THE APPLICATION OF NINE DEGREES OF FREEDOM SENSOR IN DETERMINATION OF TELESCOPE POSITION

D. S. VUJIČIĆ
Faculty of Technical Sciences in Čačak, Sv. Save 65, 32000 Čačak, Serbia
E–mail: dejan.vujicic@ftn.kg.ac.rs

Abstract. This paper shows the application of nine degrees of freedom sensor to determine the position of the telescope. Nine degrees of freedom sensor is consisted of three sensors: gyroscope, accelerometer, and magnetometer. By using DCM algorithm for sensor data fusion, Euler angles are calculated, upon which the coordinates of horizontal coordinate system are calculated as well. The coordinates of equatorial coordinate system are retrieved by using these coordinates together with time data and sensor location, which can be gotten either via GPS or by manually entering them. The PC application is written in C# programming language for retrieval of data from sensor and calculation of horizontal and equatorial coordinates. Furthermore, the errors and imprecisions that have emerged are explained, together with some of the ways of their elimination.

1. INTRODUCTION

This paper shows the problem of determination of celestial equatorial coordinates of the point the telescope is aiming at. In order to achieve that, the horizontal coordinates of the point have to be known, together with mean sidereal Greenwich time and geographic longitude and latitude of the place where the telescope is located. A nine degrees of freedom (9DOF) sensor is used to obtain these data. This sensor incorporates three sensors: gyroscope, accelerometer, and magnetometer which enable determination of the sensor plate orientation in the horizontal coordinate system. The data about exact time and geographic longitude and latitude are obtained with GPS receiver. However, if the environmental conditions do not allow sufficient precision of the GPS receiver, the time gotten from computer can be used, along with manually entered coordinates of the telescope position.

In order to determine celestial equatorial coordinates of the point, the PC application is written. It receives the sensor data via Bluetooth and then calculates the equatorial coordinates.

2. EULER ANGLES

Euler angles (yaw, pitch, and roll) are the most suitable form of representation of the rigid body position. They represent the rotations of the one coordinate system relative to the other coordinate system around all three axes. These two coordinate systems are world coordinate system (x, y, z), fixed in inertial space, and body-fixed coordinate system (X, Y, Z), which is related to the object and relative to the world coordinate system [Diebel 2006].

The rotation matrix $\mathbf{R}$ describes the orientation of one coordinate system relative to the other. The columns of this matrix are unit vectors of one coordinate system as seen from another. The vector in one coordinate system can be transformed to the vector in another coordinate system by multiplying it with the rotation matrix. The opposite transformation is done with transpose rotation matrix. The rotation matrix elements represent combinations of the trigonometrical functions of Euler angles.

The rotation matrix is also known as direction cosine matrix, since it is consisted of sines and cosines of the angles between the axes of the two coordinate systems.

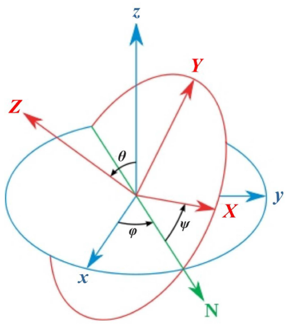

Figure 1: Geometrical definition of Euler angles.

In order to geometrically define Euler angles, it is necessary to define an additional axis, line of nodes ($\mathbf{N}$). The line of nodes is defined as the intersection between $\mathbf{xy}$ and $\mathbf{XY}$ coordinate plates. With this representation, the Euler angles are defined as follows (Fig. 1):

- α (or φ) is the angle between the $\mathbf{x}$ axis and the $\mathbf{N}$ axis, and represents a rotation around the $\mathbf{z}$ axis.
- β (or θ) is the angle between the $\mathbf{z}$ axis and the $\mathbf{Z}$ axis, and represents a rotation around the $\mathbf{N}$ axis.
- γ (or ψ) is the angle between the $\mathbf{N}$ axis and the $\mathbf{X}$ axis, and represents a rotation around the $\mathbf{Z}$ axis.

The Euler angles φ, θ, and ψ, are known respectively as spin, nutation, and precession. However, the commonly used terms are roll, pitch, and yaw, respectively, which originate from the theory of aerodynamics.

The direction cosine matrix, expressed in the terms of trigonometrical functions of Euler angles, can be expressed as [Premerlani and Bizard 2009]:

$$\mathbf{R} = \begin{bmatrix} \cos\theta\cos\psi & \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi & \cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi \\ \cos\theta\sin\psi & \sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi & \cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{bmatrix} \quad (1)$$

3. DCM ALGORITHM

DCM (Direct Cosine Matrix) algorithm was developed for applications in modeling aviation technologies. In this context, the axes of an airplane are denoted as yaw (perpendicular axis), pitch (lateral axis), and roll (longitudinal axis), as shown in Fig. 2.

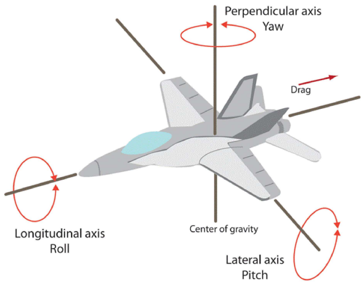

Figure 2: Orientation of axes on airplanes.

DCM algorithm uses data from accelerometer, gyroscope, and GPS receiver or magnetometer in order to determine the attitude of an object. The initial version of the algorithm was using GPS data as it was intended for usage with moving objects. However, if the object is still, then magnetometer has to be used. The working principle block diagram of the DCM algorithm is shown in Fig. 3 [Premerlani and Bizard 2009]

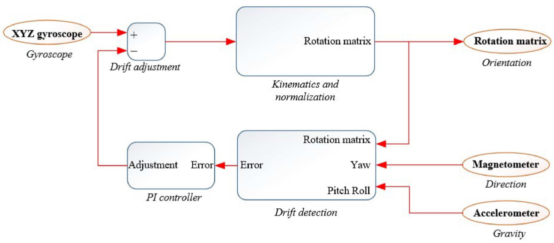

Figure 3: The block diagram of the DCM algorithm.

The accuracy of DCM algorithm mostly depends on characteristics of the gyroscope, such as sensitivity (gyroscope gain for converting rotation rate to voltage), offset (output of the gyroscope in the absence of rotation), drift (integrated effects over time of a slowly varying offset and noise), calibration (application of correct gain multipliers to the gyroscope signal), and saturation (happens when the rotation rate of the object passes the maximum range of the gyroscope).

Gyroscope is used for gathering the information about the orientation of the object. These information are produced by integrating nonlinear differential kinematic equations that show relation between the orientation change time rate and the rotation rate of the object. This process of integration eventually accumulates numerical errors, so the process of normalization has to take place. With normalization, the small adjustment to the rotation matrix are made.

In order to detect errors in DCM elements, the reference vectors have to exist. The main request for reference vectors is that the drift is not present. As reference vectors are used vectors obtained from accelerometer and magnetometer. Magnetometer is used to detect errors in yaw, and accelerometer for detection of errors in roll and pitch. Beside numerical errors, there are also errors due to gyroscope drift and offset. Based on measured errors, a PI (Proportional-Integral) controller is used for making adjustments to the gyroscope outputs. The PI controller adjusts the rotation rate of the gyroscope.

4. CHARACTERISTICS AND WORKING PRINCIPLE OF 9DOF SENSOR

The SparkFun 9DOF RAZOR IMU SEN-10736 sensor board [SparkFun 2016] was used as 9DOF sensor. It incorporates 3-axes ADXL345 accelerometer, 3-axes ITG-3200 gyroscope, and 3-axes HMC8553L magnetometer. The onboard processor is ATMega 328 @ 8MHz, and can be programmed with Arduino software suite. Its appearance is shown in Fig. 4.

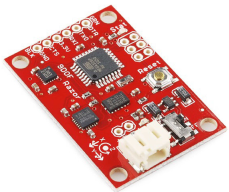

Figure 4: SparkFun 9DOF RAZOR IMU SEN-10736 sensor board.

For Bluetooth transmission of data to the computer, the SparkFun Bluetooth Mate Gold device is used, and as a GPS receiver, the GlobalTop FGPMMOPA6B device is used.

The firmware which incorporates DCM algorithm for this device can be downloaded from [GitHub 2016]. The system has the ability to send data either via USB serial connection or via Bluetooth. The output of the sensor readings are in the following format:

#YPR=yaw,pitch,roll

where yaw, pitch, and roll represent the actual values of those angles (e.g. #YPR = 10.15, -2.56, 71.25). These values are shown in degrees, and can have negative values.

Since the sensors incorporated in the 9DOF IMU (Inertial Measurement Unit) possess some irregularities due to its fabrication process, they have to be calibrated. For accelerometer calibration, the sensor plate has to move in every one of nine possible directions of the axes and the minimum and maximum values of acceleration are given. In order to calibrate the gyroscope, the sensor plate has to remains still for a period of ten seconds, and this also gives minimum and maximum readings of the average noise in the sensor. The magnetometer calibration process is somewhat more complicated and involves loading the Processing sketch for collection of magnetometer data. The sensor has to be moved in all possible directions until the unit sphere is evenly covered with readings [GitHub 2016]. This calibration output is presented in the Fig. 5.

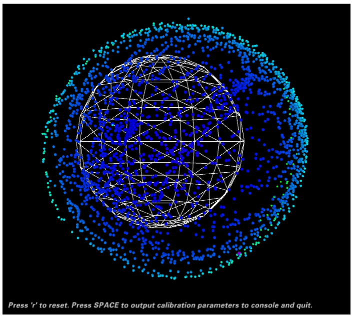

Figure 5: Appearance of the calibration sphere in mostly undisrupted environment.

The disturbances in the magnetic field that can corrupt magnetometer readings have to be annulled or at least minimized. They can be divided in hard iron and soft iron disturbances.

Hard iron disturbances may occur if the environment is occupied with objects with their own magnetic field. The magnetic field distribution on the unit sphere shows misaligned measurements with the center of the sphere. Soft iron disturbances may occur if the environment is prone to perturbations in the magnetic field, due to some iron or nickel materials present. The magnetic field distribution on the unit sphere shows an ellipse.

5. THE APPLICATION OF 9DOF SENSOR FOR DETERMINATION OF TELESCOPE POSITION

In order to show sensor orientation information (yaw, pitch, and roll) and to determine azimuth, altitude, right ascension and declination, the PC application was written in C# programming language.

5. 1. DESCRIPTION OF THE PC APPLICATION

The appearance of this application is shown in Fig. 6. As can be seen, the PC application consists of several sections: **Bluetooth veza (Bluetooth connection)**, the system searches for Bluetooth module on 9DOF sensor in order to establish connection; **Uspostavljanje veze sa senzorom (Sensor connection)**, the user chooses the serial port on which the data is received at the speed of 57,000bps; **Ojlerovi uglovi (Euler angles)**, yaw, pitch, and roll values read from the sensor; **Horizontske koordinate (Horizontal coordinates)**, azimuth (azimut) and elevation (visina) in degrees. Azimuth is calculated from yaw, and elevation from pitch; **JD, GMST, LMST, HA**, calculations of Julian day, Greenwich mean sidereal time, Local mean sidereal time and hour angle; **Ekvatorske koordinate (Equatorial coordinates)**, right ascension (rektascenzija) and declination (deklinacija), together with the stability assessments of their calculations (Stabilnost odredjivanja); **GPS**

prijemnik (GPS receiver), data gathered from GPS receiver. The user chooses the serial port on which the data are received at the speed of 9.600bps and the information about UTC time (UTC vreme), geographic longitude and latitude (geografska dužina, geografska širina), altitude (nadmorska visina), etc. are displayed; **Local time and coordinates (lokalno vreme i koordinate)**, the system doesn't have to use GPS receiver, but instead can locally save geographic longitude and latitude of the telescope positions and to gather time from the computer. It also has possibilities to delete and add new locations; **Location map info (prikaz lokacije na mapi)**, it shows the map location based on selected geographic coordinates, with Google Maps API implementation. It also possess the slider for zooming in and out the map.

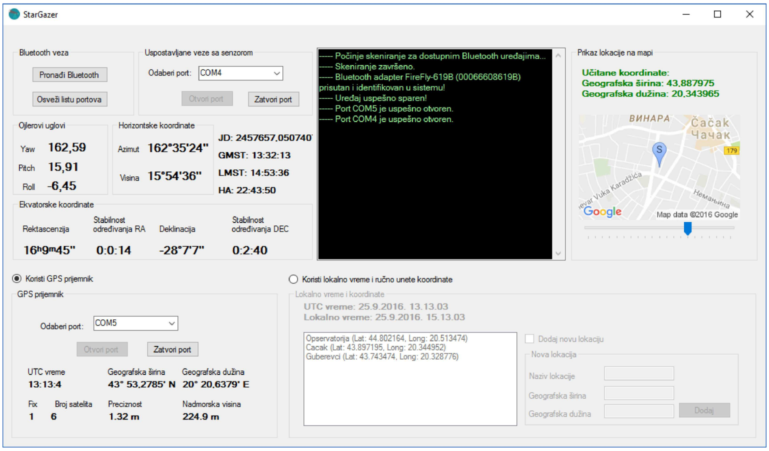

Figure 6: The appearance of the PC application.

5. 2. CALCULATION OF EQUATORIAL COORDINATES AND TIME PARAMETERS

Julian day number is calculated based on Meeus's algorithm [Meeus 1998] as follows:

Let Y be the year, M the month, and D the day. Let INT(x) be the function that gives largest integer that is less or equal to the number x.

If M > 2, then Y and M don't change. If M = 1 or M = 2, then Y = Y - 1, and M = M + 12. The following values are calculated:

$$A = INT(\frac{Y}{100}) \tag{2}$$

$$B = 2 - A + INT(\frac{A}{4}) \tag{3}$$

Julian day is then calculated as follows:

$$JD = INT(365.25(Y + 4716)) + INT(30.6001(M + 1)) + D + B - 1524.5 \tag{4}$$

If the current time is different from 12^h UT, then it is needed an additional factor (h is for hours, m for minutes, and s for seconds) that sums up with JD above:

$$add\_factor = \frac{h}{24} + \frac{m}{1440} + \frac{s}{86400} \tag{5}$$

Greenwich mean sidereal time (GMST) is calculated based on the following algorithm [Meeus 1998]:

Calculate the value of T:

$$T = \frac{JD - 2451545.0}{36525} \tag{6}$$

GMST in degrees is then:

$$\theta_0 = 280.46061837 + 360.98564736629(JD - 2451545.0) + 0.0003879 33T^2 - \frac{T^3}{38710000} \tag{7}$$

The given value is fitted into interval from 0° to 360°, and then divided with 15 to get a value in hours.

Local mean sidereal time (LMST) is calculated as follows:

$$LMST = GMST + \frac{longitude}{15} \tag{8}$$

Declination is calculated as follows:

$$\delta = \arcsin(\sin\varphi \sin h - \cos\varphi \cos h \cos A) \tag{9}$$

and hour angle as follows:

$$H = \arctan(\frac{\sin A}{\cos A \sin\varphi + \tan h \cos\varphi}) \tag{10}$$

where α is right ascension, δ declination, h elevation, A azimuth, φ latitude, and H local hour angle.

Right ascension is given as:

$$\alpha = LMST - H \tag{11}$$

5. 3. THE RESULTS AND DISCUSSION

The system is tested on the private telescope in the village of Guberevci, Lučani municipality. The tests were conducted by pointing the telescope to the seven stars (Polaris, Capella, Mirfak, Mizar, Altair, Vega, and Fomalhaut). The absolute error is calculated for readings of azimuth, elevation, right ascension and declination, and their average values are given in Table 1.

Table 1: The average values of absolute errors of horizontal coordinates measurements in the format degrees : (arc) minutes : (arc) seconds

Coordinate	Azimuth	Elevation	Right Ascension	Declination
Average absolute error	8:43:05	2:00:03	02:25:33	5:33:59

The absolute error of azimuth determination varies from 1° to almost 14°, primarily due to nonlinearity and imprecision in magnetometer measurements. The absolute error of elevation determination is somewhat constant and around 2°. If we exclude Polaris from right ascension measurements, then the absolute error doesn't supersede 1°. The absolute error of declination measurements also varies a lot, from 1° to over 9°.

The some form of stability assessment is done by creating a buffer with ten current values of declination and right ascension. The buffer data is averaged, and then the total variation around average value and standard deviation are calculated.

The possible solutions for improvement of sensor reading might be linearization of magnetometer [Sreejith et al. 2014], creation of look-up table with corresponding values of correct and measured azimuth, or usage of another algorithm for Euler angles determination, such as Madgwick filter with quaternions [Madgwick 2010], Extended Kalman filter [Caron et al. 2006] or Complementary filter [Mahony et al. 2008].

6. CONCLUSION

This paper presented the usage of 9DOF sensor in the determination of the coordinates of the point the telescope is oriented at. The DCM algorithm for sensor data fusion enables the calculation of Euler angles based on data received from 3-axes gyroscope, accelerometer, and magnetometer.

The sensor that was used in the measurements is relatively cheap and not sufficiently precise. Because of that, the errors on the sensor outputs are relatively high. In order to improve the precision of the measurements, some filtering has to be done. The other solution would be the usage of more expensive and accurate devices that come shielded from magnetic and other disturbances. Of course, these systems come at much greater price, so a balance between price and accuracy has to be made.

Acknowledgment

The work presented in this paper was funded by grant TR32043 for the period 2011-2017, by the Ministry of Education, Science, and Technological Development of the Republic of Serbia.

References

Caron, F., Duflos, E., Pomoranski, D., Vanheeghe, P.: 2006, "GPS/IMU data fusion using multisensory Kalman filtering: introduction of contextual aspects", *Information Fusion*, **7**, 2, pp. 221-230.

Diebel, J.: 2006, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors", Stanford University.

GitHub: 2016, "github/ptrbrtz/razor-9dof-ahrs", Available at: https://github.com/ptrbrtz/razor-9dof-ahrs.

Madgwick, S.: 2010, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays", Internal Report.

Mahony, R., Hamel, T., Pflimlin, J. M.: 2008, "Nonlinear Complementary Filters on the Special Orthogonal Group", *IEEE Transactions on Automatic Control*, **53**, 5, pp.1203-1218.

Meeus, J.: 1998, *Astronomical algorithms*, Willmann-Bell, Richmond, Virginia.

Premerlani, W., Bizard, P.: 2009, "Direction Cosine Matrix IMU: Theory", Internal Report.

SparkFun: 2016, "9 Degrees of Freedom Razor IMU", Available at: https://www.sparkfun.com/products/10736.

Sreejith, A. G. et al.: 2014, "A Raspberry Pi Based Attitude Sensor", *Journal of Astronomical Instrumentations*, **3**, 2, pp.1440006 1-10.