NovaTel Differential GPS Tutorial

Author: Noah Kuntz (2009)
Contact: nk752@drexel.edu

Keywords: NovaTel, differential, GPS, RTCA, navigation, map

This tutorial assumes the reader:
(1) Has a basic knowledge of Visual C++
(2) Has some familiarity with GPS concepts

The rest of the tutorial is presented as follows:

Drexel Mini Grand Challenge Vehicle with NovaTel GPS

Step 1: Introduction

A constant challenge in robotics is determining the robot's location. This is at the heart of navigation, before obstacles can even be considered. The obvious sensor for determining position is the Global Positioning System. However getting accurate enough data from GPS is a struggle. The NovaTel GPS systems with differential GPS aim to improve the accuracy of GPS data by using two GPS reciever systems in tandem. A unmoving base station at a known position compares satelite time stamp data with that recieved by a moving rover station, and is able to calculate a more accurate position than would be possible with just one GPS. This tutorial doesn't delve into the theory behind this, but covers the basics of using a pair of NovaTel GPS units to utilize differential GPS. NovaTel ProPack G2 sensors are used, but the procedure is similiar among the various models.

Step 2: System Wiring

The first step in setting up the system is wiring the GPS and wireless modems. The NovaTel has four connections we are concerned with: the antenna port, which should of course be attached to the antenna. COM1, which should be used with the included USB to serial cable to interface your computer with the NovaTel. COM2, which should be connected to a serial modem or other way of connecting the base station and rover station. And lastly the power connector, which by default goes to a 12V car adapter plug. You can get a jack to match that connector, or cut the cable, or use alligator clips, but somehow get 12V to the NovaTel. The tip contact is +12V and the side contacts are GND. The 900mhz serial modem I used to interface the two NovaTels also required 12V, so I made a wiring harness for each station that connected the NovaTel and the modem to a connector for a 12V 2000 mAh LiPo battery, enough power for about an hours worth of testing.

Step 3: GPS Configuration

Once the GPS is wired up and powered on, each unit needs to be configured. Connect to the GPS via its COM1 port, using HyperTerminal or a similiar program. Use the settings of 38400 baud, 8 bits, no parity, 1 bit, and no flow control or handshaking. Enter these commands into the prompt for the base station NovaTel:

Base Station Config:

```
com com2 9600 n 8 1 n
saveconfig

log com2 rtcaobs ontime 2
log com2 rtcaref ontime 10
log com2 rtcadata1 ontime 5
interfacemode com2 none rtca
saveconfig

fix position LAT LON ALT
saveconfig
```

LAT, LON, and ALT are the latitude, longitude, and altitude of the base station location, the more accurately determined these are the better the accuracy of the differential GPS system will be.

Rover Station Config:

```
com com2 9600 n 8 1 n
interfacemode com2 rtca none
saveconfig
```

Note you don't have to have a computer connected to the base station after configuring it. Now to see the current position of the rover, type into its prompt "log bestposa" and hit enter. The Lat/Long/Alt will be amoung the data.

Step 4: C++ Datalogging

Lastly I will include a basic program for logging the Latitude and Longitude from the NovaTel GPS. You may need to change the 3 characters used to identify the start of the Lat/Long data as that is part of a status message. There are more sophisticated ways of parsing the data but this will suffice for a basic datalogger.

```cpp
int GPSport = 9;
unsigned char *GPSBuffer = new unsigned char[400];
int i;

// Time to run
int seconds = 250;

CSerial GPS_Serial;
if(GPS_Serial.Open( GPSport, 57600)){

        int count = 0;

        for(int j = 0; j < seconds; j++){
                for(i=0; i < 400; i++){
                        GPSBuffer[i] = 0;
                }

                // Get GPS Data
                static char command[14];
                command[0] = '\r';
                command[1] = 'L';
                command[2] = 'O';
                command[3] = 'G';
                command[4] = ' ';
                command[5] = 'B';
                command[6] = 'E';
                command[7] = 'S';
                command[8] = 'T';
                command[9] = 'P';
                command[10] = 'O';
                command[11] = 'S';
                command[12] = 'A';
                command[13] = '\r';

                unsigned char lat_gps[15], long_gps[15];
                int lat_size = 0;
                int long_size = 0;
                int data_waiting = 0;

                GPS_Serial.SendData( command, 14 );
                while(GPS_Serial.ReadDataWaiting() < 219);
                data_waiting = GPS_Serial.ReadDataWaiting();
                GPS_Serial.ReadData( GPSBuffer, data_waiting);

                printf("Got GPS data...\n");

                for(i=4; i < data_waiting; i++){
                        if( (GPSBuffer[i-4] == 'O') && (GPSBuffer[i-3] == 'A') && (GPSBuffer[i-2] == 'T') && (GPSBuffer[i-1] == ','
                                printf("Single found\n");
                                int k = 0;
                                while(GPSBuffer[i+k] != ','){
                                        lat_gps[k] = GPSBuffer[i+k];
                                        k++;
                                }
                                lat_size = k;
                                i += k+1;
                                k = 0;
                                while(GPSBuffer[i+k] != ','){
                                        long_gps[k] = GPSBuffer[i+k];
                                        k++;
                                }
                                long_size = k;
                        }
                }

                // Begin Text File Output
                FILE *textOutput = fopen("SICK_Data.csv", "a");
```

```
                // Output GPS data
                for(i=0; i < lat_size; i++){
                        fprintf(textOutput,"%c", lat_gps[i]);
                }
                fprintf(textOutput,",");
                for(i=0; i < long_size; i++){
                        fprintf(textOutput,"%c", long_gps[i]);
                }
                fprintf(textOutput,"\n");

                fclose(textOutput);
                //End Text File Output
        }

        delete [] GPSBuffer;

        return 1;
}else{
        printf("Com port failed to open\n");
        return 0;
}
```

Final Words

This tutorial's objective was to show how to set up and use a NovaTel differential GPS system.

Click here to email me.
Click here to return to my Tutorials page.