Rotations In KM3NeT

V. Kulikovskiy, M. de Jong

November 17, 2026

1 Introduction

The current note explains how DOM orientation is evaluated using Accelerometer and Magnetometer measurements implemented in the so-called 6D Tilt and Compass devices. The KM3NeT collaboration uses several 6D devices:

1. AHRS gdrive link. This is common for CLBv2 boards. This is actually a 9D since it has also a gyroscope but we don't use it for simplicity.
2. LSM303D
3. LSM303AGR (most recent CLBs, CLBv4 for example).

In following we will call all of them AHRS for simplicity (Attitude and Heading Reference System),

1.1 Coordinates

KM3NeT uses right-handed coordinate systems. The default coordinate system is UTM where X-axis is UTM Easting, and the Y-axis, is UTM Northing, the Z-axis corresponds to the vertical direction, i.e. from the center of the Earth upwards with its zero value at the sea surface[1].

In following it will be useful to use intermediate KM3NeT magnetic coordinate system with:

- X-axis (Magnetic) East,
- Y-axis Magnetic North,
- Z-axis Up.

as well as KM3NeT geographic coordinate system:

- X-axis Geographic East,
- Y-axis Geographic North,
- Z-axis Up.

1.1.1 UTM convergence angle

Convergence is the angular difference between true North (angle towards north pole) and UTM Grid North. In the case of the ANTARES detector (centre of the 12-Line detector is at $42°47.935'$N and $6 \circ 09.942'$E, this correction angle amounts to -1.924846 degrees, i.e. UTM Northing points slightly towards the geographical West direction.

Conversion from Geographic yaw angle (from East to North) to UTM yaw angle (from Easting to Northing) is the following:

```
yaw_utm = yaw_geo + convergence
```

The convergence angle calculation is implemented in km3astro.coord.

[1] More details are given in KM3NeT_SOFT_WD_2016_002.

1.1.2 Magnetic Declination

Digital compasses consist of 3 magnetometers aligned with the system axes (X,Y,Z). The compass principal direction, magnetic north, is thus, different from the true north by quantity named magnetic declination.

Magnetic declination definition: the angle of difference between true North and magnetic North. For instance, if the declination at a certain point were 10° W, then a compass at that location pointing north (magnetic) would actually align 10° W of true North. True North would be 10° E relative to the magnetic North direction given by the compass. Declination varies with location and slowly changes in time [NOAA]. To calculate magnetic declination of your site, use: [NOAA]. Additional convention: "easterly" is positive, i.e. declination 10° W is a -10 correction.

Conversion from KM3NeT magnetic yaw angle (from Magnetic East to Magnetic North) to Geographic yaw angle (from East to North) is the following:

```
yaw_geo = yaw_mag - magnetic_declination
```

The magnetic declinations for known coordinates can be obtained from NOAA website. See example here how it was produced for ANTARES, ARCA and ORCA sites: TiltCompass git project.

1.2 Active/passive

An active transformation is a transformation which actually changes the physical position (alibi, elsewhere) of a point, or rigid body, which can be defined in the absence of a coordinate system; whereas a passive transformation is merely a change in the coordinate system in which the object is described (alias): change of coordinate map, or change of basis. In KM3NeT the convention is physics driven, for example:

- Tilt and Compass (TC) measurements (yaw-pitch-roll) are evaluated as passive rotations of the constant field vectors and rotating measurements device (accelerometer, magnetometer),
- DOM (and its elements, for example PMT orientations) for dynamic positions are evaluated as active rotations of the nominal orientations (i.e. DOM oriented perfectly vertical or PMT in the bottom having orientation (0,0,-1) etc).

Mathematically if active rotation is done with R matrix then passive rotation is done with R^{-1} matrix.

1.3 Rotation order - intrinsic/extrinsic

In an intrinsic system, each of the elemental rotations is performed on the coordinate system as rotated by the previous operation(s). In an extrinsic system, each rotation is performed around the axes of the world coordinate system, which does not move. In KM3NeT we use active extrinsic (body rotates axes fixed) or passive intrinsic (coordinates rotate and they rotate each time around newly produced coordinates axes).

In the general case, any intrinsic rotation can be converted to its extrinsic equivalent and vice-versa by reversing the order of elemental rotations.[2]

1.4 Euler (Tait–Bryan) angles yaw, pitch, roll

For the following we will need to define three principal rotations (see Fig. 1):

yaw rotation around z-axis (by angle ϕ from x-axis towards y-axis),

pitch rotation around y-axis (by angle θ from z-axis towards x-axis)

roll rotation around x-axis (by angle ψ from y-axis towards z-axis)

Note that for now we just defined the axes of rotations. What will make these angles Tait-Bryan angles later (and not proper or classic Euler angles) is the order in which rotations are performed (Section 3.2).

2 Matrices

Convention: vector coordinates defined as columns and matrices applied leftwards:

$$X = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

[2] No proof found - taken from here and some discussions here: here.

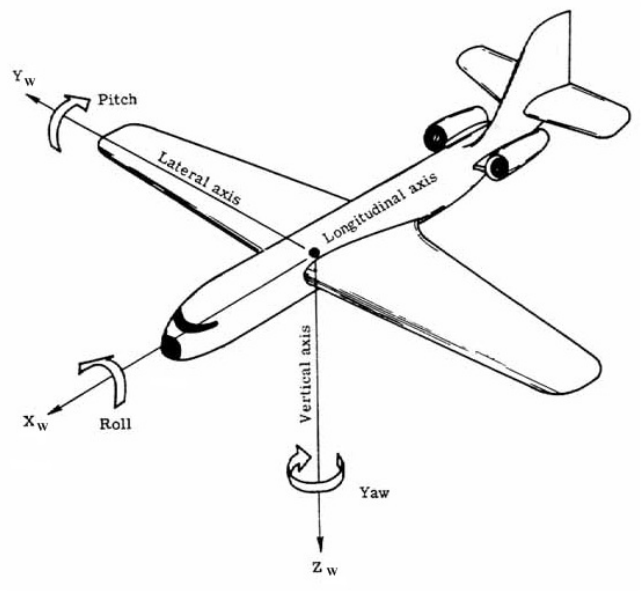


Figure 1: Yaw, pitch, roll angles

and

$$X' = \begin{pmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a11 \times x + a12 \times y + a13 \times z \\ a21 \times x + a22 \times y + a23 \times z \\ a31 \times x + a32 \times y + a33 \times z \end{pmatrix} \tag{2}$$

2.1 Passive rotations

Yaw:

$$Y = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

Pitch:

$$P = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \tag{4}$$

Roll:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \tag{5}$$

Cross-check: vector pointing towards x-axis appears to be rotated by $-\phi$ once axes are rotated from x to y around z by ϕ (passive yaw rotation):

$$X' = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \cos\phi \\ -\sin\phi \\ 0 \end{pmatrix} \tag{6}$$

2.2 Active rotations

Yaw:

$$Y^{-1} = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{7}$$

Pitch:

$$P^{-1} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{8}$$

Roll:

$$R^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \tag{9}$$

2.3 Combining rotations

Passive intrinsic rotation (DOM AHRS rotates while Earth magnetic field and gravity remains the same). The yaw-pitch-roll sequence is applied on the vector:

$$X' = R(P(YX)) = RPYX \tag{10}$$

Active extrinsic rotation (PMT is rotated in the DOM having rotation defined with yaw-pitch-roll angles):

$$X' = Y^{-1}P^{-1}R^{-1}X = (RPY)^{-1}X \tag{11}$$

Note that:

$$(RPY)^{-1} \times (RPY) = 1$$

$$Y^{-1}P^{-1}(R^{-1}R)PY = Y^{-1}(P^{-1}P)Y = Y^{-1}Y = 1$$

$$(RPY)^{-1} = Y^{-1}P^{-1}R^{-1} \tag{12}$$

Here one can see that that between Eq. 10 and Eq. 11 matrices inversion comes from active/passive change and the order inversion appears naturally as the algebra properties of matrices (associative multiplication) and it corresponds to the extrinsic/intrinsic change.

3 Quaternions

Quaternion is $q = q_0 + iq_1 + jq_2 + kq_3 = (q0, \vec{q})$, where $\vec{q} = (q_1, q_2, q_3)$. Conjugation: $q^* = (q0, -\vec{q})$.

In KM3NeT for quaternions the Hamilton convention is used:

$$i^2 = k^2 = j^2 = ijk = -1, \tag{13}$$

while in the alternative, JPL convention $ijk = 1$.

The convention affects quaternion multiplication which is for the Hamilton convention:

$$(p0, \vec{p}) \times (q0, \vec{q}) = p_0q_0 - \vec{p} \cdot \vec{q} + p_0\vec{q} + q_0\vec{p} + \vec{p} \times \vec{q} \tag{14}$$

Quaternion norm:

$$|q| = \sqrt{q0^2 + q1^2 + q2^2 + q3^2} \tag{15}$$

3.1 Rotation with quaternion

Unit quaternions representation:

$$q = \begin{pmatrix} cos(\theta/2) \\ sin(\theta/2)cos(\beta_x) \\ sin(\theta/2)cos(\beta_y) \\ sin(\theta/2)cos(\beta_z) \end{pmatrix} \tag{16}$$

Rotation of the vector $\vec{v}$ through an angle θ about $\vec{u} = (cos(\beta_x), cos(\beta_y), cos(\beta_z))$ as the axis of rotation (active rotation)[1]:

$$v' = q \times v \times q^* \tag{17}$$

Here v is a quaternion $(0, \vec{v})$. This simplifies to:

$$\vec{t} = 2 * \vec{q} \times \vec{v} \tag{18}$$

$$\vec{v'} = \vec{v} + q_0 * \vec{t} + \vec{q} \times \vec{t} \tag{19}$$

where vector t is an intermediate vector used for above calculation.

Rotation of the coordinate frame about the axis $\vec{u} = (cos(\beta_x), cos(\beta_y), cos(\beta_z))$ through an angle θ while $\vec{v}$ is not rotated (passive rotation):

$$\vec{v'} = q^* \times \vec{v} \times q \tag{20}$$

3.2 Euler angles - quaternions rotation

For AHRS we use Tait–Bryan angles convention (Body 3-2-1 sequence) in which the airplane first does yaw (Body-Z) turn during taxiing onto the runway, then pitches (Body-Y) during take-off, and finally rolls (Body-X) in the air. This is consistent with wiki. Thus we can use the same conversion code.

```
struct Quaternion
{
    double w, x, y, z;
};

Quaternion ToQuaternion(double roll, double pitch, double yaw) // roll (x), pitch (y), yaw (z), angles
{
    // Abbreviations for the various angular functions

    double cr = cos(roll * 0.5);
    double sr = sin(roll * 0.5);
    double cp = cos(pitch * 0.5);
    double sp = sin(pitch * 0.5);
    double cy = cos(yaw * 0.5);
    double sy = sin(yaw * 0.5);

    Quaternion q;
    q.w = cr * cp * cy + sr * sp * sy;
    q.x = sr * cp * cy - cr * sp * sy;
    q.y = cr * sp * cy + sr * cp * sy;
    q.z = cr * cp * sy - sr * sp * cy;

    return q;
}
```

Note that later it will be shown how to recover from AHRS measurements the Euler angles that correspond to the active "Body 3-2-1 sequence" thus this quaternion can be used for active DOM elements rotation (PMT orientation etc).

3.3 Quaternions rotation - euler angles

Inverse rotation (from the same wiki):

```
    // roll (x-axis rotation)
    double sinr_cosp = 2 * (q.w * q.x + q.y * q.z);
    double cosr_cosp = 1 - 2 * (q.x * q.x + q.y * q.y);
    angles.roll = std::atan2(sinr_cosp, cosr_cosp);

    // pitch (y-axis rotation)
    double sinp = std::sqrt(1 + 2 * (q.w * q.y - q.x * q.z));
    double cosp = std::sqrt(1 - 2 * (q.w * q.y - q.x * q.z));
    angles.pitch = 2 * std::atan2(sinp, cosp) - M_PI / 2;

    // yaw (z-axis rotation)
    double siny_cosp = 2 * (q.w * q.z + q.x * q.y);
    double cosy_cosp = 1 - 2 * (q.y * q.y + q.z * q.z);
    angles.yaw = std::atan2(siny_cosp, cosy_cosp);
```

4 Existing implementations

4.1 6D compass-accelerometer measurements in the DOMs

The final AHRS coordinate system is the following:

- X aligned with magnetic North,
- Z aligned with direction towards center (gravity), i.e. "down",
- Y aligned with East since the system is right-handed.

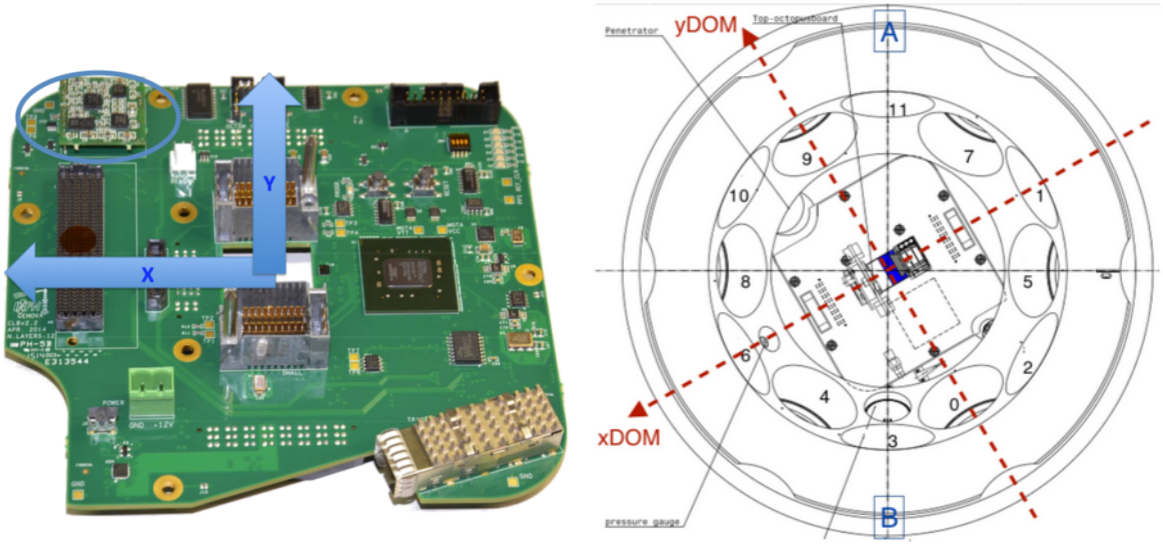


Figure 2: Axes of the CLB (v2 is shown) and DOM from [2]. For the DOM - this is the view of the open uppen hemisphere (so from inside the DOM towards the top).

4.1.1 Calibration and coordinate system unification

The AHRS device have their own coordinate system that differs from model to model (AHRS, LSM303D, LSM303AGR) and may even depend on the board configuration or embedded software. In the DB two 3x3 rotation matrices, R_A, R_H, are provided to convert accelerometer measurements A_x, A_y, A_z and magnetometer measurements H_x, H_y, H_z to the coordinate system aligned with CLB axes:

- X indicated in Fig. 3,
- Z looking down.

coordinates like in North-Eeast-Down (NED), used in aerospace).

DOM axes [3]:

In case of magnetometer this matrix corrects also non-linearities in the magnetometer measurements (soft-iron). Additionally, offsets are applied to the raw values (in case of magnetometers this is a hard-iron correction). The values in the DOM AHRS-like coordinate system becomes as follows:

$$A = R_A(A_{\text{raw}} - A_{\text{offset}}) \tag{21}$$
$$H = R_H(H_{\text{raw}} - H_{\text{offset}}) \tag{22}$$

In the KM3NeT DB this is organized in following fields:

```
"AHRS_Vector_Index(-)"
"AHRS_Matrix_Column(-)"
"AHRS_Matrix_Row(-)"
"AHRS_Acceleration_Offset(g/ms^2-)"
"AHRS_Acceleration_Rotation(-)"
"AHRS_Magnetic_Rotation(-)"
"AHRS_Magnetic_XMin(G-)"
"AHRS_Magnetic_XMax(G-)"
"AHRS_Magnetic_YMin(G-)"
"AHRS_Magnetic_YMax(G-)"
"AHRS_Magnetic_ZMin(G-)"
"AHRS_Magnetic_ZMax(G-)"
```

The helper vectors are needed to establish the order of elements, here is the pseudocode:

```
VectorIndex = db("AHRS_Vector_Index(-)")
MatrixColumn = db("AHRS_Matrix_Column(-)")
```

[3] KM3NeT_DET_2014_002-PMT_map_v6

```
MatrixRow = db("AHRS_Matrix_Row(-)")

for i in [1..9]:
    R_A[MatrixRow[i], MatrixColumn[i]] = db("AHRS_Acceleration_Rotation(-)") [i]
    R_H[MatrixRow[i], MatrixColumn[i]] = db("AHRS_Magnetic_Rotation(-)") [i]

for i in [1..3]:
    A0 [VectorIndex[i]] = db("AHRS_Acceleration_Offset(g/ms^2-)") [i]

H0[0] = ( db("AHRS_Magnetic_XMin(G-)") + db("AHRS_Magnetic_XMax(G-)") ) /2.
H0[1] = ( db("AHRS_Magnetic_YMin(G-)") + db("AHRS_Magnetic_YMax(G-)") ) /2.
H0[2] = ( db("AHRS_Magnetic_X~Min(G-)" ) + db("AHRS_Magnetic_ZMax(G-)")) /2.
```

4.1.2 Transformation from A and H measurements to the orientation

The accelerometers measure acceleration of the device. For a steady device gravity makes the device feel like it is accelerated upwards, opposite to $\vec{g}$ direction.

The magnetometer measures both horizontal and vertical magnetic component (at KM3NeT latitudes the field vertical component is pointed downwards and it is slightly higher than the horisontal component). The AHRS aligned with X-axis at North and Z-down, thus measures $H = (H_{\mathrm{hor}}, 0, H_{\mathrm{vert}})$ both H_{hor} and H_{vert} are positive values.

By comparing axes in Fig. 3 one can see that $Y_{\mathrm{DOM}} = X_{\mathrm{AHRS}}$, while $X_{\mathrm{DOM}} = -Y_{\mathrm{AHRS}}$. Finally, Z_{AHRS} looks down, however, the board mounted upside-down so in DOM it looks up and coincides with Z_{DOM}. Thus, the following transformation should be performed:

```
A = np.array([-A[1],A[0],A[2]])
H = np.array([-H[1],H[0],H[2]])
```

Note that in these coordinates a DOM aligned horizontally with X at East, Y at North, Z up is measuring $A = (0, 0, 1)$, $H = (0, H_{\mathrm{hor}}, -H_{\mathrm{vert}})$.

During the rotation of AHRS device, the A and H measurements are changing their values. This is a passive rotation in which the vectors (acceleration, $-\vec{g}$, and direction to magnetic north) do not change, however the coordinate system that measures projections is changing. Thus the AHRS is measuring $X' = RPY\,X$ values.

For accelerometer initial vector this becomes:

$$A' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -\sin\theta \\ \cos\theta\sin\psi \\ \cos\theta\cos\psi \end{pmatrix} \quad (23)$$

From which:

```
Pitch = atan2(-A[0], sqrt(A[1]*A[1]+A[2]*A[2])
Roll = atan2(A[1], A[2])
```

Calculated in this way, they have "natural" ranges: pitch $\theta = [-\pi/2, \pi/2]$, roll $\psi = (-\pi, \pi]$.

The tilted compass measurements can be brought to horizontal compass measurements by "unroll, unpitch" operation this can be expressed as follows:

$$H_{\mathrm{mod}} = P^{-1}R^{-1}H = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{pmatrix} \begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix} = \begin{pmatrix} H_x\cos\theta + H_y\sin\theta\sin\psi + H_z\sin\theta\cos\psi \\ H_y\cos\psi - H_z\sin\psi \\ -H_x\sin\theta + H_y\cos\theta\sin\psi + H_z\cos\theta\cos\psi \end{pmatrix} \quad (24)$$

In the nominal position in KM3NeT magnetic coordinates the compass measures $H = (0, H_{\text{hor}}, -H_{\text{vert}})$ where H_{hor} and H_{vert} are positive numbers. During yaw passive transformation this changes as:

$$H_{\text{mod}} = \begin{pmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ H_{\text{hor}} \\ -H_{\text{vert}} \end{pmatrix} = \begin{pmatrix} H_{\text{hor}} \sin\phi \\ H_{\text{hor}} \cos\phi \\ -H_{\text{vert}} \end{pmatrix} \tag{25}$$

Thus:

```
yaw = atan2(Hmod[0],Hmod[1])
```

Note that calculations presented earlier are done in the AHRS axis system thus some signs etc is different (because nominal acceleration and magnetic vectors are different).

4.2 Summary of the calculations

```
//calibration

A_ahrs = R_A.cdot(A_raw-A0)
H_ahrs = R_H.cdot(H_raw-H0)

//from AHRS  to DOM

A = np.array([-A_ahrs[1],A_ahrs[0],A_ahrs[2]])
H = np.array([-H_ahrs[1],H_ahrs[0],H_ahrs[2]])

//Theta = pitch, psi = roll
Pitch = atan2(-A[0], sqrt(A[1]*A[1]+A[2]*A[2])
Roll = atan2(A[1], A[2])

cr = cos(Roll)
sr = sin(Roll)

cp = cos(Pitch)
sp = sin(Pitch)

//unpitch + unroll of the
Hmod[0] = H[0]*cp + H[1] * sp * sr + H[2]* sp * cr
Hmod[1] = H[1] * cr - H[2] * sr

yaw_mag = atan2(Hmod[0],Hmod[1])

#from yaw magnetic to geographic to UTM
yaw_geo = yaw_mag - magnetic_declination
yaw_utm = yaw_geo + convergence


//quaternions
cr = cos(roll * 0.5)
sr = sin(roll * 0.5)
cp = cos(pitch * 0.5)
sp = sin(pitch * 0.5)
cy = cos(yaw * 0.5)
sy = sin(yaw * 0.5)

q.w = cr * cp * cy + sr * sp * sy
q.x = sr * cp * cy - cr * sp * sy
q.y = cr * sp * cy + sr * cp * sy
q.z = cr * cp * sy - sr * sp * cy

#Cross-checks
q     = [q.w,q.x,q.y,q.z]
```

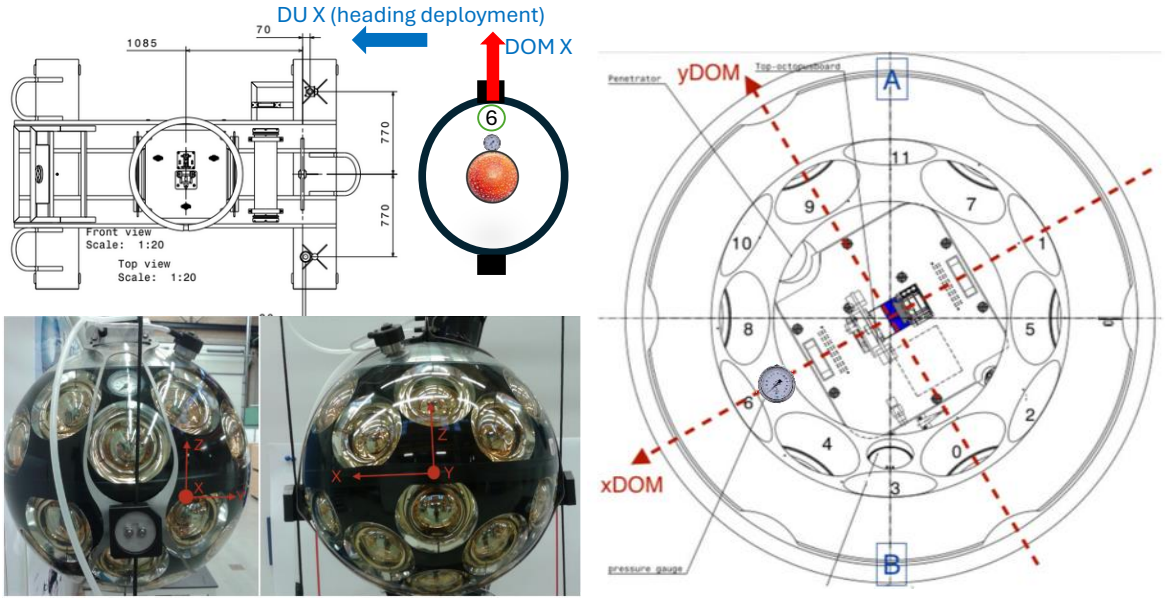


Figure 3: Heading measurement in ARCA.

```
qconj = [q.w,-q.x,-q.y,-q.z]

#Cross-check for A
vec = [0,0,0,1] #nominal A 4-vector
#passive rotation with quaternion
t = 2*np.cross(qconj[1:4],vec[1:4])
vecmod = vec[1:4]+q[0]*t+np.cross(qconj[1:4],t)
#comparison with measured values
print("TestA:",A,vecmod)

#Cross-check for H
vec = [0,0,0.12,-0.27] #nominal H 4-vector
#passive rotation with quaternion
t = 2*np.cross(qconj[1:4],vec[1:4])
#comparison with measured values - absolute values of H may not match
vecmod = vec[1:4]+q[0]*t+np.cross(qconj[1:4],t)
print("TestH:",H,vecmod)
```

4.3 DU deployment orientations

In ARCA the heading is provided as the angle from the North(ing) to the axis defined by the ROV mateable connector (matching with AHRS X axis) going easterly. While DOM rotation should be measured from East(ing) towards north. This means the nominal detector rotation in ARCA should be done by:

```
yaw_DU = -yaw_DU_deployment
```

For example, if DU heading is 0, then DU points North while DOMs point East so there is no need for a rotation. While if DU heading during marine campaign is 90° then DU points East, DOM points South, so its nominal direction should be rotated by −90°.

For ORCA the rotation is:

```
yaw_DU = -(180+yaw_DU_deployment)
```

4.4 Acronyms

See KM3NeT acronyms.

References

[1] Yan-Bin Jia, Quaternions, Com S 477/577 Notes

https://faculty.sites.iastate.edu/jia/foundations-robotics-and-computer-vision-com-s-477577.

[2] M. Circella, K. Graf, D. van Eijk, "Definition of DOM coordinate system and PMT map", KM3NeT_DET_2014_002-PMT_map_v6, gdrive.