Course notes, module 2

Global Navigation Satellite System

Kjeld Jensen kjen@mmmi.sdu.dk

1 Agenda

1. Presentation of module 1 exercise results
2. Practical information
3. Introduction to the module theory and exercises.
4. Exercises.

2 Theory presented in class

2.1 Global Navigation Satellite Systems

1. Introduction to GNSS.
2. Introductional theory of positioning using satellites.
3. Error sources.
4. Compensation for errors using differential measurements.
5. Phase vs. code measurements.

2.2 Coordinate systems and datums

1. Geographic coordinates
2. Representations using decimal degrees, minutes and seconds.
3. Decimal accuracy
4. Transverse Mercator projection
5. Universal Transverse Mercator (UTM) projection
6. Datums

3 Global Navigation Satellite Systems

The Global Positioning System (GPS) `http://www.gps.gov` is the best known implementation of a Global Navigation Satellite System (GNSS). Please name the other GNSS operational today. Please briefly describe relevant differences.

3.1 GPS architecture

Please describe using three paragraphs the GPS *Space segment*, *Control segment* and *User segment*

3.2 GNSS error sources

Please describe using short paragraphs the most dominant error sources of a GNSS. What are the expected contributions to the final inaccuracy in meter? Please include proper references.

3.3 Dilution of Precision (DOP)

What are GDOP, PDOP, HDOP and VDOP?

3.4 Real Time Kinematics GNSS

Please explain briefly the principles of Real Time Kinematics (RTK) GNSS.

What does *integer ambiguity* mean?

3.5 GNSS accuracy

Please describe using short paragraphs the following types of positioning and the expected accuracy hereof (remember to include references):

1. Standard Positoning Service (SPS)
2. Differential GPS (DGPS)
3. RTK float
4. RTK fixed

4 Coordinate systems

4.1 Universal Transversal Mercator (UTM) accuracy

Using Python and the example scripts available in the course materials please calculate the UTM positioning error obtained over a distance of 1 km.

One way to approach this is to define a reference position represented by latitude/longitude coordinates, convert this position to UTM coordinates, then project another reference position with a known distance (according to UTM) of 1 km, convert that position back to latitude/longitude cooordinates, then calculate the distance between these same positions based on a Great Circle Distance formula (see Aviation Formulary document). This is partly implemented in the test script.

Please use the coordinates N55.47° E010.33° as one of the reference points to ease comparison of results with other teams. Why is this relevant?

Is there a difference if the distance is along the same latitude vs. along the same longitude?

Why did I (intentionally) put a 0 before the 10.33 in the listed coordinate set in the paragraph above?

4.2 National Marine Electronics Association (NMEA) 0183 data

In the course materials you will find example NMEA data which origins from current research projects. The file nmea_trimble_gnss_eduquad_flight.txt contains a log from a RTK-GNSS installed on a quadrotor drone during flight. The file nmea_ublox_neo_24h_static.txt contains 24 hours of NMEA data logged while the GNSS was at a fixed position.

An example script for importing comma separated data is provided. Please build upon this script to parse the data and visualize:

1. Altitude above Mean Sea Level with respect to time during the drone flight.
2. Number of satellites tracked with respect to time during the drone flight.
3. Map showing the drone track during the drone flight.
4. Plot showing static GNSS accuracy over 24 hours.
5. Signal to Noise Ratio (SNR) for the satellites in view with respect to time over 24 hours. (optional)
6. A hemisphere map of the location of the satellites simulated over time over 24 hours (optional)

The Python script is formatted as a class. A test function using the class is provided in the main section. It is highly recommended that you adopt this formatting for use when writing Python scripts in this course (and beyond) as this enable reuse of your script without any modifications.

Another Python script is provided for exporting positions to the KML format. You may use this to visualize tracks using e.g. Google Maps and similar mapping applications.

You will use the results of this exercise in a subsequent module on UAV positioning and navigation.

4.3 Local Tangent Plane (LTP) accuracy (optional)

Using Python and the documentation in the course materials please calculate the positioning error obtained over a distance of 1 km compared to a calculation based on latitude/longitude (Great Circle Distance).