

## Capítulo 5. CONVERSIÓN DE COORDENADAS

Tal y como se ha descrito en el apartado anterior (4.2.8), la función *print\_ob* devuelve la posición en coordenadas cartesianas de los objetos detectados. Esta posición está referida al sistema de coordenadas con origen en el LIDAR y se conoce como Sistema de Coordenadas Cuerpo (o Body). Sin embargo, este sistema de referencia no es el más adecuado.

El sistema de referencia más empleado es el Sistema de Coordenadas Geodésicas, también llamado LLA por el nombre de las coordenadas (Longitud, Latitud, Altura). Por ello, nuestro objetivo será pasar de coordenadas cuerpo a coordenadas geodésicas. Este cambio no es inmediato ya que se necesita pasar por sistemas coordenados intermedios.

### 5.1. *Sistemas de Coordenadas*

Los sistemas de coordenadas que se van a estudiar son:

- Sistema de Coordenadas Geodésicas (LLA)
- Sistema de Coordenadas Centradas en la Tierra, Fijas en la Tierra (ECEF)
- Sistema de Coordenadas Norte, Este, Abajo (NED)
- Sistema de Coordenadas Cuerpo (BODY)

Gráficamente:

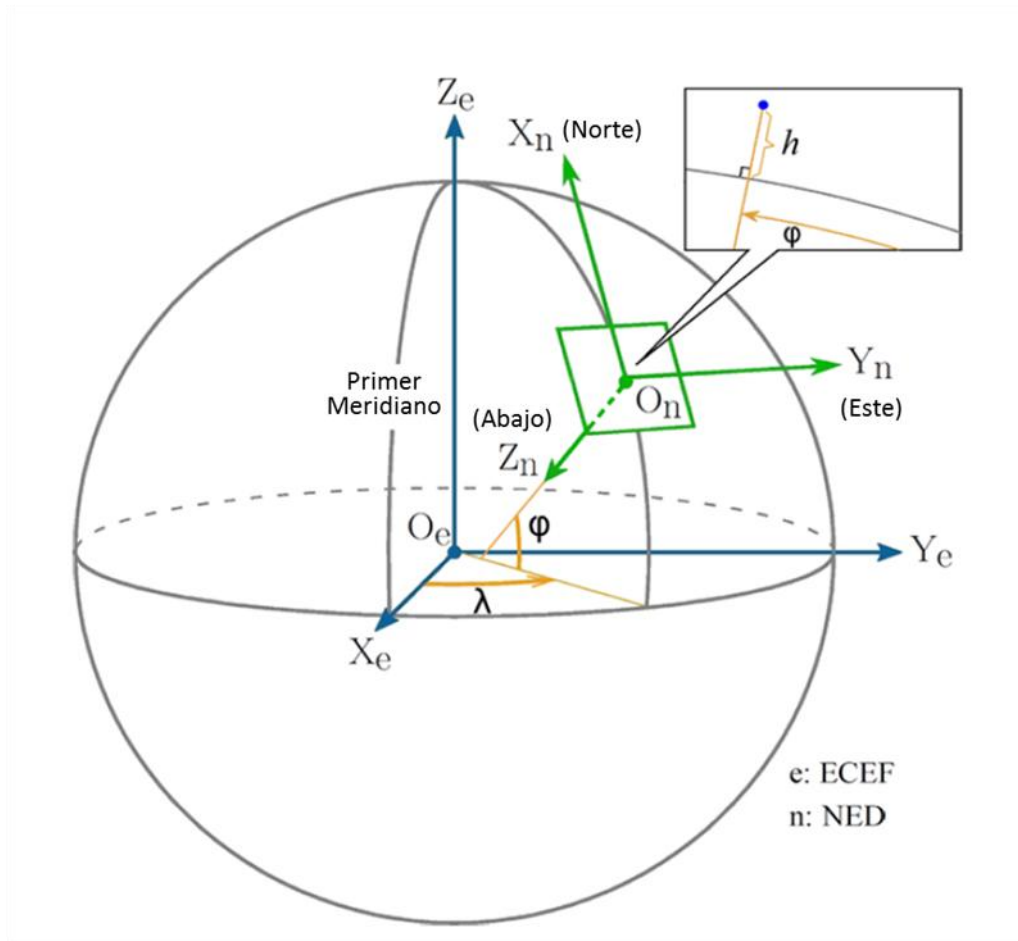


Figura 16: Sistemas de Coordenadas Geodésico, ECEF y NED

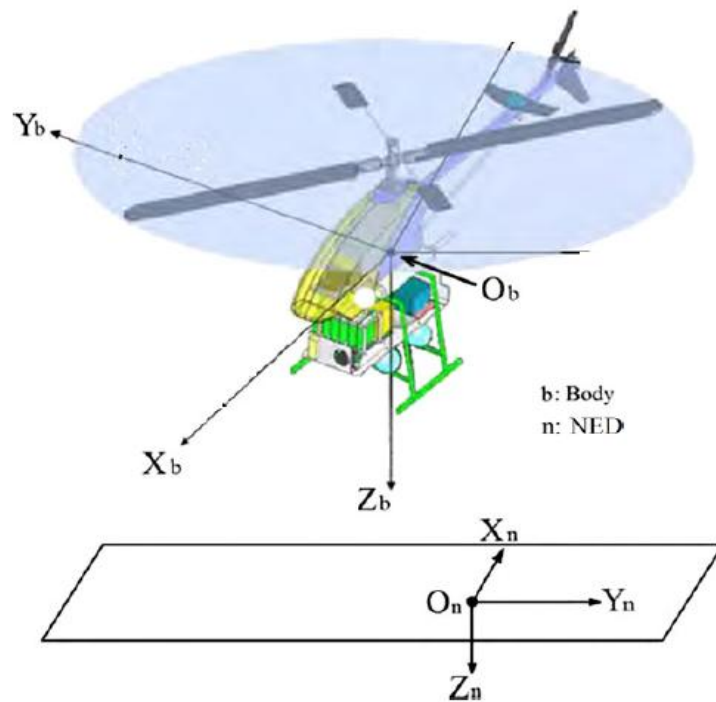


Figura 17: Sistema de Coordenadas Cuerpo

### 5.1.1. Sistemas de coordenadas geodésicas (LLA)

Este sistema de coordenadas (ver *Figura 16*) es muy usado en sistemas de navegación basados en GPS. Como se puede observar, no es un sistema de coordenadas cartesianas habitual sino un sistema que caracteriza a un punto de coordenadas cercano a la superficie de la Tierra expresado en términos de longitud, latitud y altura, los cuales se denotan, respectivamente, como  $\lambda$ ,  $\varphi$  y  $h$ . La longitud mide el ángulo rotacional (en el rango de  $-180^\circ$  a  $180^\circ$ ) entre el primer meridiano y el punto medido. La latitud mide el ángulo (en el rango de  $-90^\circ$  a  $90^\circ$ ) entre el plano ecuatorial y la normal del elipsoide de referencia que pasa a través del punto medido. La altura es la distancia local vertical entre el punto medido y el elipsoide de referencia.

El vector de posición en el sistema de coordenadas geodésico se denota por:

$$P_g = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix} \quad (33)$$

Los parámetros más importantes asociados a este sistema son:

- El semieje mayor  $R_{Ea}$
- El factor de achatamiento  $f$
- El semieje menor  $R_{Eb}$
- La primera excentricidad  $e$
- La segunda excentricidad  $e'$
- El radio de curvatura meridiano  $M_E$
- El radio de curvatura del primer vertical  $N_E$

Los dos primeros parámetros están perfectamente definidos mientras que los demás se derivan del modelo elipsoidal basado en el WGS 84. Más concretamente:

$$R_{Ea} = 6,378,137.0 \text{ m} \quad (34)$$

$$f = \frac{1}{298.3257223563} \quad (35)$$

$$R_{Eb} = R_{Ea} (1 - f) = 6,356,752.0 \text{ m} \quad (36)$$

$$e = \frac{\sqrt{R_{Ea}^2 - R_{Eb}^2}}{R_{Ea}} = 0.08181919 \quad (37)$$

$$e' = \frac{\sqrt{R_{Ea}^2 - R_{Eb}^2}}{R_{Eb}} = 0.08209441 \quad (38)$$

$$M_E = \frac{R_{Ea}(1-e^2)}{(1-e^2 \sin^2 \varphi)^{\frac{3}{2}}} \quad (39)$$

$$N_E = \frac{R_{Ea}}{\sqrt{1-e^2 \sin^2 \varphi}} \quad (40)$$

### 5.1.2. **Sistemas de coordenadas centradas en tierra, fijas en tierra (ECEF)**

El sistema de coordenadas ECEF rota con la Tierra alrededor de su eje de rotación. Como tal, un punto fijo sobre la superficie de la Tierra tiene un conjunto de coordenadas fijo. El origen y el eje de este sistema de coordenadas, mostrados en la *Figura 16*, están definidos como sigue:

- El origen (denotado como  $O_e$ ) se localiza en el centro de la tierra.
- El eje z (denotado como  $Z_e$ ) se extiende a lo largo del eje de rotación de la Tierra, apuntando al polo norte.
- El eje x (denotado como  $X_e$ ) corta a la esfera de la tierra en latitud y longitud  $0^\circ$ .
- El eje y (denotado como  $Y_e$ ) es ortogonal a los ejes x y z en el sentido definido por la regla de la mano derecha.

De forma similar a las coordenadas geodésicas, el vector de posición en el sistema de coordenadas ECEF se denota por:

$$P_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} \quad (41)$$

### 5.1.3. **Sistema de coordenadas norte, este, abajo (NED)**

El sistema de coordenadas NED se trata de un sistema fijado a un punto de la superficie de la Tierra. Se basa en modelo elipsoidal WGS 84 y tiene su origen y ejes, mostrados en la *Figura 16*, definidos como:

- El origen (denotado por  $O_n$ ) se fija arbitrariamente en un punto de la superficie terrestre.
- El eje x (denotado por  $X_n$ ) apunta hacia el norte geodésico.
- El eje y (denotado por  $Y_n$ ) apunta hacia el este geodésico.
- El eje z (denotado por  $Z_n$ ) apunta hacia abajo de la normal del elipsoide.

El sistema NED juega un papel muy importante en el control de vuelo y navegación. El vector de posición en el sistema de coordenadas NED se denota por:

$$P_n = \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} \quad (42)$$

### 5.1.4. **Sistema de coordenadas cuerpo (BODY)**

El sistema de coordenadas cuerpo se define directamente sobre el cuerpo del vehículo. Su origen y ejes, mostrados en la *Figura 17*, vienen dados como:

- El origen (denotado por  $O_b$ ) se localiza en el centro de gravedad del vehículo.
- El eje x (denotado por  $X_b$ ) apunta hacia delante del vehículo.
- El eje y (denotado por  $Y_b$ ) apunta hacia el lado derecho del vehículo.

- El eje  $z$  (denotado por  $Z_b$ ) apunta hacia abajo siguiendo la regla de la mano derecha.

## 5.2. Transformación de coordenadas

Una vez descritos los sistemas de coordenadas en los cuales se va a trabajar es importante presentar cómo se realiza la transformación entre unos y otros, para ello existen varios métodos que se presentarán a continuación.

### 5.2.1. DCM

La matriz de direcciones coseno es una matriz que permite cambiar un sistema de coordenadas llamémosle “b” a otro marco coordenado “a”.

$$C_b^a = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (43)$$

Cada componente de esta matriz es uno de los cosenos de los ángulos entre los ejes de los dos sistemas de coordenadas por lo que acaba siendo una matriz de cambio de coordenadas con la cual para transformar un vector en el plano “b” a el plano “a” se realizaría lo siguiente:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^a = C_b^a \begin{bmatrix} x \\ y \\ z \end{bmatrix}^b \quad (44)$$

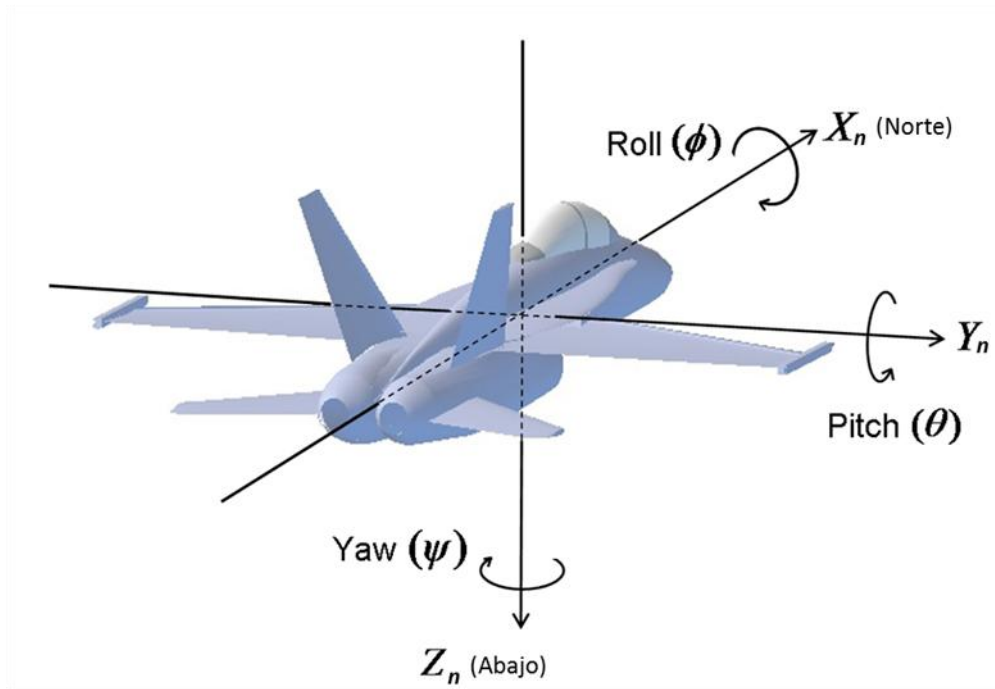
Al igual que todas las matrices de cambio de coordenadas cumple con un par de propiedades:

$$\text{Det}(C_b^a) = 1 \quad (45)$$

$$(C_b^a)^{-1} = (C_b^a)^T = C_a^b \quad (46)$$

### 5.2.2. Ángulos de Euler

Otro método bastante popular para especificar la orientación angular de un sistema de coordenadas respecto a otro es el uso de los tres ángulos de Euler. Tres ángulos que, mediante una sucesión ordenada de giros, definen el cambio de un sistema de coordenadas a otro. Los ángulos de Euler ( $\phi$ ,  $\theta$ ,  $\psi$ ) corresponden con los ángulos convencionales de *roll* ( $\phi$ ), *pitch* ( $\theta$ ), *yaw* ( $\psi$ ) que se utilizan en navegación para especificar la actitud de un móvil. Tomado un avión como ejemplo, el ángulo de *roll* ( $\phi$ ) es el ángulo que forman las alas de éste con el eje X del sistema de referencia respecto al que rota, el ángulo de *pitch* ( $\theta$ ) es la inclinación del morro y el ángulo de *yaw* ( $\psi$ ) es el giro del morro del avión respecto al norte. Esta representación se muestra en la *Figura 18*.



*Figura 18: Coordenadas NED y ángulos de Euler*

Los ángulos de Euler deben de calcularse siempre realizando las rotaciones en el mismo orden ya que los mismos ángulos, si son rotados en órdenes diferentes, dan lugar a transformaciones diferentes.

### **5.2.3. Cambio de ejes Cuerpo a NED con los ángulos de Euler y viceversa**

El origen de coordenadas cuerpo (o Body) es el centro de masas del móvil, en nuestro caso, del LIDAR. Esta transformación es una rotación de los ejes, totalmente ordinaria, pero se deben definir bien estos giros. En este sistema de coordenadas cuerpo, el eje  $X_b$  apunta hacia delante del LIDAR, el eje  $Y_b$  apunta a la izquierda y el eje  $Z_b$  apunta hacia abajo.

Estos ejes corresponden a los ejes de roll, pitch y yaw del vehículo. Los signos de estos ángulos estarán definidos tal que el ángulo de roll  $\phi$  es positivo cuando el móvil se mueva por debajo del plano de horizonte, el ángulo de pitch  $\theta$  es positivo cuando el morro se eleva por encima de la línea de horizonte y el ángulo de yaw  $\psi$  se define positivo cuando el morro está rotando desde el norte hacia el este.

En la *Figura 19* se muestra una comparación de los ejes de coordenadas NED con respecto a los ejes cuerpo, en rojo encontramos el sentido en el que los ángulos son positivos.

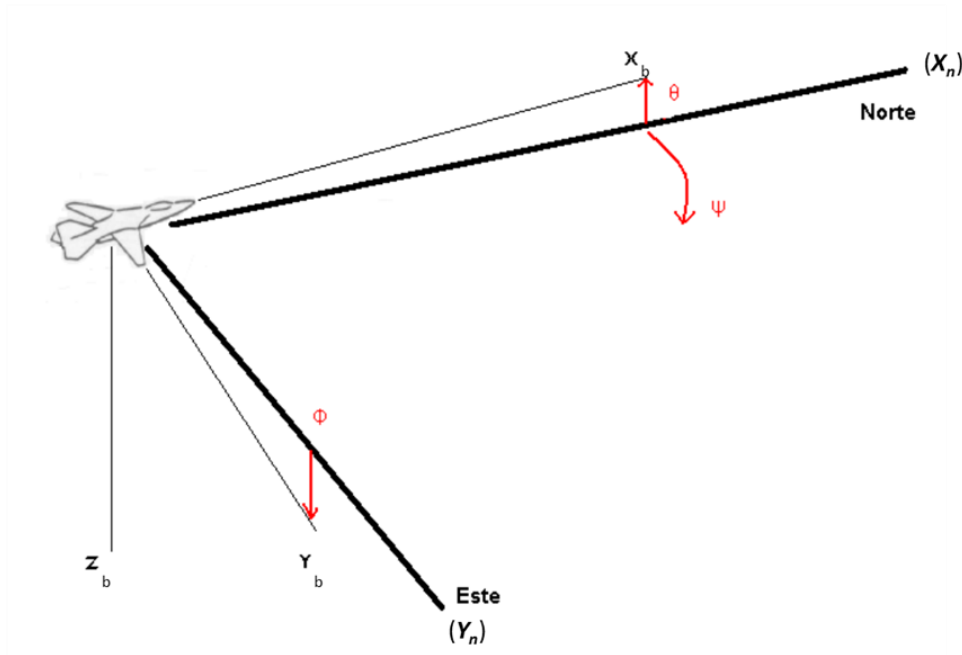


Figura 19: Coordenadas NED (negrita) respecto a ejes cuerpo  $x_b y_b z_b$

A continuación se muestra cual es el orden utilizado en el proceso de estimación de este estimador para calcular la matriz de rotación.

El primer giro se realiza en yaw:

$$C_\psi = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (47)$$

Se muestra en la Figura 20.

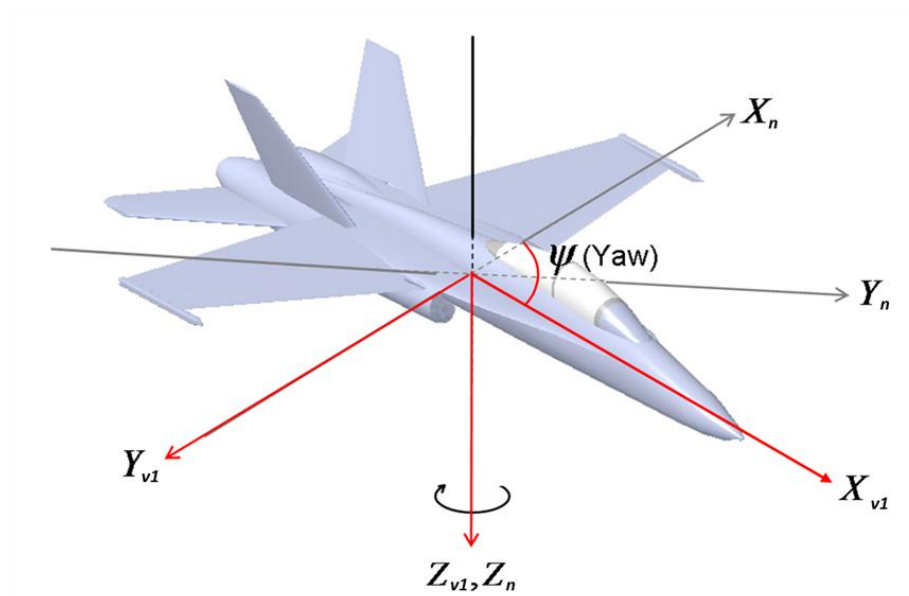
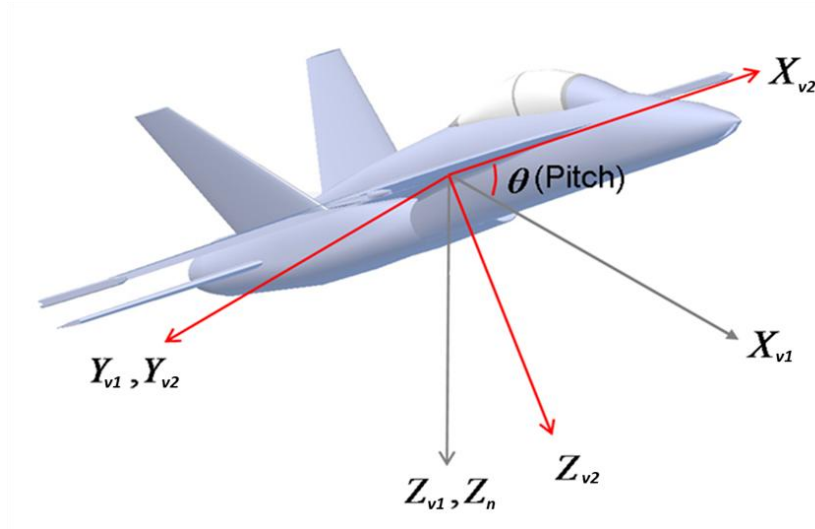


Figura 20: Giro en yaw

El segundo giro se realiza en pitch:

$$C_{\theta} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \quad (48)$$

cuyo resultado se muestra en la *Figura 21*.

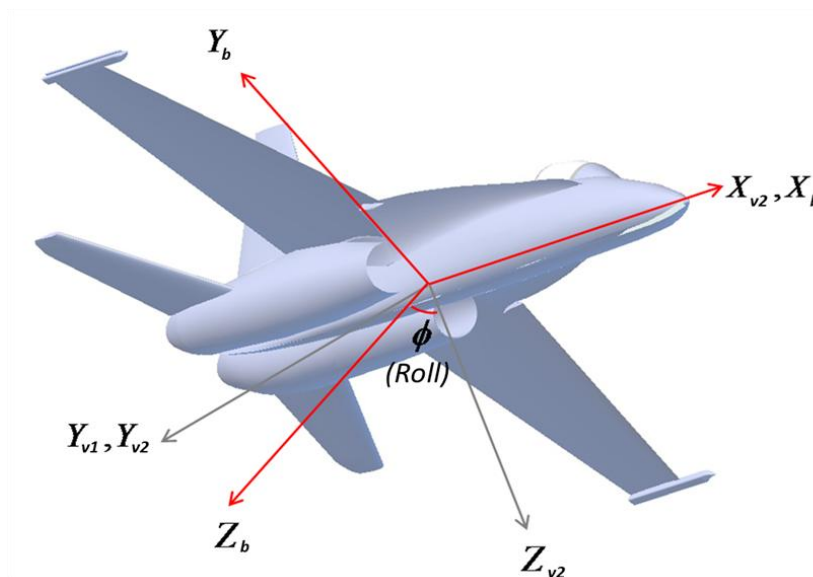


*Figura 21: Giro en pitch*

Y por último realizamos el giro en roll:

$$C_{\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \quad (49)$$

Su resultado gráfico se muestra en la



*Figura 22: Giro en roll*



De esta manera, ya se puede definir la matriz de cambio de coordenadas NED a ejes cuerpo de la siguiente manera:

$$C_n^b = C_\psi C_\theta C_\phi \quad (50)$$

$$C_n^b = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \sin \phi \\ \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (51)$$

Por ello, el cambio a realizar para pasar de coordenadas NED a ejes cuerpo es:

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = C_n^b \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \quad (52)$$

Teniendo en cuenta las propiedades de las matrices de rotación anteriormente citadas nos encontramos con que la matriz de cambio de coordenadas desde ejes cuerpo a NED se puede calcular de la siguiente manera:

$$R_b^n = (C_n^b)^T \quad (53)$$

$$R_b^n = \begin{bmatrix} \cos \theta \cos \psi & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \cos \theta \sin \psi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (54)$$

En resumen, para cambiar de ejes cuerpo a NED habría que aplicar la siguiente transformación:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = R_b^n \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (55)$$

#### 5.2.4. Cambio de coordenadas LLA a ECEF

El vector de transformación de posiciones en el sistema geodésico al sistema coordenado ECEF es un paso intermedio para convertir la medida de la posición dada por un GPS al sistema de coordenadas NED. Dado un punto en coordenadas geodésicas (LLA), es decir,

$$P_g = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix}$$

puede obtenerse el punto expresado en coordenadas ECEF como

$$P_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix} = \begin{pmatrix} (N_E + h) \cos \varphi \cos \lambda \\ (N_E + h) \cos \varphi \sin \lambda \\ [N_E(1 - e^2) + h] \sin \varphi \end{pmatrix} \quad (56)$$

donde  $e$  y  $N_E$  vienen determinadas por las expresiones dadas en (37) y (40), respectivamente.

### 5.2.5. Cambio de coordenadas ECEF a LLA

El paso inverso al anterior es un poco más complejo. Dado un punto en coordenadas ECEF, es decir,

$$P_e = \begin{pmatrix} x_e \\ y_e \\ z_e \end{pmatrix}$$

una de las técnicas más precisas para convertir dicho punto a coordenadas geodésicas (LLA) se compone de los siguientes cálculos:

$$r = \sqrt{x_e^2 + y_e^2} \quad (57)$$

$$E^2 = R_{Ea}^2 - R_{Eb}^2 \quad (58)$$

$$F = 54R_{Eb}^2 z_e^2 \quad (59)$$

$$G = r^2 + (1 - e^2)z_e^2 - e^2 E^2 \quad (60)$$

$$C = \frac{e^4 F r^2}{G^3} \quad (61)$$

$$S = \sqrt[3]{1 + C + \sqrt{C^2 + 2C}} \quad (62)$$

$$P = \frac{F}{3\left(s + \frac{1}{s} + 1\right)^2 G^2} \quad (63)$$

$$Q = \sqrt{1 + 2e^4 P} \quad (64)$$

$$r_0 = \frac{-(Pe^2 r)}{1+Q} + \sqrt{\frac{1}{2}a^2 \left(1 + \frac{1}{Q}\right) - \frac{P(1-e^2)z_e^2}{Q(1+Q)} - \frac{1}{2}Pr^2} \quad (65)$$

$$U = \sqrt{(r - e^2 r_0)^2 + z_e^2} \quad (66)$$

$$V = \sqrt{(r - e^2 r_0)^2 + (1 - e^2)z_e^2} \quad (67)$$

$$Z_0 = \frac{R_{Eb}^2 z_e}{R_{Ea} V} \quad (68)$$

Las coordenadas del punto expresadas en LLA son:

$$P_g = \begin{pmatrix} \lambda \\ \varphi \\ h \end{pmatrix}$$

donde:

$$\lambda = \arctan2[y_e, x_e] \quad (69)$$

$$\varphi = \arctan \left[ \frac{z_e + e' Z_0}{r} \right] \quad (70)$$

$$h = U \left( 1 - \frac{R_{Eb}^2}{R_{Ea} V} \right) \quad (71)$$

### 5.2.6. Cambio de coordenadas ECEF a NED

Para una transformación completa de coordenadas geodésicas (LLA) a NED, es necesario hacer una transformación del sistema geodésico (LLA) a ECEF (como se vio en el apartado 5.2.4) junto a la transformación de la posición en coordenadas ECEF a NED. De forma más específica, se tiene:

$$P_n = R_e^n (P_e - P_{e,ref}) \quad (72)$$

donde  $P_{e,ref}$  es la posición del origen del sistema NED (es decir,  $O_n$ ) expresado en el sistema de coordenadas ECEF, y  $R_e^n$  es la matriz de cambio de coordenadas ECEF a NED, la cual viene dada por

$$R_e^n = \begin{bmatrix} -\sin \varphi_{ref} \cos \lambda_{ref} & -\sin \varphi_{ref} \sin \lambda_{ref} & \cos \varphi_{ref} \\ -\sin \lambda_{ref} & \cos \lambda_{ref} & 0 \\ -\cos \varphi_{ref} \cos \lambda_{ref} & -\cos \varphi_{ref} \sin \lambda_{ref} & -\sin \varphi_{ref} \end{bmatrix} \quad (73)$$

siendo  $\lambda_{ref}$  y  $\varphi_{ref}$  la longitud y latitud geodésica correspondiente al punto  $P_{e,ref}$ .

### 5.2.7. Cambio de coordenadas NED a ECEF

Despejando  $P_e$  de la ecuación (72), se obtiene el paso inverso:

$$P_e = (R_e^n)^{-1} P_n + P_{e,ref} = C_n^e P_n + P_{e,ref} \quad (74)$$

donde  $P_{e,ref}$  es la posición del origen del sistema NED (es decir,  $O_n$ ) expresado en el sistema de coordenadas ECEF, y  $C_n^e$  es la matriz de cambio de coordenadas NED a ECEF, la cual viene dada por

$$C_n^e = \begin{bmatrix} -\sin \varphi_{ref} \cos \lambda_{ref} & -\sin \lambda_{ref} & -\cos \varphi_{ref} \cos \lambda_{ref} \\ -\sin \varphi_{ref} \sin \lambda_{ref} & \cos \lambda_{ref} & -\cos \varphi_{ref} \sin \lambda_{ref} \\ \cos \varphi_{ref} & 0 & -\sin \varphi_{ref} \end{bmatrix} \quad (75)$$

siendo  $\lambda_{ref}$  y  $\varphi_{ref}$  la longitud y latitud geodésica correspondiente al punto  $P_{e,ref}$ .

Como era de esperar, las matrices de cambio de coordenadas,  $C_n^e$  y  $R_e^n$ , verifican la propiedad de la ecuación (53):

$$C_n^e = (R_e^n)^{-1} = (R_e^n)^T$$

### 5.3. Implementación del cambio de coordenadas NED a LLA

La posición de los objetos devuelta por la clase background explicada en el Capítulo 4 está dada en el sistema de referencia Cuerpo centrado en el origen del LIDAR. Para realizar los cambios de coordenadas citados en el apartado anterior se ha creado una nueva clase, denominada *coordinate\_converter*.

Al constructor de esta nueva clase (*conversor*) se le llamará desde la función *print\_ob* (explicada en el apartado 4.2.8) y se procederá al cambio de coordenadas a través de la llamada a la función *newConversion*. Las funciones de esta nueva clase se detallan en los apartados siguientes y su programación en C++ en el Capítulo 6.

#### 5.3.1. Constantes

Para el desarrollo de esta clase es necesario definir las siguientes constantes:

Nombre	Valor	Descripción
PI	3.14159265	Valor de la constante $\pi$
LAT_REF	37.0	Latitud de referencia geodésica (en grados) correspondiente al origen del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
LON_REF	-5.0	Longitud de referencia geodésica (en grados) correspondiente al origen del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
H_REF	0.0	Altura de referencia geodésica (en metros) correspondiente al origen del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
PITCH	0.0	Ángulo pitch (en grados) del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
YAW	0.0	Ángulo yaw (en grados) del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
ROLL	180.0	Ángulo roll (en grados) del LIDAR. Deberá comprobarse su valor una vez colocado el dispositivo y antes de iniciar la ejecución del programa
deg2rad	PI/180.0	Factor de conversión de grados a radianes

*Tabla 12: Constantes necesarias para la conversión de coordenadas*

Como se observa en la *Tabla 12*, el algoritmo se ha programado suponiendo que se encuentra en las coordenadas de referencia (-5°, 37°, 0) y que se encuentra alineado con el sistema NED, es decir, que los ángulos pitch y yaw son 0° y roll es 180°. Para que el sistema cuerpo esté alineado con NED el ángulo roll tiene que ser 180° puesto que el eje  $y_b$  del LIDAR tiene sentido contrario al que aparece en la *Figura 19*.

### 5.3.2. Conversor

Su formato es:

*conversor (const double refLon, const double refLat, const double refH,  
const double pitch, const double yaw, const double roll)*

Esta función es el constructor y se ejecuta una única vez (la primera). Sirve para inicializar a un determinado tamaño las variables globales de esta clase, en concreto:

Nombre	Descripción
ned	Vector de tres elementos que almacena las coordenadas del sistema NED: $[x_n, y_n, z_n]$
ecef	Vector de tres elementos que almacena las coordenadas del sistema ECEF: $[x_e, y_e, z_e]$
lla	Vector de tres elementos que almacena las coordenadas del sistema LLA: $[\lambda, \varphi, h]$
body	Vector de tres elementos que almacena las coordenadas del sistema cuerpo: $[x_b, y_b, z_b]$

*Tabla 13: Vectores inicializados en conversor*

Y guardar las medidas de referencia dadas por los parámetros de entrada (que no varían durante la ejecución del programa):

Parámetro de entrada	Variable almacenada	Descripción
refLon	_refLon	Longitud de referencia geodésica (en radianes) correspondiente al origen del LIDAR
refLat	_refLat	Latitud de referencia geodésica (en radianes) correspondiente al origen del LIDAR
refH	_refH	Altura de referencia geodésica (en metros) correspondiente al origen del LIDAR
pitch	_pitch	Ángulo pitch (en radianes) del LIDAR
yaw	_yax	Ángulo yaw (en radianes) del LIDAR
roll	_roll	Ángulo roll (en radianes) del LIDAR

*Tabla 14: Parámetros de entrada y variables en las que se guardan*

### 5.3.3. NewConversion

Cada vez que se obtiene una nueva posición de un objeto detectado por el LIDAR se llama a esta función que es la que a su vez llama a las funciones necesarias para conseguir transformar de un sistema de coordenadas a otro.

Su formato es:

*std :: vector < double > newConversion (const double x,  
const double y, const double z)*

Los parámetros de entrada son precisamente cada una de las posiciones de los objetos en el sistema de referencia del LIDAR (cuerpo). Como parámetro de salida devuelve un vector (de tres elementos) que contiene la posición del objeto expresada en el sistema de referencia LLA.

Lo primero que tiene que hacer la función es pasar del sistema de referencia cuerpo, que es en el que viene dados los parámetros de entrada, al sistema de referencia NED. Esto se consigue mediante la llamada a la función *body2ned*. Una vez obtenida la posición del objeto expresada en coordenadas NED hay que pasar al sistema intermedio ECEF, con la función *ned2ecef*. Finalmente, se transformada de coordenadas ECEF a LLA mediante la función *ecef2lla*.

Cada una de estas funciones se verá de forma más detallada en los siguientes apartados.

#### 5.3.4. *Body2ned*

Su formato es:

```
void body2ned (const double x, const double y, const double z)
```

Esta función convierte de coordenadas en el sistema cuerpo a NED.

Como parámetros de entrada esta función recibe las tres coordenadas de un objeto expresada en el sistema cuerpo (body). Tras aplicar las transformaciones especificadas en el apartado 5.2.3 se obtienen las coordenadas de este objeto expresadas en el sistema NED. En concreto, estas transformaciones eran las siguientes:

$$x_n = \cos \theta \cos \psi x_b + (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) y_b + (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) z_b$$

$$y_n = \cos \theta \sin \psi x_b + (\cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi) y_b + (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) z_b$$

$$z_n = -\sin \theta x_b + \cos \theta \sin \phi y_b + \cos \theta \cos \phi z_b$$

Siendo  $\theta$ ,  $\psi$  y  $\phi$  los ángulos pitch, yaw y roll respectivamente.

Estos resultados  $(x_n, y_n, z_n)$  se almacenan en la variable global especificada para ello, es decir, en el vector *ned*.

#### 5.3.5. *Ned2ecef*

Su formato es:

```
void ned2ecef()
```

Esta función convierte de coordenadas NED a ECEF.

Como se observa en su formato, no necesita parámetros de entrada porque la posición del objeto en coordenadas NED ya está almacenada como una variable global de esta clase.

Tras aplicar las transformaciones especificadas en el apartado 5.2.7 se obtienen las coordenadas de este objeto expresadas en el sistema ECEF. En concreto, estas transformaciones eran las siguientes:

$$x_e = -\sin \varphi_{ref} \cos \lambda_{ref} x_n - \sin \lambda_{ref} y_n - \cos \varphi_{ref} \cos \lambda_{ref} z_n + x_{e,ref}$$

$$y_e = -\sin \varphi_{ref} \sin \lambda_{ref} x_n + \cos \lambda_{ref} y_n + \cos \lambda_{ref} z_n + y_{e,ref}$$

$$z_e = \cos \varphi_{ref} x_n - \sin \varphi_{ref} z_n + z_{e,ref}$$

Donde  $x_{e,ref}$ ,  $y_{e,ref}$ ,  $z_{e,ref}$  es la posición del origen del sistema NED (es decir, el origen del LIDAR) expresado en el sistema de coordenadas ECEF. Este punto se obtiene transformando el punto de referencia, que viene expresado en coordenadas LLA, a coordenadas ECEF mediante la función *lla2ecef* (pasándole como parámetros de entrada *\_refLon*, *\_refLat* y *\_refH*).

El resultado final ( $x_e, y_e, z_e$ ) se almacena en la variable global especificada para ello, es decir, en el vector *ecef*.

### 5.3.6. *Ecef2lla*

Su formato es:

*void ecef2lla ()*

Esta función convierte de coordenadas ECEF a geodésicas (LLA).

Como se observa en su formato, necesita parámetros de entrada porque la posición del objeto en coordenadas ECEF ya está almacenada como una variable global de esta clase.

Tras aplicar las transformaciones especificadas en el apartado 5.2.5 se obtienen las coordenadas de este objeto expresadas en el sistema LLA. En concreto, estas transformaciones eran las siguientes:

$$\lambda = \arctan2[y_e, x_e]$$

$$\varphi = \arctan \left[ \frac{z_e + e' Z_0}{r} \right]$$

$$h = U \left( 1 - \frac{R_{Eb}^2}{R_{Ea} V} \right)$$

Tras dicha transformación,  $\lambda$  y  $\varphi$  están en radianes, por lo que se expresa en grados aplicando la transformación:

$$ang(^{\circ}) = ang(rad) \frac{180^{\circ}}{\pi} \quad (76)$$

El resultado final ( $\lambda, \varphi, h$ ) se almacena en la variable global especificada para ello, es decir, en el vector *lla*.

### 5.3.7. *Lla2ecef*

Su formato es:

*void lla2ecef (const double lon, const double lat, const double h)*

Esta función convierte de coordenadas geodésicas (LLA) a ECEF.

Como se observa en su formato, recibe como parámetros de entrada un punto expresado en coordenadas LLA, es decir, recibe la longitud ( $\lambda$ ), latitud ( $\varphi$ ) y altura ( $h$ ) de dicho punto en radianes. Tal y como se especificó en el apartado 5.3.5, es utilizada para transformar el punto de referencia, dado en coordenadas LLA, a coordenadas ECEF.

Tras aplicar las transformaciones especificadas en el apartado 5.2.4 se obtienen las coordenadas de este objeto expresadas en el sistema ECEF. En concreto, estas transformaciones eran las siguientes:

$$x_e = (N_E + h) \cos \varphi \cos \lambda$$

$$y_e = (N_E + h) \cos \varphi \sin \lambda$$

$$z_e = U[N_E(1 - e^2) + h] \sin \varphi$$

El resultado final ( $x_e, y_e, z_e$ ) se almacena en la variable global especificada para ello, es decir, en el vector *ecef*.

### 5.3.8. *Ecef2ned*

Su formato es:

*void ecef2ned (const double Xe, const double Ye, const double Ze)*

Esta función convierte de coordenadas ECEF a NED.

Como se observa en su formato, como parámetro de entrada esta función recibe las tres coordenadas de un objeto expresada en el sistema ECEF. Tras aplicar las transformaciones especificadas en el apartado 5.2.6 se obtienen las coordenadas de este objeto expresadas en el sistema NED. En concreto, estas transformaciones eran las siguientes:

$$x_n = -\sin \varphi_{ref} \cos \lambda_{ref} (x_e - x_{e,ref}) - \sin \varphi_{ref} \sin \lambda_{ref} (y_e - y_{e,ref}) \\ + \cos \varphi_{ref} (z_e - z_{e,ref})$$

$$y_n = -\sin \lambda_{ref} (x_e - x_{e,ref}) + \cos \lambda_{ref} (y_e - y_{e,ref})$$

$$z_n = -\cos \varphi_{ref} \cos \lambda_{ref} (x_e - x_{e,ref}) - \cos \varphi_{ref} \sin \lambda_{ref} (y_e - y_{e,ref}) \\ - \sin \varphi_{ref} (z_e - z_{e,ref})$$

Donde  $x_{e,ref}, y_{e,ref}, z_{e,ref}$  es la posición del origen del sistema NED (es decir, el origen del LIDAR) expresado en el sistema de coordenadas ECEF. Este punto se obtiene transformando el



punto de referencia, que viene expresado en coordenadas LLA, a coordenadas ECEF mediante la función *lla2ecef* (pasándole como parámetros de entrada *\_refLon*, *\_refLat* y *\_refH*).

Estos resultados  $(x_n, y_n, z_n)$  se almacenan en la variable global especificada para ello, es decir, en el vector *ned*.

Esta función no es llamada en ninguna ocasión durante la ejecución del programa pero ha sido utilizada para comprobar que las transformaciones de unas coordenadas a otras eran correctas.

### 5.3.9. *Ned2body*

Su formato es:

*void ned2body ()*

Esta función convierte de coordenadas NED al sistema cuerpo.

Como se observa en su formato, necesita parámetros de entrada porque la posición del objeto en coordenadas NED ya está almacenada como una variable global de esta clase.

Tras aplicar las transformaciones especificadas en el apartado 5.2.3 se obtienen las coordenadas de este objeto expresadas en el sistema cuerpo. En concreto, estas transformaciones eran las siguientes:

$$x_b = \cos \theta \cos \psi x_n + \cos \theta \sin \psi y_n - \sin \theta z_n$$

$$y_b = (\cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi) x_n + (\cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi) y_n + \cos \theta \sin \phi z_n$$

$$z_b = (\cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi) x_n + (\sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi) y_n + \cos \theta \cos \phi z_n$$

Siendo  $\theta$ ,  $\psi$  y  $\phi$  los ángulos pitch, yaw y roll respectivamente.

El resultado final  $(x_b, y_b, z_b)$  se almacena en la variable global especificada para ello, es decir, en el vector *body*.

Esta función no es llamada en ninguna ocasión durante la ejecución del programa pero ha sido utilizada para comprobar que las transformaciones de unas coordenadas a otras eran correctas.

