Toolbox for GPS-based attitude determination: An implementation aspect

Zhen Dai, Stefan Knedlik and Otmar Loffeld

Center for Sensor Systems, University of Siegen
Germany

1. Introduction

The global positioning system (GPS) is known as its positioning and timing applications. Besides that, GPS multi-antenna system has become a high-accurate approach for attitude determination (Cohen et al. 1994; Van Grass and Braasch 1991). In comparison with the traditional inertial sensors, the GPS multi-antenna system provides attitude results without drift effects, and it has the advantages due to the cost-effectiveness and the flexible installation. The drawback is that GPS signals can be interfered or blocked in some shadow environments. Also, resolving the carrier phase ambiguity is another challenging task for single-frequency receiver and real-time applications.

This MATLAB toolbox is developed for attitude determination using GPS code data, which is partly introduced in (Dai et al. 2008). However, the implementation aspects will be highlighted herein. Some development details will be stressed and some problems occurred during the development procedure will also be presented in the following parts. It derives the attitude parameters based on the double-differenced (carrier phase smoothed) C/A code measurement. After the baselines between the antennas have been calculated, two algorithms can be invoked to determine the attitude: a direct computation method and a least-squares estimation approach (Lu 1995). The motivation of developing such a toolbox is not to provide a powerful and reliable software package for real applications, but to present the basic functions needed for GPS-based attitude determination.

It should be pointed out that this toolbox contains no functions to resolve double-differenced integer phase ambiguities. However, there is already some source code available (Chang and Zhou 2007; Jonge de and Tiberius 1996), so that those who want to employ the phase measurement to achieve a precise attitude result can choose the proper source code and combine it with our programs. As an example, we provide a demo program with a set of resolved ambiguities to show the results based on carrier phase.

In the following parts, we will first introduce the theoretical background of the attitude determination using GPS multiple antenna system, and then the implementation aspects will be highlighted. The test results will be presented to show the performance of the toolbox. Finally this study will be summarized and the future work will be suggested.
2. Coordinate frames and general mathematic model

In this section, the coordinate frames involved in the attitude determination system will be first explained. Then the attitude parameterization based on the Euler angles will be introduced. The general mathematic model for attitude determination will be presented.

2.1 Coordinate frames

In order to clarify the attitude determination using GPS multiple antennas, several coordinate frames needs to be distinguished, including the Earth-Centered-Earth-Fixed (ECEF) frame, the Local Level Frame (LLF), the Antenna Body Frame (ABF).

The ECEF, also referred to as terrestrial equatorial system, is defined as follows: the origin is the geocenter; \( X_{ECEF} \) is located in the equatorial plane and points towards then Greenwich meridian; \( Z_{ECEF} \) is the rotation axis of the earth; \( Y_{ECEF} \) completes the right-handed Cartesian system along with \( Z_{ECEF} \) and \( X_{ECEF} \).

The local level frame (LLF) describes the local coordinates of a point with respect to a reference point, and it is usually expressed in East-North-Up directions. The origin of LLF is chosen as the reference point. \( X_{LLF} \) points to ellipsoidal east and \( Y_{LLF} \) to north; \( Z_{LLF} \) is along with the ellipsoidal norm and points upwards. LLF is usually adopted as the reference frame in the attitude determination frame.

The antenna body application (ABF) is formed by the GPS antennas. Here we assume that the antennas are mounted on a rigid platform, i.e. the relative distances between the antennas remain unchanged. One antenna is chosen as the master antenna, and the other antennas are called slave antennas. Actually, three antennas are sufficient to determine the antenna body frame. The origin is chosen as the position of the phase center of antenna 1, namely the master antenna. \( Y_{ABF} \) is assumed along with the baseline from antenna 1 to antenna 2. \( X_{ABF} \) is perpendicular with \( Y_{ABF} \) and lies in the plane defined by antenna 1, 2 and 3. \( Z_{ABF} \) is perpendicular to both of the \( X_{ABF} \) and \( Y_{ABF} \) axis and points upwards.
2.2 Euler angles
A three-dimensional rotation can be decomposed into three individual rotations with each around a single axis. Euler angles represent the rotation angles with respect to three axes and usually comprise of yaw, pitch and roll angles, as shown in Fig. 2. Note that in the right-handed frame, the Euler angles describe counter-clockwise rotations when viewed from the end of the positive axes and clockwise rotation when viewed from the origin of the positive axes.

![Fig. 2. Euler angles](image)

LLF = Local Level Frame

Each rotation can be described by a Direction Cosine Matrix (DCM). A three-dimensional rotation can be obtained by multiplying the three DCMs into a specific order, yielding the combined rotation matrix. An example using the yaw-pitch-roll sequence is given below

\[
x_{ABF} = \begin{bmatrix}
  c(r)c(y) - s(r)s(p)s(y) & c(r)s(y) + s(r)s(p)c(y) & -s(r)c(p) \\
  -c(p)s(y) & c(p)c(y) & s(p) \\
  s(r)c(y) + c(r)s(p)s(y) & s(r)s(y) - c(r)s(p)c(y) & c(r)c(p)
\end{bmatrix} \begin{bmatrix}
  x_{LLF}
\end{bmatrix} \tag{1}
\]

where y, p and r are short-hand notations for yaw, pitch and roll angles, respectively; c and s denote the cosine and sine operators, respectively.

3. Algorithms used for attitude determination in the toolbox
In section 2, the general mathematic model for attitude determination is presented. However, how to resolve the attitude parameters using this model is still an issue. In this section, we will introduce two algorithms, the direct attitude computation and least-squares attitude determination.

3.1 Least-squares attitude determination
In order to resolve the three dimensional unknown Euler angles using the nonlinear model given in (1), more than three equations are needed, or in other words, more than three baseline vectors are needed. Each master-slave antenna baseline provides three baseline vectors, and hence we need a minimum of two non-collinear slave antennas. Based on the
linearization of the DCM around the proper attitude parameters \( y_0, r_0 \) and \( p_0 \), we have the following model to construct the least-squares attitude estimation (Lu 1995):

\[
\begin{bmatrix}
A_2 \\
A_3 \\
\vdots \\
A_n \\
\end{bmatrix}
\begin{bmatrix}
\Delta y \\
\Delta p \\
\Delta r \\
\end{bmatrix}
= \begin{bmatrix}
\mathbf{R}_0 - \mathbf{I} \\
\mathbf{O} \\
\vdots \\
\mathbf{O} \\
\mathbf{O} \\
\end{bmatrix}
\begin{bmatrix}
\mathbf{O} \\
\mathbf{O} \\
\vdots \\
\mathbf{O} \\
\mathbf{O} \\
\end{bmatrix}
\begin{bmatrix}
l_2 \\
l_3 \\
\vdots \\
l_n \\
\end{bmatrix}
\begin{bmatrix}
\Delta l_2 \\
\Delta l_3 \\
\vdots \\
\Delta l_n \\
\end{bmatrix}
= 0
\] (2)

In order to describe the matrix \( A_i \) for \( i = 2, 3, \ldots, n \), we express the combined rotation matrix \( \mathbf{R} \) in terms of row vectors, i.e. \( \mathbf{R} = [\mathbf{r}_1 \, \mathbf{r}_2 \, \mathbf{r}_3]^T \), then the matrix \( A_i \) has the form:

\[
A_i = \begin{bmatrix}
\frac{\partial (r_{1,i})}{\partial y} & \frac{\partial (r_{1,i})}{\partial p} & \frac{\partial (r_{1,i})}{\partial r} \\
\frac{\partial (r_{2,i})}{\partial y} & \frac{\partial (r_{2,i})}{\partial p} & \frac{\partial (r_{2,i})}{\partial r} \\
\frac{\partial (r_{3,i})}{\partial y} & \frac{\partial (r_{3,i})}{\partial p} & \frac{\partial (r_{3,i})}{\partial r} \\
\end{bmatrix}
\] (3)

In model (2), \( \mathbf{R}_0 \) is the DCM at \( y_0, r_0 \) and \( p_0 \); \( \Delta b_i \) and \( \Delta l_i \) are the errors contained in the antenna body frame and the local level frame of the antenna \( i \), respectively; \( \mathbf{I} \) denotes the identity matrix and \( \mathbf{O} \) the zero matrix. A detailed \( A_i \) can be seen below:

\[
A_{i,1,1} = \frac{\partial (r_{1,i})}{\partial y} = \begin{bmatrix}
-c(r)x_{i,j} - s(r)s(p)y_{i,j} & s(y) + [s(r)s(p)x_{i,j} + c(r)y_{i,j}]c(y)
\end{bmatrix}
\]

\[
A_{i,1,2} = \frac{\partial (r_{1,i})}{\partial p} = \begin{bmatrix}
s(r)y_{i,j} + [s(r)x_{i,j} + c(r)y_{i,j}]c(y)
\end{bmatrix}
\]

\[
A_{i,1,3} = \frac{\partial (r_{1,i})}{\partial r} = \begin{bmatrix}
s(r)y_{i,j} + [s(r)x_{i,j} + c(r)y_{i,j}]c(y)
\end{bmatrix}
\]

\[
A_{i,2,1} = \frac{\partial (r_{1,i})}{\partial y} = \begin{bmatrix}
s(y)x_{i,j} - c(y)y_{i,j}
\end{bmatrix}c(y)
\]

\[
A_{i,2,2} = \frac{\partial (r_{1,i})}{\partial p} = \begin{bmatrix}
s(y)x_{i,j} - c(y)y_{i,j}
\end{bmatrix}c(y)
\]

\[
A_{i,2,3} = \frac{\partial (r_{1,i})}{\partial r} = 0
\]

\[
A_{i,3,1} = \frac{\partial (r_{1,i})}{\partial y} = \begin{bmatrix}
s(r)x_{i,j} + c(r)s(p)y_{i,j}
\end{bmatrix}c(y)
\]

\[
A_{i,3,2} = \frac{\partial (r_{1,i})}{\partial p} = \begin{bmatrix}
s(r)x_{i,j} - c(r)c(y)y_{i,j}
\end{bmatrix}c(p)
\]

\[
A_{i,3,3} = \frac{\partial (r_{1,i})}{\partial r} = \begin{bmatrix}
s(r)x_{i,j} - c(r)c(y)y_{i,j}
\end{bmatrix}c(r)
\]
where the subscript \( l \) indicate the local level frame. Based on this model, the least-squares adjustment can be carried out. The correction values for the three Euler angles corresponding to a rotation matrix \( R_0 \) are computed by:

\[
[\Delta y \quad \Delta p \quad \Delta r]^T = \left[ \sum_{i=1}^{n} A_i^T \left( R_i^T \text{Cov}(1, R_i + \text{Cov}(b_i))^{-1} A_i \right) \right]^{-1} \\
\times \left[ \sum_{i=1}^{n} A_i^T \left( R_i^T \text{Cov}(1, R_i + \text{Cov}(b_i))^{-1} (\Delta l_i - \Delta b_i) \right) \right]
\]

(5)

where the short-hand notation \( \text{Cov}(\cdot) \) denotes the error covariance matrix. The least-squares adjustment proceeds until the correction values converge to a certain threshold or the maximal iteration number is reached.

3.2 Direct attitude computation approach

Another fast algorithm for attitude determination is referred to as direct attitude computation approach. Based on the definition of antenna body frame, the coordinates of the slave antenna in the antenna body frame can be expressed as \( b_2=[0 \ b_{12} \ 0]^T \), where \( b_{12} \) is the magnitude of baseline from the master antenna (antenna 1) to the slave antenna (antenna 2), as shown in Fig. 1. Substituting the antenna body frame coordinate of the slave antenna into model (1) and using the orthogonality of the rotation matrix yields the local level frame coordinate of slave antenna:

\[
\begin{bmatrix}
  x_{2,l} \\
  y_{2,l} \\
  z_{2,l}
\end{bmatrix} = b_{12} \begin{bmatrix}
  -c_p\delta_y \\
  c_p c_y \\
  \delta_p
\end{bmatrix}
\]

(6)

After that, the yaw angle and pitch angle can be directly obtained as:

\[
yaw = -\tan^{-1}\left( \frac{x_{2,l}}{y_{2,l}} \right)
\]

\[
pitch = \sin^{-1}\left( \frac{z_{2,l}}{b_{12}} \right) = \tan^{-1}\left( \frac{z_{2,l}}{\sqrt{x_{2,l}^2 + y_{2,l}^2}} \right)
\]

(7)

From both of the expressions of pitch given in (7) it can be seen that the pitch angle is acquirable using only the local level frame coordinate of the slave antenna instead of using the baseline length \( b_{12} \). This reveals a significant advantage of the direct attitude computation that the magnitude of baseline length does not need to be measured in advance.
The coordinate of antenna 3 in the local level frame is then required to fix the value of roll. We can first rotate the antenna 3 by yaw and pitch resulted from (7) in order to obtain the rotation matrix including roll:

\[
\begin{bmatrix}
    x'_{3,l} \\
    y'_{3,l} \\
    z'_{3,l}
\end{bmatrix} =
\begin{bmatrix}
    1 & 0 & 0 \\
    0 & c(p) & s(p) \\
    0 & -s(p) & c(p)
\end{bmatrix}
\begin{bmatrix}
    c(y) & s(y) & 0 \\
    -s(y) & c(y) & 0 \\
    0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_{3,l} \\
    y_{3,l} \\
    z_{3,l}
\end{bmatrix}
\]

(8)

Then following relationship holds by employing the coordinate of antenna 3 into the antenna body frame:

\[
\begin{bmatrix}
    x_{3,b} \\
    y_{3,b} \\
    0
\end{bmatrix} =
\begin{bmatrix}
    \cos(r) & 0 & -\sin(r) \\
    0 & 1 & 0 \\
    \sin(r) & 0 & \cos(r)
\end{bmatrix}
\begin{bmatrix}
    x'_{3,l} \\
    y'_{3,l} \\
    z'_{3,l}
\end{bmatrix}
\]

(9)

Note that the first two scalars \(x_{3,b}\) and \(y_{3,b}\) in the body frame vector are not to be explicitly specified, from the scalar 0 in body frame of the antenna 3 we can simply derive the roll angle:

\[
r = -\tan^{-1}\left(\frac{z'_{3,l}}{x'_{3,l}}\right)
\]

(10)

The direct attitude computation and least-squares approaches apply to different scenarios. If the baselines are not given in advance, the direct attitude computation can still yield the attitude results. However, it does not take all the measurement into calculation and hence leads to a non-optimal solution.

4. Implementation aspects

The flowchart of the toolbox is given in Fig. 3. Some key points at each step will be highlighted in each subsection.
4.1 Graphic user interface

After copying all the files into a directory and loading the main program `ControlPanel()` into the MATLAB environment, the toolbox can be executed. Shown first is the GUI for setting the application scenarios and assigning the relevant parameters.
The layout of the GUI is composed of three columns. In the left column, the user can choose the RINEX observation files (upper) and navigation file (lower). Note that only the navigation file of an antenna is needed because all the antennas were receiving the satellite information simultaneously. In the upper parts of the middle column, the user can input the baseline of the antennas in units of meters. In the lower parts, three parameters can be assigned, including a) the smoothing interval which determines the length of epochs for code data smoothing, b) the processed epochs which allows the user to process the data at a certain length in a large data set, and c) the elevation mask angle which excludes some low angle satellites due to their potential large troposphere error and multipath error. Note that the parameters in the middle column are only optional items. The user can leave them blank, and in this case, only the direct attitude computation will be carried out and the default values of the parameters will be used instead. In the right column there are two buttons for viewing the readme file and starting the calculation.

4.2 Processing of RINEX data
Receiver Independent Exchange Format (RINEX) is a standard format widely accepted by the receiver manufactures to record the GPS navigation and measurement messages. However, there are slight differences in the files outputted from different receivers. A robust program should take these differences into account in order to expand the application scenarios. The main program for reading and analyzing the RINEX data can be seen from the function ProcessRINEX(). This function will invoke two functions LoadRinexNav() and LoadRinexOBS() to read the navigation file and the observation file, respectively.

This toolbox is dedicated to the post-processing, so that the data will be analyzed and saved into MATLAB data files (.MAT file). The RINEX navigation file contains the ephemeris data...
and is mainly used to calculate the information related to the satellites, like the satellite coordinates. The RINEX observation file is mainly composed of the available GPS measurement of each time epoch.

As the first task for analyzing the RINEX observation data, the head part should be read. The head part includes some global information, like the available measurement types supported by this receiver and their corresponding orders recorded in the following paragraphs. Only the C/A code data and carrier phase data on L1 signal are used in this toolbox. The carrier phase data will not be directly used for positioning, but can smooth the code data to reduce the potential multipath error and improve the quality of code data. Another important factor is the starting and end time of the data records, as well as the sampling interval. Based on these information, the total epochs involved can be calculated. In this toolbox the measurement of each antenna is recorded in a matrix with the first order being the time index and second order being the measurement type. After knowing the total number of epochs and measurement types, the matrix can be predefined. However, the starting and end epoch are not always available for some receivers. In this case, the matrix has to be extended dynamically by epoch, and this can significantly reduce the processing efficiency for a large data set.

Each paragraph of the RINEX observation file represents the measurement of each epoch. The first line of the paragraph indicates the visible satellites. Due to the all-in-view function and the integration of GPS with other GNSS system like GLONASS, a single line might not be enough to identify all the satellites in view, so that an additional line(s) follows, like the following example from a GPS/GLONASS receiver:

```
08 11 20 14 56 36.0000000  0 16G19G 8G 3G22G18G 6G15G 7G16G21R 7R14
R22R13R21R 6
```

This case must be taken into account in order to prevent the ignorance of the satellites. In this toolbox, only GPS data will be used and the other GNSS data are excluded.

When processing the GPS ephemeris data, more than one ephemeris data for a satellite might appear in a single navigation file. In this case, the ephemeris data close to the current epoch will be utilized. This is realized in the function `SeekEpheEpoch()`.

Each line of a standard RINEX file should be composed of exactly 80 characters. However, in some lines without meaningful words at the end, the line terminator always appears before the 80th character. This always causes the problem when analyzing the RINEX data. We use a function `getline80()` to solve this problem. This function first invokes the function `fgets()` to read each line and add space characters to the end if the length of the line is less than 80 characters.

### 4.3 Data synchronization and verification

The data synchronization is to collect the GPS measurement of the common satellites acquired at the same time epoch. In most receivers, the time can be accurate to the second level or deci-second level. Nevertheless, some low cost receivers output the time with a slight floating part. For example the data at the same epoch from two different receivers with 1 Hz sampling rate:

```
08 11 20 14 16 35.0000000  0 11G03G06G07G15G16G18G19G21G22G25G26
08 11 20 14 16 35.0040000  0 07G03G07G15G18G19G21G25
```
The slight floating part can be seen from the second receiver \((35.0040000 \text{ s})\). Both data cannot be synchronized due to the slight time difference. For this reason, it is recommended to use the receivers of the same model and the same manufacture. The synchronization manifests itself also in the search for the common satellites. The common satellites herein mean the satellites tracked simultaneously by all the receivers. According to the different tracking abilities, different receivers might not track the same number of satellites, especially at the first several epochs. However, only the common satellites will be employed.

The data verification is mainly to check if the measurement of a receiver is continuously available. If the receiver is working properly, the measurement from more than 4 satellites should be always acquirable in an open-air environment. The sampling rate is needed to check the data interruption. If the sampling interval is identified in the RINEX file, it will be directly used; otherwise the program will calculate this quantity from the first several epochs. Once the data of the current epoch is available, the data of the next epoch with the known time interval should also be presented, otherwise the data interruption can be concluded. In this case, the program should warn the user regarding this issue and the data will be synchronized again from the next common epochs. The data synchronization and verification are implemented in the function \(\text{ProcessRINEX()}\).

### 4.4 Single point positioning for master antenna

The position of the master antenna serves as the reference point for the coordinate transformation from the ECEF to the local level frame(Hofmann-Wellenhof et al. 2001). The master antenna can be positioned by the single point positioning. Although the single point positioning yields only an accuracy of several meters to several tens of meters, it will bring just millimeter error to the transformed coordinate of the slave antennas(Lu 1995).

As the first step, the GPS code data will be smoothed by the carrier phase data in order to reduce the potential multipath error and thermal noise(Hatch 1982), and this is done by the function \(\text{Smoothing()}\). The length of data sequence for smoothing can be identified in advance in the GUI. A zero-valued length means that the code will be directly used for positioning without smoothing. For simplicity, the ionosphere correction and troposphere correction are not implemented in the algorithm for single point positioning.

Another operation of the single point positioning is to obtain the satellite coordinates at each epoch. These coordinates will be saved and further used for the differential positioning applied later because these satellites can be tracked by all the receivers. A key satellite with the highest elevation angle will be identified and used for the differential positioning. However, if the carrier phase data will be used, this criterion for key satellite selection can be changed to choosing the satellite having the longest observation time. This is done in the function \(\text{SinglePointGPS()}\).

### 4.5 Differential positioning for each master-slave antenna baseline estimation

The baseline between the master and slave antennas needs to be determined precisely. Here we apply the double-differential positioning to determine the baseline. The resulted baseline should be at the decimeter level for high-level receiver and smoothed code data. Related information can be seen in the function \(\text{DGPS_code()}\).
4.6 Selection of the algorithms for attitude determination

In this toolbox, the user can fix the baseline length between antennas once this has been measured in advance using other sensors or approaches. A simple way to determine the baseline is to collect the GPS data for a long observation session and perform the differential GPS technique using carrier phase data. There are also some free internet services for this purpose, like AUSPOS and CSRS-PPP services. However, inputting the baseline is not a prerequisite for resolving the attitude from GPS data. As stated before, the direct attitude computation approach does not need the premeasured baseline. If the baselines are not identified, the direct attitude computation will be the only approach. One drawback of this approach is that the measurements are not fully utilized so that it yields non-optimal estimates. Once the baselines are given by the user, the least-squares attitude determination will also be performed. The user can input the GPS data from 3 or more redundant receivers. The redundant data can only be employed once the corresponding baselines are provided.

4.7 Attitude determination

The baseline between each master-slave antenna pair is obtained initially in ECEF frame and will be project into the local level frame using the function ECEF2ENU(). After that, the aforementioned approaches can be carried out to determine the attitude. Both can be seen in the function AD_Direct() and AD_LSQ(), respectively. The least-squares adjustment needs an initial guess of the Euler angles. The initial values can be chosen freely using the value like zeros. The attitude results obtained from the direct approach can also be used to initialize the least-squares adjustment in order to reduce the iteration number. The range of the Euler angles should be fixed, and in this toolbox the Euler angles can range from −180 to 180 degrees.

Before applying the least-squares approach, the antenna body frame needs to be fixed from the given baselines, and this can be seen from the function GetBodyCoordinates(). If we have three antennas, the antenna body frame can be simply calculated according to the definition. Once more than three antennas are available, the coordinate of the redundant antennas in the antenna body frames can be fixed based on the least-squares principle. As the coordinates of the first three antennas have been obtained, and the distance between the redundant antenna to the first three antennas are known, the coordinate of the redundant antenna should result in the minimal least-squares residuals. However, this is a multiple-solution problem, as we cannot explicitly calculate the sign of the Z value of the coordinate of the slave antenna, namely we do not know whether the redundant antenna is above or under the X-Y plane. However, taking the redundant antennas into calculation should yield a compatible result with the results from other algorithms. For this reason, the attitude parameters can be first resolved by applying the direct attitude computation or the least-squares approach involving only the first three antennas. The resulted attitude parameters are then used to identify the sign of the Z-value of the redundant antenna in the antenna body frame.
4.8 A demo example using carrier phase data
The only difference between the attitude determination using code data and carrier phase data lies in the integer ambiguity resolution. Although the ambiguity resolution algorithm is not included in the toolbox, we still provide a set of resolved integer ambiguities to demonstrate the attitude results using carrier phase data. The ambiguity set is recorded into the file `ambiguity.mat` and only corresponds to the test data embedded in the toolbox. Note that the ambiguity is related to a common key satellite during the entire observation session. The differential positioning based on carrier phase is implemented in the function `DGPS_phase()`, where the ambiguity appears in the variable statements of the function.

5. Results
In the toolbox, a set of RINEX files obtained from a static experiment is provided to demonstrate the performance of the toolbox. The GPS measurements are acquired by using a NovAtel® DL-4 receiver and IFEN NavX® RF GPS simulator. The GPS simulator will generate the GPS RF signals according to the antenna position in ECEF frame specified by the user. The signals are then transferred to the GPS receiver and the measurements will be outputted in RINEX format. By setting four antenna reference points we can obtain an antenna frame composed of four distributed antennas, and we also know the true baselines. The observation session takes about 10 minutes with 1 Hz data rate. The reference coordinates of four antennas in ECEF and the true values of yaw, roll and pitch are given in Table 1.

<table>
<thead>
<tr>
<th>Ant.</th>
<th>X (m)</th>
<th>Y (m)</th>
<th>Z (m)</th>
<th>Attitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3991096.821</td>
<td>563014.827</td>
<td>4927065.332</td>
<td>Yaw=51.6656°</td>
</tr>
<tr>
<td>2</td>
<td>3991081.107</td>
<td>562998.402</td>
<td>4927061.001</td>
<td>Roll=26.1822°</td>
</tr>
<tr>
<td>3</td>
<td>3991081.400</td>
<td>563019.756</td>
<td>4927065.029</td>
<td>Pitch=-39.1834°</td>
</tr>
<tr>
<td>4</td>
<td>3991093.445</td>
<td>563007.247</td>
<td>4927059.915</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. True antenna coordinates and attitude parameters

Depicted below are two sets of results, one is based on the C/A coda data and the other one is based on the carrier phase data with resolved phase ambiguities. Both results are obtained using least-squares attitude determination approach.
In the plotted attitude results, the X-axis shows the epochs and the Y-axis shows the estimated Euler angles in units of degrees. The title for each subplot also shows the mean value and the standard deviation of the results. It can be seen that the attitude results from the carrier phase data are more precise than that from code data.

Fig. 5. Result based on C/A code (upper) and phase data (lower)
Besides the attitude results, the magnitude of the errors contained in the estimated baselines are also outputted. The baseline is a fixed value for the rigid platform and hence can be used to evaluate the positioning error from GPS. The function is only activated if the baseline lengths are already given in the toolbox. Examples for code data and carrier phase data are given in Fig. 6.

![Fig. 6. Baseline length estimated by code data (upper) and carrier phase data (lower)](image-url)
The accuracy of the estimated magnitude of the baseline error from carrier phase data is at centimeter level, whereas the results from code data are at decimeter level.

Following the user manual the user can also process other data set by using the toolbox. The result will be saved into a data file results.txt for further analysis, as shown below:

......
At Epoch 2006.10.29 01:44:30.00 -> YAW=50.875  ROLL=24.440  PITCH=-38.319
At Epoch 2006.10.29 01:44:31.00 -> YAW=50.886  ROLL=24.484  PITCH=-38.332
......

6. Conclusion

This MATLAB toolbox presents some basic functions needed for the attitude determination of a rigid non-dedicated GPS multiple antenna system. Since RINEX observation and navigation files are required, this toolbox can only be used for post-processing. This toolbox is oriented to the (smoothed) GPS C/A code. If the baseline between antennas are not measured in advance, the direct attitude computation approach offers a rapid solution. If the baselines are identified, the least-squares attitude estimation provides an optimal solution based on the pre-defined antenna body frame.

As is already mentioned in (Dai et al. 2008), dual-frequency data processing and ambiguity resolution technique should be added into the toolbox. These functions are currently under the development in order to fully extend this toolbox to the processing of carrier phase data. Besides that, a robust cycle-slip detection algorithm for carrier phase data per satellite should be implemented and the attitude results based on float ambiguities should also be provided. Also, the source code given in this toolbox can be further refined by using more efficient mathematical or MATLAB internal functions.

The toolbox can be accessed via the website of ZESS (http://www.zess.uni-siegen.de/cms/upload/navigroup/AttDet_16_3_2009.zip). Any suggestions, corrections, or comments about this toolbox are sincerely welcomed and can be emailed to dai@zess.uni-siegen.de. Other contact information can be read from the user manual in the toolbox.

7. Acknowledgment

This work was funded in part by the German Research Foundation (DFG) under grant number KN 876/1-1 and KN 876/1-2, which is gratefully acknowledged.

8. References


Hatch, R. "The synergism of GPS code and carrier measurements." Presented at International Geodetic Symposium on Satellite Doppler Positioning, Las Cruces, NM.


