the magnitude of the recovered latitude increases (i.e. from the equator to the poles), NL changes *after* the recovered latitude crosses the NL Transition Latitude. This is illustrated in Figure 3-8.

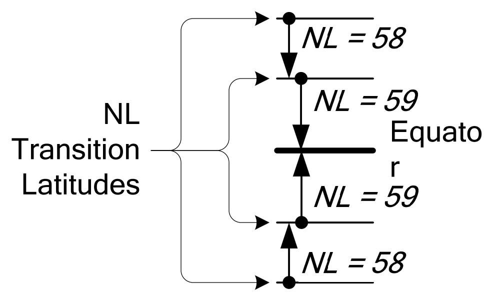


Figure 3-8 NL Values and Transition Latitudes

A rearranged version of Eq 2 gives NL Transition Latitudes as a function of the number of longitude zones. The rearranged equation is presented in Eq 3:

$$lat_{NL\ Transition} = \frac{180°}{\pi} \bullet \arccos\left(\sqrt{\frac{1-\cos\left(\frac{\pi}{2NZ}\right)}{1-\cos\left(\frac{2\pi}{NL}\right)}} \right) \quad for\ NL = 2\ to\ 4NZ-1$$

Eq 3

[A1.7.2.d, note 5]

The longitude angular zone width, $Dlon_i$ (in degrees of longitude) depends on the value of NL and is given by Eq 4:

$$Dlon_i = \begin{cases} \dfrac{360°}{NL(Rlat_i)-i}, & when\ NL(Rlat_i)-i > 0 \\ 360°, & when\ NL(Rlat_i)-i = 0 \end{cases}$$

Eq 4

[A.1.7.3.d]

A visual representation of these equations is shown in Figure 3-9. Red lines are even longitude zone boundaries and blue lines are odd longitude zone boundaries. Black lines of constant latitude are the transition latitudes defined by Eq 3.

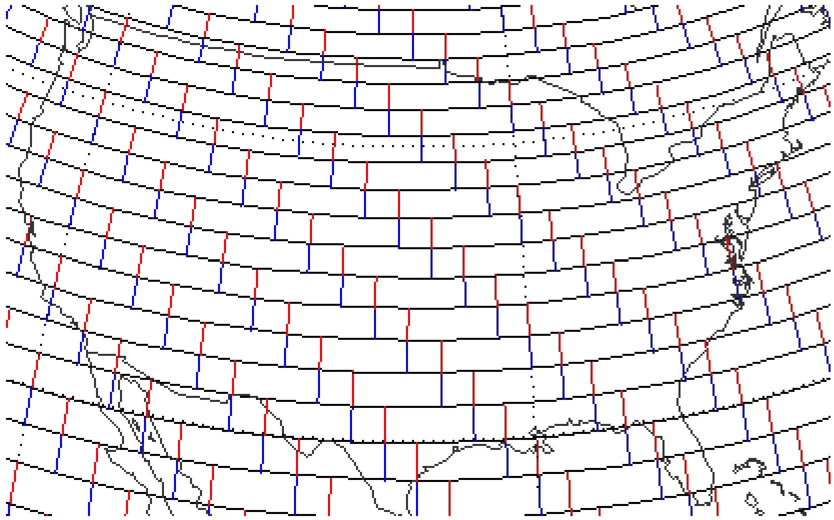


Figure 3-9 Even and Odd Longitude Zones With Transition Latitudes

Zone indexing for longitude is similar to latitude except that zone index 0 extends east from the Prime Meridian and the "tape measure" continues eastward around the globe, returning to the Prime Meridian. The number of zones on the "tape measure" varies with recovered latitude as described above.

Longitude zones are divided into bins in the same manner as latitude zones except that bin number zero is located at the western edge of each zone and bin numbers increase to the east. Even bin numbers are represented by the variable XZ_0 and odd bin numbers are represented by the variable XZ_1.

Figure 3-10 is an example showing even and odd bin numbers for airborne position at 76.06°W. Red lines delimit the even zone boundary and blue lines delimit the odd zone boundary. The point of interest, colored magenta, is in even bin number 119938 and odd bin number 16559.

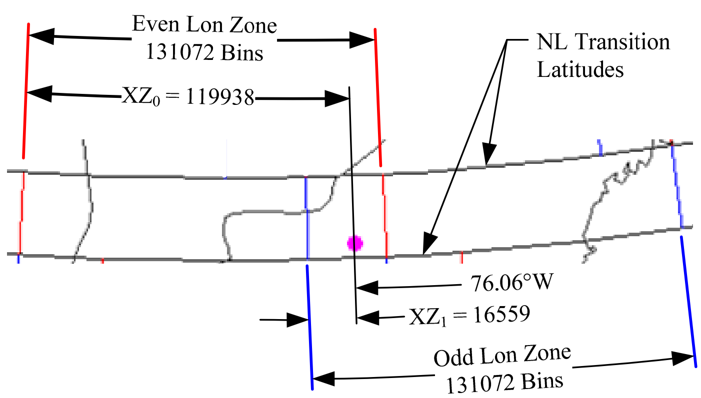


Figure 3-10 XZ for Even and Odd Longitude Zones at 76.06°W

The longitude zone index number for both the even and odd zones at the longitude shown in Figure 3-10 is 33.

4 CPR Encoding

The first step of the encoding process is to determine the latitude zone size $Dlat_i$. $Dlat_i$ is calculated with Eq 1.

The second step of the encoding process converts the input latitude into an output bin number YZ_i. The input latitude can be any real number between -90° and +90°. YZ_i is limited to a finite set of integers (0 to 131072) which represent bin numbers. Eq 5 converts input latitude *lat* into a bin number YZ_i.

$$YZ_i = floor\left(2^{Nb} \bullet \frac{MOD(lat, Dlat_i)}{Dlat_i} + \frac{1}{2}\right) \qquad \text{Eq 5} \quad [A.1.7.3.b]$$

The modulus function returns the number of degrees between the southern edge of the zone containing *lat* and *lat*. Dividing this by the latitude zone size $Dlat_i$ produces the "fraction into the zone" or "zone fraction" of the input latitude. The product of the zone fraction and the number of bins in the zone (i.e., 2^{Nb}) is approximately the bin number of the input latitude. The product is not necessarily an integer so it is rounded to the nearest integer by adding ½ and taking the floor of the sum.

The third step in the encoding process is to recover the bin centerline latitude of YZ_i. Because Eq 5 performs rounding, the bin centerline latitude of YZ_i will not equal the input latitude in most cases.

$$Rlat_i = Dlat_i\left(\frac{YZ_i}{2^{Nb}} + floor\left(\frac{lat}{Dlat_i}\right)\right) \qquad \text{Eq 6} \quad [A.1.7.3.c]$$

$YZ_i\ /2^{Nb}$ is the zone fraction of YZ_i. floor($lat/Dlat_i$) is the zone index number. The first latitude zone north of the equator is index number 0. The sum of the terms inside the parentheses is the number of whole zones plus the zone fraction of YZ_i. The product of this sum and $Dlat_i$, the number of degrees per zone, is the bin centerline latitude of YZ_i.

In the fourth step, $Rlat_i$ from Eq 6 is used to determine *NL* from Eq 2 or from a lookup table based on Eq 3. Then, $Dlon_i$ is calculated with Eq 4. Using $Rlat_i$ in the determination of *NL* ensures that the transmitted value of YZ_i is consistent with the value of *NL* used for longitude encoding. The step also ensures that only one value of *NL* is used for any latitude within a single latitude bin. The receiver only knows the latitude that is recovered from YZ_i and must determine *NL* based on this value alone. If more than one value of *NL* were used for latitudes in the same bin, the receiver would have no reliable way to determine which value of *NL* was appropriate.

The next step is to encode the longitude. This is done with Eq 7.

$$XZ_i = floor\left(2^{Nb} \bullet \frac{MOD(lon, Dlon_i)}{Dlon_i} + \frac{1}{2}\right) \qquad \text{Eq 7} \quad [A.1.7.3.e]$$

Eq 7 works in the same manner as Eq 5 except that the inputs and results are longitude values.

The final step trims YZ_i and XZ_i to the number of bits available in the position message. For surface position, this fits a 19 bit number into a 17 bit space by discarding the 2 most significant bits. For airborne and coarse TIS-B position, this step takes care of the case where the latitude or longitude falls within the most northern or most eastern ½ bin in a zone. YZ_i or XZ_i calculated from Eqs 5 or 7 would equal 2^{Nb} in these cases which is too large be encoded in the respective position messages. This step drops the most significant bit leaving only a string of zero bits for transmission. See Figure 4-1 for an example with latitude.

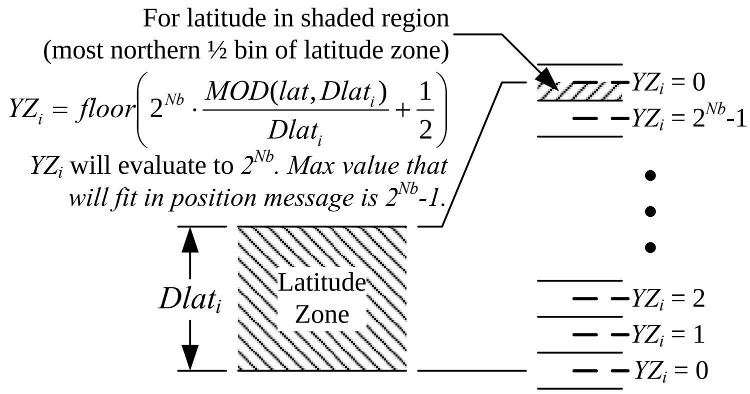


Figure 4-1 YZ_i When Latitude is in Northernmost 1/2 Bin of Latitude Zone

The following equations from section A.1.7.3f trim YZ_i and XZ_i to the appropriate number of bits:

$$\left.\begin{aligned} YZ_i &= MOD\left(YZ_i, 2^{17}\right) \\ XZ_i &= MOD\left(XZ_i, 2^{17}\right) \end{aligned}\right\} \text{ for Airborne encoding}$$

$$\left.\begin{aligned} YZ_i &= MOD\left(YZ_i, 2^{17}\right) \\ XZ_i &= MOD\left(XZ_i, 2^{17}\right) \end{aligned}\right\} \text{ for Surface encoding}$$

$$\left.\begin{aligned} YZ_i &= MOD\left(YZ_i, 2^{12}\right) \\ XZ_i &= MOD\left(XZ_i, 2^{12}\right) \end{aligned}\right\} \text{ for Coarse TIS} - \text{B encoding}$$

5 CPR Decoding

Every latitude and longitude can be mapped to a unique triple of zone index, bin number and CPR Format. However, only the bin number and CPR Format are transmitted in position messages. Receivers must recover the latitude and longitude using only the transmitted bin numbers and CPR Format. Remember that any latitude bin number YZ_i will exist in each latitude zone and that there are 15 even and 14 ¾ odd latitude zones in both the northern and southern hemispheres. The ambiguity in longitude is similar except the number of longitude zones varies from 1 to 59 depending on latitude. The MOPS provides two methods for recovering the zone and determining the position of a target in section A.1.7. One method is called Globally Unambiguous CPR Decoding and the other is called Locally Unambiguous CPR Decoding. These methods are commonly called "global decoding" and "local decoding."

Most implementations use global decoding to determine position when a target is first acquired and then switch to local decoding for ongoing position updates. Global decoding can determine the location of a target anywhere on the globe but must have one even CPR format position message and one odd CPR format position message received within 10 seconds of each other to perform a valid decode. Local decoding is limited to a range of ±½ zone in latitude and longitude from the last position update but can use an encoded position of either format with no time limit between updates.

5.1 Global Decoding for Airborne and TIS-B Position

Global decoding uses the bin numbers from one even position message and one odd position message to determine which zones contain the target and to recover the latitude and longitude. The target could be anywhere on the globe and global decoding will still work correctly.

5.1.1 Airborne and TIS-B Latitude Global Decoding

A receiver needs to determine the zone index of the target because the zone index is not transmitted in position messages. Global decoding works on the idea that the zone index is related to the offset between the southern edges of the even and odd zones that contain the latitude bin numbers transmitted in the two position messages. The principle is illustrated for latitude in Figure 5-1.

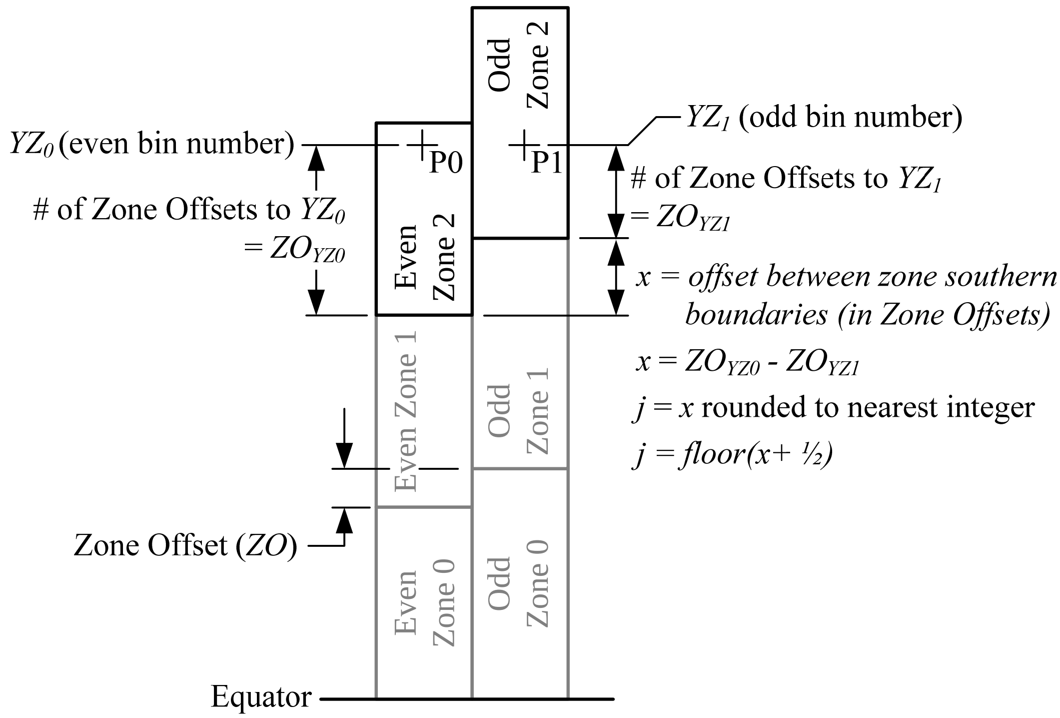


Figure 5-1 Principle of Global Decoding

Figure 5-1 is a magnified view of the first three zones north of the equator in Figure 3-3. The latitude bin numbers YZ_0 and YZ_1, which are received in separate even and odd position messages, are also shown.

Zone Offset (*ZO*) is the difference in size between an even zone and an odd zone. This difference, in degrees, is defined by Eq 8.

$$ZO = Dlat_1 - Dlat_0 = \frac{360°}{4NZ - 1} - \frac{360°}{4NZ} = \frac{360°}{4NZ(4NZ-1)} \qquad \text{Eq 8}$$

$$ZO = \frac{360°}{(59)(60)}$$

Observe in Figure 5-1 that as the zone index number increases, the offset between zone southern boundaries also increases by the value of *ZO*. The difference in latitude between the southern edges of the even and odd zones with index 1 is *ZO* and the difference in latitude between the southern edges of zones with index 2 is 2*ZO*. The zone index number and offset between zone southern boundaries are linearly related. Therefore, the offset between zone southern boundaries can be used to determine the zone index number.

Global decoding is easier to see if the following two assumptions are made:

> Assumption 1. P0 and P1 have the same latitude (e.g. stationary target or one moving due east or due west).
>
> Assumption 2. P0 and P1 are in zones with the same index number.

Figure 5-1 incorporates these assumptions. The offset between zone southern boundaries (x) of the zones with index number 2 (measured in Zone Offsets) is the number of Zone Offsets from the southern edge of the even zone to YZ_0 (ZO_{YZ0}) minus the number of Zone Offsets from the southern edge of the odd zone to YZ_1 (ZO_{YZ1}). The result will be a number that is very close to an integer, which is the expected answer, because zones are always offset by an integer multiple of the Zone Offset. If the result (x) is rounded to the nearest integer (j), the zone index number of the even or odd zone containing P0 or P1 will be equal to the number of Zone Offsets (j) between the zone southern boundaries.

The number of Zone Offsets from the southern boundary of the zone to YZ_0 (ZO_{YZ0}) or YZ_1 (ZO_{YZ1}) is the product of the number of Zone Offsets per zone (ZO_i) and the zone fraction of YZ_0 or YZ_1. The number of Zone Offsets per zone is determined with Eq 9.

$$ZO_i = \left\{\frac{Dlat_i}{ZO}\right\} = \left\{\frac{\left(\dfrac{360°}{4NZ - i}\right)}{\left(\dfrac{360°}{4NZ(4NZ-1)}\right)}\right\} \qquad \text{Eq 9}$$

$$ZO_i = \frac{4NZ(4NZ-1)}{4NZ - i} = \begin{cases} 4NZ - 1 = 59 \;\; when\ i = 0\ (even) \\ 4NZ = 60 \;\; when\ i = 1\ (odd) \end{cases}$$

The zone fraction is the bin number of P0 or P1 divided by the number of bins in the zone ($YZ_i/2^{Nb}$).

Eqs 10 & 11 define the number of zone offsets from the southern edge of a latitude zone to a bin with coordinate YZ_i.

$$ZO_{YZ0} = 59\left(\frac{YZ_0}{2^{Nb}}\right)\ for\ even\ zones \qquad \text{Eq 10}$$

$$ZO_{YZ1} = 60\left(\frac{YZ_1}{2^{Nb}}\right)\ for\ odd\ zones \qquad \text{Eq 11}$$

Subtracting Eq 11 from Eq 10 produces the number of Zone Offsets between the southern boundaries of the zones containing P0 and P1.

$$x = ZO_{YZ0} - ZO_{YZ1} = 59\left(\frac{YZ_0}{2^{Nb}}\right) - 60\left(\frac{YZ_1}{2^{Nb}}\right)$$

Eq 12

$$x = \frac{59YZ_0 - 60YZ_1}{2^{Nb}}$$

The number of Zone Offsets between zone southern boundaries must be an integer because of the way zones are defined. P0 and P1 will not have exactly the same latitude in most cases because, even if the input latitudes are equal, the CPR encoding process will result in the centerline of the even bin being a slightly different latitude than the centerline of the odd bin. Therefore, *x* must be rounded to the nearest integer to determine the true (i.e. integer) number of Zone Offsets between zone southern boundaries. Eq 13 rounds the result of Eq 12 to the nearest integer. The result is the equation given in A.1.7.7b.

$$j = floor\left(x + \frac{1}{2}\right)$$

Eq 13

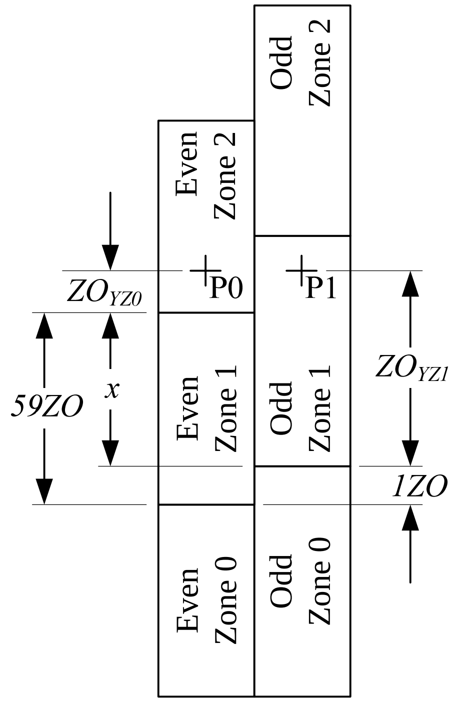
$$j = floor\left(\frac{59YZ_0 - 60YZ_1}{2^{Nb}} + \frac{1}{2}\right)$$

[A.1.7.7b]

Assumption 1 is not practical for real applications because aircraft and ground vehicles move during the period between position messages. The rounding performed by Eq 13 will accommodate up to ½ Zone Offset between P0 and P1. Assumption 1 can be relaxed to allow up to ½ Zone Offset between P0 and P1 including vehicle movement, position errors and CPR quantization errors.

Note: *Appendix A, Section A.1.7.7, Note 2 states that global decodes cannot be performed with even-odd position message pairs received more than 10 seconds apart because a target could move more than 3 nmi in a 10 second period. The 3 nmi limit comes from the requirement that the latitude (or longitude) change between P0 and P1 not exceed ½ ZO. From Eq 8, ZO ≅ 0.10169°. ½ ZO ≅ 0.05085° ≅ 3.051 nmi.*

Assumption 2 is also impractical because P0 and P1 could end up in zones with different zone index numbers. An example is shown in Figure 5-2.



$$x = ZO_{YZ0} - ZO_{YZ1} \approx -59ZO + 1ZO = -58ZO$$
$$j = floor(x + ½) = floor(ZO_{YZ0} - ZO_{YZ1} + ½)$$
$$j = -59ZO + 1ZO = -58ZO$$

Figure 5-2 - Points in Zones With Different Zone Index Numbers

In Figure 5-2, P0 is in a zone with index number 2 while P1 is in a zone with index number 1. The value of *j* in Figure 5-2 will be -58 because Even Zone 1 is 59 Zone Offsets tall and the bottom of Odd Zone 1 is 1 Zone Offset above the bottom of Even Zone 1. The Modulus function defined in Appendix A can be used to recover the zone index number when the offset between zone southern boundaries is negative. The Modulus function is repeated in Eq 14.

$$MOD(x, y) = x - y \bullet floor\left(\frac{x}{y}\right) \; where \; y \neq 0$$

Eq 14

[A.1.7.2.c]

The value of *y* is chosen to produce the zone index number from *j*. Figure 5-3 depicts the relationship between *j* and the index number of each zone.

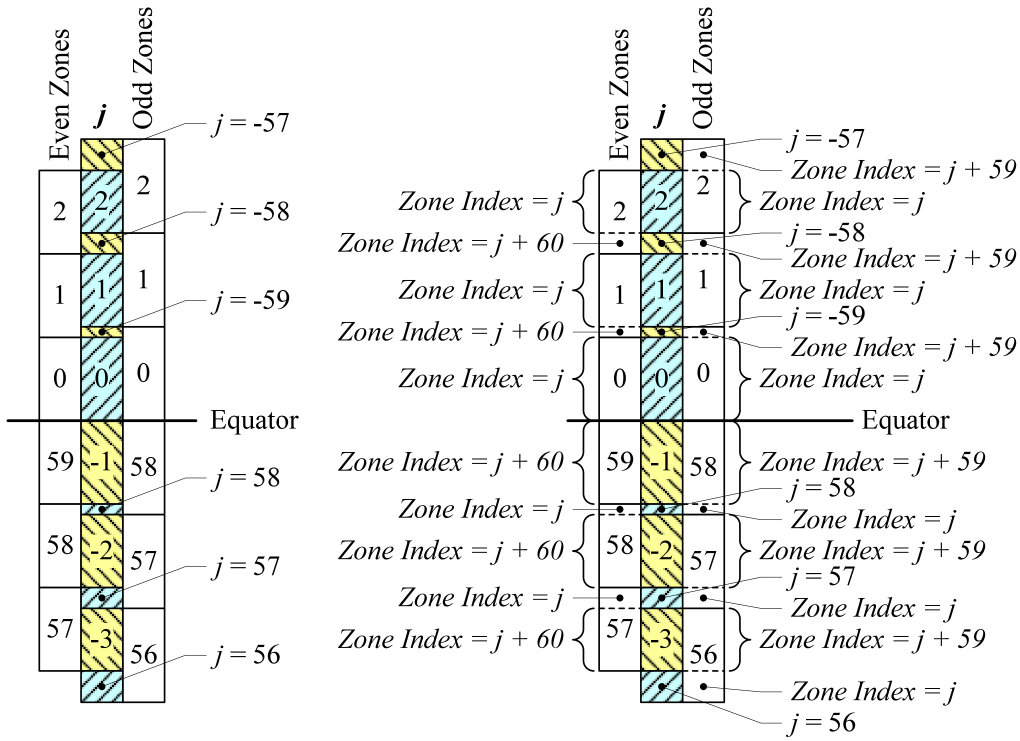


Figure 5-3 Relationship Between *j* and Zone Index

If *y* in Eq 14 is replaced with the number of even (60) or odd (59) zones around the globe (refer to the "tape measure" in Figure 3-3) and if *x* is replaced with *j*, Eq 14 becomes:

$$MOD(j,4NZ-i) = j-(4NZ-i)\bullet floor\left(\frac{j}{4NZ-i}\right) \qquad \text{Eq 15}$$

The range of *j* is -59 to +58. The quantity *4NZ-i* is either 59 or 60. *floor(j/(4NZ-i))* will be 0 when *j* is positive and -1 when *j* is negative. When j is positive, the second term of Eq 14 is zero and the zone index equals *j*. When j is negative, the sign in front of the second term is positive and the value of the term is *4NZ-i*. These forms of the equation match the forms shown in Figure 5-3. The zone index can be determined from *j* using Eq 16.

$$\begin{aligned} Zone\ Index &= MOD(j,4NZ-i) \\ Zone\ Index &= MOD(j,60-i) \end{aligned} \qquad \text{Eq 16}$$

The latitude is recovered using the zone index and bin number (YZ_i). The bin number divided by the number of bins in a zone equals the zone fraction of the bin number. Multiplying the height of a zone by the sum of the zone index and the zone fraction will produce the bin centerline latitude of YZ_i.

$$Rlat_i = Dlat_i\left(MOD(j,60-i)+\frac{YZ_i}{2^{Nb}}\right) \qquad \text{Eq 17 [A.1.7.7.c]}$$

$Rlat_0$ is the latitude of the even position message used in the global decode and $Rlat_1$ is the latitude of the odd position message. Recalling the tape measure of Figure 3-3, the northern hemisphere will have zone index numbers between 0 and 14 and latitude values between 0° and 90°. The southern hemisphere will have zone index numbers between 45 and 59 and latitude values between 270° and 360°. Latitudes between 90° and 270° are invalid. Southern hemisphere latitude values can be put in the range of 0° to -90° by subtracting 360° from the result of Eq 17 or by using Eq 18.

$$Rlat_i = MOD(Rlat_i + 180°, 360°) - 180° \qquad \text{Eq 18}$$

5.1.2 Airborne and TIS-B Longitude Global Decoding

Longitude global decoding is similar to latitude global decoding in that even and odd messages are used to count zone offsets and determine the zone index. The concept of offset between zone southern boundaries is replaced with the idea of offset between zone western boundaries. The measured offset is rounded to the nearest integer, which is named *m* instead of *j*, and the Modulus function is used to get the zone index from positive or negative values of *m*. The rule that the even and odd positions must be within ½ Zone Offset of each other also applies to longitude global decoding.

The main difference between longitude global decoding and latitude global decoding is that the number of longitude zones around the globe varies with latitude. There are always 60 even zones and 59 odd zones around the globe in latitude global decoding. In longitude global decoding, there are between 1 and 59 longitude zones, as depicted in Figure 3-2.

The first step in longitude global decoding is to ensure that the values of NL for the even and odd positions are the same. If they are different, the Zone Offset sizes will be different and the equation to compute the number of Zone Offsets will produce erroneous results. Eqs 17 and 18 will produce one latitude for each bin number used in the latitude global decode (i.e. $Rlat_0$ and $Rlat_1$). *NL* for each of these latitudes is computed with Eq 2. If the *NL* values are different, the receiver must wait until a pair of even and odd messages arrive that produce the same value for *NL*.

If the *NL* values are the same, longitude global decoding proceeds using the same logic as latitude global decoding. In Eqs 8 and 9, the factor *4NZ*, the number of even latitude zones around the globe, is replaced with *NL*, the number of even longitude zones around the globe at $Rlat_0$ and $Rlat_1$. Eq 8 for the size of the Zone Offset becomes Eq 19.

$$\begin{aligned} ZO &= Dlon_1 - Dlon_0 = \frac{360°}{NL-1} - \frac{360°}{NL} \quad for\ NL > 1 \\ ZO &= \frac{360°}{NL(NL-1)} \quad for\ NL > 1 \end{aligned} \qquad \text{Eq 19}$$

Eq 9 for the number of Zone Offsets in a zone becomes Eq 20

$$\begin{aligned} ZO_i &= \left\{\frac{Dlon_i}{ZO}\right\} = \left\{\frac{\left(\frac{360°}{NL-i}\right)}{\left(\frac{360°}{NL(NL-1)}\right)}\right\} \quad for\ NL > 1 \\ ZO_i &= \frac{NL(NL-1)}{NL-i} = \begin{cases} NL-1 \ \ when\ i = 0\ (even) \\ NL \ \ when\ i = 1\ (odd) \end{cases} for\ NL > 1 \end{aligned} \qquad \text{Eq 20}$$

Eq 10 for the number of Zone Offsets to an even bin becomes Eq 21.

$$ZO_{XZ0} = (NL-1)\left(\frac{XZ_0}{2^{Nb}}\right) \ for\ even\ zones \qquad \text{Eq 21}$$

Eq 11 for the number of Zone Offsets to an odd bin becomes Eq 22.

$$ZO_{XZ1} = (NL)\left(\frac{XZ_1}{2^{Nb}}\right) \; for\ even\ zones \qquad \text{Eq 22}$$

Subtracting Eq 22 from Eq 21 produces the number of Zone Offsets between the western boundaries of the zones containing P0 and P1.

$$x = ZO_{XZ0} - ZO_{XZ1} = (NL-1)\left(\frac{XZ_0}{2^{Nb}}\right) - (NL)\left(\frac{XZ_1}{2^{Nb}}\right)$$
$$x = \frac{(NL-1)XZ_0 - (NL)XZ_1}{2^{Nb}} \qquad \text{Eq 23}$$

The result of Eq 23 must be rounded to the nearest integer. Instead of assigning the result to *j*, as is done for latitude in Eq 13, the result is assigned to *m*.

$$m = floor\left(x + \frac{1}{2}\right) \qquad \text{Eq 24}$$
$$m = floor\left(\frac{(NL-1)XZ_0 - (NL)XZ_1}{2^{Nb}} + \frac{1}{2}\right) \qquad \text{[A.1.7.7.f]}$$

As was the case with Eq 13, the rounding performed by Eq 24 will accommodate up to ½ Zone Offset between P0 and P1 including vehicle movement, position errors and CPR quantization errors.

m will behave the same as *j* in that the value of *m* will be negative when the even and odd longitude zones containing P0 and P1 have different Zone Index numbers. The Zone Index is recovered from *m* in the same manner that it is from *j*. Again, the term *4NZ* is replaced with *NL* and *j* is replaced with *m* in Eq 15 to form Eq 25.

$$MOD(m, NL-i) = m - (NL-i) \bullet floor\left(\frac{m}{NL-i}\right) \; NL > 1 \qquad \text{Eq 25}$$

Eq 16 becomes Eq 26.

$$Zone\ Index = MOD(m, NL-i) \; for\ NL > 1 \qquad \text{Eq 26}$$

The recovered longitude $Rlon_i$ is determined in the same manner as the recovered latitude $Rlat_i$.

$$Rlon_i = Dlon_i\left(MOD(m, NL-i) + \frac{XZ_i}{2^{Nb}}\right) \; for\ NL > 1 \qquad \text{Eq 27}$$

$Rlon_i$ will be a value between 0° and 360°. It can be placed in the range of -180° to +180° with Eq 28.

$$Rlon_i = MOD(Rlon_i + 180°, 360°) - 180° \qquad \text{Eq 28}$$

Eqs 19, 20, 25, 26 and 27 will be indeterminate when *NL*=1. From Eq 2, *NL* will be 1 at latitudes greater than 87°N or less than -87°S. When *NL* = 1, $Dlon_i$, the longitude zone width, equals 360° (ref Eq 4) for both even and odd encoding. With one zone, there is no need to determine the zone index. Longitude can be recovered by multiplying the zone width ($Dlon_i$) by the zone fraction ($XZ_i/2^{Nb}$). While this is a straightforward approach, the algorithm description in Appendix A does not handle *NL*=1 this way.

When NL is 1, Eq 24 (A.1.7.7f) for m reduces to either 0 or -1 for all values of XZ_i and Nb. Section A.1.7.7g defines a variable n_i.

$$n_i = \text{greater of } [\mathrm{NL}(Rlat_i) - i] \text{ and } 1 \qquad \text{Eq 29}$$

Section A.1.7.7g also defines a formula for $Rlon_i$ in terms of m and n_i.

$$Rlon_i = Dlon_i\left(MOD(m, n_i) + \frac{XZ_i}{2^{17}}\right) \qquad \text{Eq 30 [A.1.7.7g]}$$

When m = 0 and n_i = 1, $MOD(m, n_i)$ equals 0. When m = -1, $MOD(m, n_i)$ also evaluates to 0. In either case, Eq 30 for $Rlon_i$ reduces to the zone width ($Dlon_i$) multiplied by the zone fraction ($XZ_i/2^{17}$). This is the expected result.

For values of $NL > 1$, n_i will equal $(NL - i)$ and Eqs 27 and 30 will be equivalent.

Section A.1.7.7e redefines $Dlon_i$ in terms of n_i. Eqs 4 and 31 are equivalent. The reason for the redefinition is not clear.

$$Dlon_i = \frac{360°}{n_i} \qquad \text{Eq 31 [A.1.7.7.e]}$$

5.2 Surface Global Decoding

Surface global decoding is the same as airborne and TIS-B local decoding except that the latitude and longitude zone widths $Dlat_i$ and $Dlon_i$ are reduced by a factor of 4. Surface position encoding is performed with 19 bits of resolution (reference Table 3-1) but only the 17 least significant bits are transmitted in the surface position message (A.1.7.3f). Dropping the two most significant bits reduces the number of bins per zone by a factor of 4 but the width of the bins must remain the same for both encoding and decoding. Therefore, the latitude and longitude zone widths $Dlat_i$ and $Dlon_i$ used in decoding must also be smaller by a factor of 4.

The smaller zone width means that the length of the "tailor's tape measure" described in Figure 3-3 will be ¼ of the distance around the globe. Calculations for recovered latitude and longitude $Rlat_i$ and $Rlon_i$ will be in the range of 0° to 90° (i.e. northern hemisphere and 0° to 90° east longitude). Additional solutions will exist at 90°, 180° and 270° from the calculated latitude and longitude. The receiver must choose the appropriate solution and that solution may be in a different quadrant of the globe than the receiver. Latitude solutions between 90° and 270° are discarded. If there were targets broadcasting surface position messages near either of the poles, longitude ambiguity could be a problem. This is an unlikely scenario.

For surface global and local decoding, the latitude zone width is given by Eq 32 (reference Eq 1).

$$Dlat_i = \frac{1}{4}\left(\frac{360°}{4NZ - i}\right) = \frac{90°}{4NZ - i} \qquad \text{Eq 32 [A.1.7.8a]}$$

NZ = 15
i = 0 for even encoding
i = 1 for odd encoding

Latitude Zone Offset defined in Eq 8 will also be smaller by a factor of 4. Eqs 9 through 18 are valid for surface global decoding as long as the correct values of $Dlat_i$ and *ZO* are used.

$Rlat_i$ from Eq 17 will be between 0° and 90°. An additional solution in the southern hemisphere will be located at $Rlat_i - 90°$. The receiver must choose the closer of the two.

NL is calculated using Eq 2 and $Rlat_i$ but the result is the number of longitude zones in one quadrant of the globe rather than the number of zones around the globe.

Before proceeding with longitude global decoding, the receiver must verify that $\mathrm{NL}(Rlat_0) = \mathrm{NL}(Rlat_1)$. If not, the decode is aborted and the process is restarted with a new pair of even and odd position messages.

For surface global and local decoding, the longitude zone width is obtained by dividing Eq 4 by 4. This results in Eq 33:

$$Dlon_i = \begin{cases} \dfrac{90°}{NL(Rlat_i) - i}, & when\ NL(Rlat_i) - i > 0 \\ 90°, & when\ NL(Rlat_i) - i = 0 \end{cases}$$

Eq 33

[A.1.7.8e]

Longitude Zone Offset defined in Eq 19 will also be smaller by a factor of 4. Eqs 20 through 27 are valid for surface global decoding as long as the correct values of $Dlon_i$ and *ZO* are used.

$Rlon_i$ will be between 0° and 90°. Additional solutions will exist at longitudes of 90°, 180° and 270° to the east of $Rlon_i$. The receiver must choose the appropriate longitude which will be, in most cases, the closest longitude.

5.3 Airborne and TIS-B Locally Unambiguous Decoding

Local decoding uses a “reference position” and the contents of one position message to recover the latitude and longitude of the target. The difference between the target position and the reference position must be less than ½ of a zone in both latitude and longitude. The reference position is usually the last successfully decoded position received from the target. Since ½ of a zone is approximately 180 nautical miles for airborne position and 45 nautical miles for surface position, the position change between two successfully decoded position messages is unlikely to exceed the ½ zone limit.

The effect of the local decoding algorithm is to create a sliding region 1 zone wide in latitude and longitude, centered on the reference position. Because the region is 1 zone wide, each bin number occurs only once in the region, ensuring a unique result for each encoded latitude and longitude. Figure 5-4 illustrates the behavior of the sliding region for latitude. The number of bins per latitude zone has been reduced to 8 (i.e. $Nb = 3$) to simplify the figure. Behavior of the sliding region is the same for longitude except the figure is rotated 90° clockwise and references to latitude are replaced with longitude.

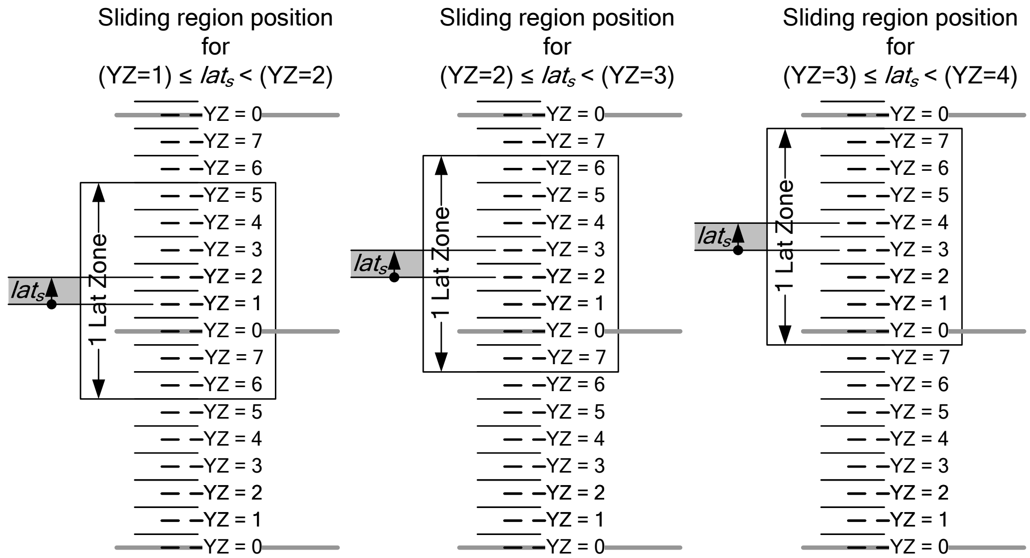


Figure 5-4 Local Decoding Sliding Region Behavior

The sliding region position changes when lat_s moves from one grey region to another.

In the unlikely event that the target position is more than ½ zone from the reference position, the target position will appear to "wrap around" the other side of the ±½ zone region. For example, if the target was in bin 7 (YZ=7) in the upper zone with the reference position between YZ=1 and YZ=2 (see left side of Figure 5-4), the recovered latitude would be for YZ=7 in the lower zone. This wrap around behavior occurs because the algorithm assumes the target is within ½ zone of the reference position and the only bin number 7 is in the lower zone. In effect, the recovered position is shifted one zone from the actual position in the direction of the reference position. Some older implementations used the receiver position as the reference position instead of the last target position. Surface targets more than ½ zone away (45 nautical miles) would decode incorrectly as described above.

The local decoding algorithm may be easier to visualize with a simplified example. Imagine a road with a marker at each kilometer stating the number of kilometers from the beginning of the road. A person walking along the road toward the end, who is known to be within ½ km of the 22 km marker, announces they are 0.2 km past the last marker. What is their location relative to the beginning of the road? Since they are within ½ km of marker 22 and they are 0.2 km past the last marker, they are 22.2 km from the beginning of the road. Suppose they had announced their position as 0.7 km past the last marker. Their location would be 21.7 km from the beginning of the road because this is the only location that is both 0.7 km past the last marker and within ½ km of 22 km marker. This is the logic embodied in the local decoding algorithm.

5.3.1 Airborne and TIS-B Latitude Local Decoding

Local decoding determines latitude first in order to compute *NL* and then determines longitude. Latitude is calculated from the bin number and CPR format transmitted in the position message, and the zone index, which is computed from the last known target position. The algorithm sums the Zone Index of the reference position (RP) and the difference between the Zone Indices of the reference position and the encoded position (EP) to obtain the Zone Index of the encoded position.

$$ZI_{EP} = ZI_{RP} + \Delta ZI_{RP \to EP} \qquad \text{Eq 34}$$

Because the encoded position is assumed to be within ½ zone of the reference position, the difference in zone indices $\Delta ZI_{RP \to EP}$ is limited to -1, 0 or +1.

The Zone Index of the reference position ZI_{RP} is determined with Eq 35.

$$ZI_{RP} = floor\left(\frac{lat_s}{Dlat_i}\right) \qquad \text{Eq 35}$$

lat_s is the reference latitude. $Dlat_i$ is the latitude zone width from Eq 1. The first zone north of the equator has an index number of zero. Latitude Zone Index numbers south of the equator start at -1 and increase in magnitude toward the South Pole. Note that southern hemisphere latitude zone indices for local decoding are numbered differently than the zone indices for global decoding.

The difference between the latitude zone indices of the reference position and of the encoded position $(\Delta ZI_{RP \to EP})$ is the difference between the latitude zone fractions of the reference position and the encoded position $(f_{RP} - f_{EP})$, rounded to the nearest integer. This is shown for $f_{RP} > ½$ in Figure 5-5.

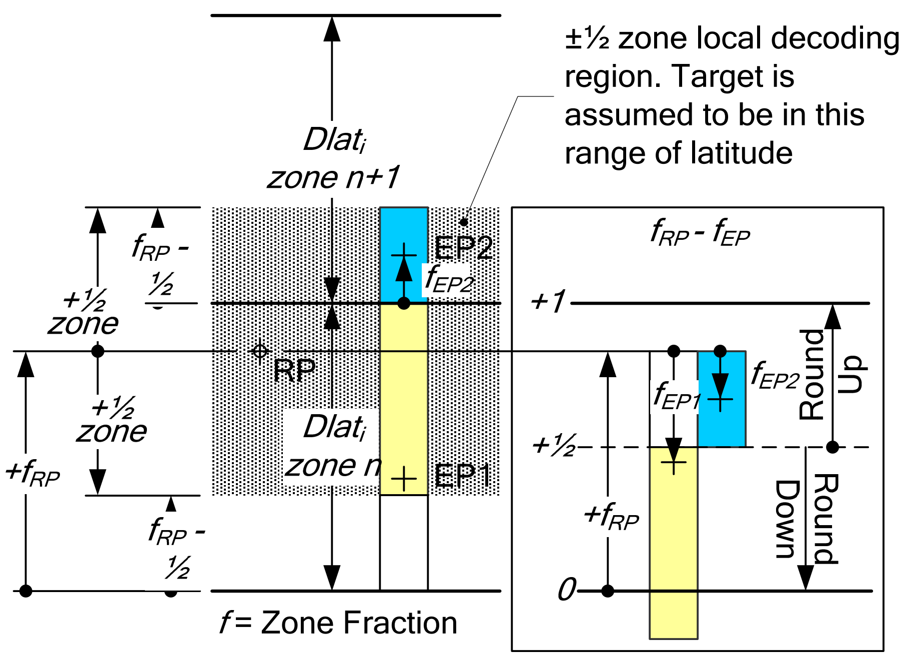


Figure 5-5 Local Decoding Zone Index Determination when f_{RP} > 1/2

f_{RP} is given by Eq 36.

$$f_{RP} = \frac{MOD(lat_s, Dlat_i)}{Dlat_i} \qquad \text{Eq 36}$$

f_{EP} is given by Eq 37.

$$f_{EP} = \frac{YZ_i}{2^{Nb}} \qquad \text{Eq 37}$$

The RP and EP are assumed to lie within the ±½ zone local decoding region (the shaded area in Figure 5-5), but the RP and EP may be in different latitude zones (e.g. RP and EP2). The difference in zone fractions is used to determine whether the EP is in the same zone as the RP, one zone to the north or one zone to the south.

The conversion from the difference in zone fractions *(f_{RP} - f_{EP})* to the difference in zone indices *($\Delta ZI_{RP\rightarrow EP}$)* is done by rounding as shown in Eq 38.

$$\Delta ZI_{RP\rightarrow EP} = floor\left(f_{RP} - f_{EP} + \frac{1}{2} \right) \qquad \text{Eq 38}$$

If the two positions are in the same zone, the difference in zone fractions will always be between -½ and +½. Eq 38 will produce a difference in Zone Indices *($\Delta ZI_{RP\rightarrow EP}$)* of 0.

If EP is in the zone to the north of RP (e.g. EP2 in Figure 5-5), the difference in zone indices $\Delta ZI_{RP\rightarrow EP}$ is expected to be +1. f_{EP} will be less than f_{RP} – ½ and therefore *($f_{RP} - f_{EP}$)* will be greater than +½. Applying Eq 38 for $\Delta ZI_{RP\rightarrow EP}$ to *(f_{RP} - f_{EP})* will produce the expected value of +1. This value is added to the zone index of RP to get the zone index of EP (n+1).

The case where f_{RP} < ½ is shown in Figure 5-6.

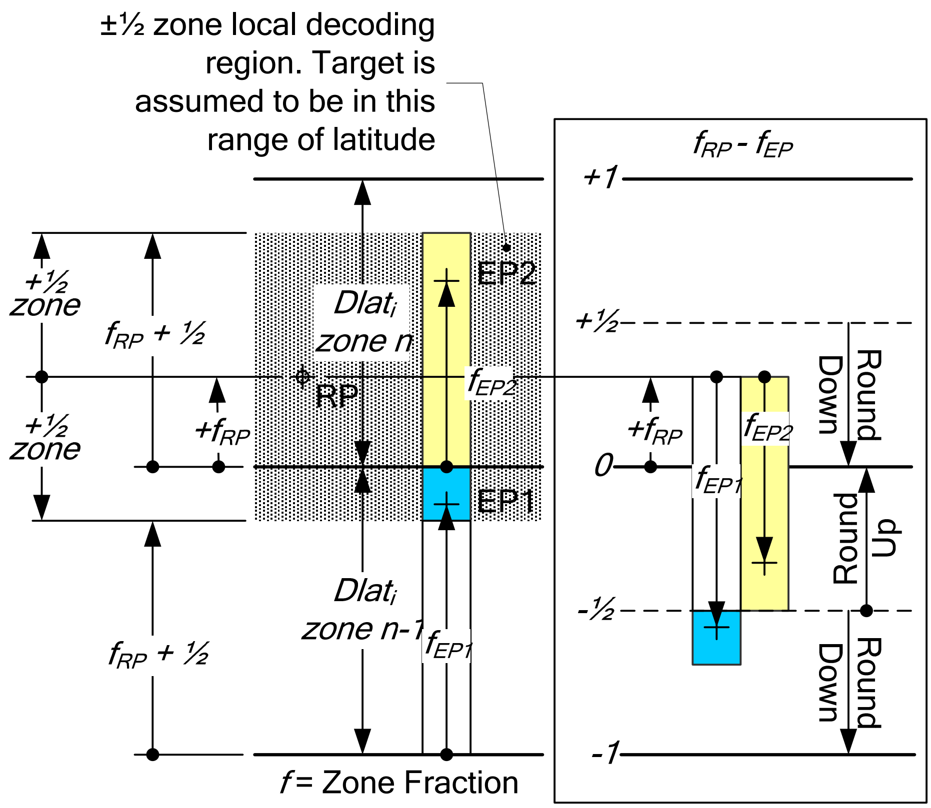


Figure 5-6 Local Decoding Zone Index Determination when f_{RP} <1/2

If EP is in the zone to the south of RP (e.g. EP1 in Figure 5-6), the difference in zone indices $\Delta ZI_{RP\rightarrow EP}$ is expected to be -1. f_{EP} will be greater than f_{RP} + ½ and therefore *($f_{RP} - f_{EP}$)* will be less than -½. Applying Eq 38 for $\Delta ZI_{RP\rightarrow EP}$ to *(f_{RP} - f_{EP})* will produce the expected value of -1. This value is added to the zone index of RP to get the zone index of EP (n-1).

Eq 38 produces the correct difference in zone indices for any two reference position and encoded position zone fractions. Substituting Eqs 35 through 38 into Eq 34 and rearranging results in Eq 39.

$$ZI_{EP} = ZI_{RP} + \Delta ZI_{RP \to EP}$$

$$ZI_{EP} = floor\left(\frac{lat_s}{Dlat_i}\right) + floor\left(f_{RP} - f_{EP} + \frac{1}{2}\right)$$

$$ZI_{EP} = floor\left(\frac{lat_s}{Dlat_i}\right) + floor\left(\frac{MOD(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{Nb}} + \frac{1}{2}\right)$$

$$j = ZI_{EP} = floor\left(\frac{lat_s}{Dlat_i}\right) + floor\left(\frac{1}{2} + \frac{MOD(lat_s, Dlat_i)}{Dlat_i} - \frac{YZ_i}{2^{Nb}}\right)$$

Eq 39

[A.1.7.5b]

The latitude of the encoded position is the latitude zone width, $Dlat_i$, multiplied by the sum of the encoded position zone index j and the encoded position zone fraction.

$$Rlat_i = Dlat_i\left(j + \frac{YZ_i}{2^{Nb}}\right)$$

Eq 40

[A.1.7.5c]

5.3.2 Airborne and TIS-B Longitude Local Decoding

Local decoding of longitude is similar to local decoding of latitude. Longitude Zone Width is calculated with Eq 4, which is repeated in [A.1.7.5d]. Then, the longitude zone index is computed using Eq 41.

$$ZI_{EP} = ZI_{RP} + \Delta ZI_{RP \to EP}$$

$$ZI_{EP} = floor\left(\frac{lon_s}{Dlon_i}\right) + floor\left(f_{RP} - f_{EP} + \frac{1}{2}\right)$$

$$ZI_{EP} = floor\left(\frac{lon_s}{Dlon_i}\right) + floor\left(\frac{MOD(lon_s, Dlon_i)}{Dlon_i} - \frac{XZ_i}{2^{Nb}} + \frac{1}{2}\right)$$

$$m = ZI_{EP} = floor\left(\frac{lon_s}{Dlon_i}\right) + floor\left(\frac{1}{2} + \frac{MOD(lon_s, Dlon_i)}{Dlon_i} - \frac{XZ_i}{2^{Nb}}\right)$$

Eq 41

[A.1.7.5e]

Eq 41 is the same as Eq 39 except the following variable substitutions have been made:

- RP latitude lat_s replaced with RP latitude lon_s
- Latitude Zone Width $Dlat_i$ replaced with Longitude Zone Width $Dlon_i$
- Latitude bin number YZ_i replaced with Longitude bin number XZ_i
- EP latitude zone index j replaced longitude zone index m

ZI and f in Eq 41 refer to longitude zone indices and zone fractions whereas the same variables in Eq 39 refer to latitude zone indices and zone fractions.

Longitude zone indices start at zero on the east side of the prime meridian and increase with increasing east longitude. If RP longitude is between 0° and 360°, the longitude zone indices will range from 0 to 59 and the recovered longitude $Rlon_i$ will be in the range of 0° to 360°. If RP longitude is between -180° and +180°, the longitude zone indices will range from -30 to +29 and the recovered longitude $Rlon_i$ will be in the range of -180° to +180°.

The longitude of the encoded position is the longitude zone width, $Dlon_i$, multiplied by the sum of the encoded position zone index m and the encoded position zone fraction.

$$Rlon_i = Dlon_i\left(m + \frac{XZ_i}{2^{Nb}}\right)$$

Eq 42

[A.1.7.5f]

5.4 Surface Local Decoding

Surface local decoding is the same as airborne and TIS-B local decoding except that the latitude and longitude zone widths $Dlat_i$ and $Dlon_i$ are reduced by a factor of 4. Surface position encoding is performed with 19 bits of resolution (reference Table 3-1) but only the 17 least significant bits are transmitted in the surface position message (A.1.7.3f). Dropping the two most significant bits reduces the number of bins per zone by a factor of 4 but the width of the bins must remain the same for both encoding and decoding. Therefore, the zone widths for decoding must also be smaller by a factor of 4.

The ±½ zone sliding region is applicable for surface local decoding, but since the zone width is smaller by a factor of 4, the sliding region width will be smaller by a factor of 4 also. This means that instead of a surface target needing to be within approximately ±180 nautical miles of the reference position, the surface target must be within approximately ±45 nautical miles.

Eqs 34 through 40 are applicable for surface latitude local decoding as long as the reduced latitude zone width $Dlat_i$ from Eq 32 is used in the calculations.

NL is calculated using Eq 2 and $Rlat_i$ but the result is the number of longitude zones in one quadrant of the globe rather than the number of zones around the globe.

Eqs 41 through 42 are applicable for surface longitude local decoding as long as the reduced longitude zone width $Dlon_i$ from Eq 33 is used in the calculations.

6 Summary

This appendix presents an expanded description of the CPR equations given in Appendix A. Any point on the globe is described by zone numbers, bin numbers and a CPR format in the CPR coordinate system. The encoding algorithm converts latitude and longitude into bin numbers. Global decoding is used to find the position of a target when it is first acquired, and recovers latitude and longitude from a pair of even and odd messages. Local decoding is used when a previous position is known and recovers latitude and longitude from a single position message and last known position of the target. Surface decoding is the same as airborne decoding except the surface zone size is ¼ of the airborne zone size.