

# Preliminary evaluation of accuracy improvement in Arduino-based single-frequency GPS receiver using ionosphere simulation data

Guilherme V. Hollweg<sup>ab1</sup>, Adriano Petry<sup>a</sup>, Telmo S. Klipp<sup>a</sup>, Everson Mattos<sup>a</sup> and Fernando E. Puntel<sup>a</sup>

<sup>a</sup>Southern Regional Space Research Center, National Institute for Space Research

<sup>b</sup>Federal University of Santa Maria, Electrical Engineering Department  
Av. Roraima, nº 1000, Cidade Universitária, Santa Maria (RS), Brazil

Received on January 01, 2015 / accepted on \*\*\*\*\*, 2015

## Abstract

The advent of Global Positioning System (GPS) technology made it possible the development of an arising number of embedded GPS signal receivers applications (e.g. cars and handheld units). GPS is a worldwide radio-navigation system composed by a constellation of 24 satellites, located in individual orbits of about approximately 20,200 Km height. It is known that the major single-frequency GPS positioning error is due to ionosphere state. Dual-frequency GPS receivers can compensate ionosphere refraction by applying ionosphere-free linear combination of observed L1 and L2 phases to correct time delay. Single-frequency GPS receivers, as those found in many commercial applications, cannot apply the same technique and, instead, relies on the use of ionospheric behavior models and estimation error techniques. Space weather program at National Institute for Space Research (INPE) have developed an operational ionosphere dynamics prediction system that provides a time-varying 3D grid with ion and electron concentration estimates in ionosphere. So, if the path from a GPS receiver to a satellite used in positioning is known, it is possible to estimate the slant total electron content (based on ionosphere simulation data) to be used in a single-frequency GPS ionosphere correction algorithm. The aim of this paper is to investigate the possibility of improving the positioning accuracy using an Arduino-based single-frequency GPS receiver and ionosphere simulation data. At this time, it was developed an algorithm for Arduino to acquire NMEA format sentences from Arduino-compatible GPS receiver GY-GPS6MV2 module, based on uBlox NEO-6M hardware. It was verified that among all NMEA sentences provided, information needed to reconstruct and improve the positioning using ionosphere data is not available. On the other hand, a different version of GPS receiver hardware is able to provide raw data information sufficient to the task.

**Keywords:** GPS, Arduino, ionosphere.

---

<sup>1</sup>E-mail Corresponding Author: guilhermehollweg@gmail.com

## 1. Introduction

GPS is a worldwide radio-navigation system developed, mainly, to provide date and time, location and velocity for objects near the Earth's surface. To provide positioning service, GPS satellites signals need to transverse the Earth's atmosphere to reach GPS receivers. Each satellite is responsible for sending a message with its orbit position (ephemeris) and other satellites positions (Almanac data) along its identification and the time signal was sent. Distances are estimated using satellite orbit position and measuring time delay for the signal to travel from satellite to the receiver [1].

Despite the high frequency of GPS signals, it is known that the ionosphere layer, that ranges from approximately 50 to 1000 Km above Earth's surface, is responsible for the most significant positioning error in a GPS receiver. The signals interact with free electrons of ionosphere which acts as a dispersive medium producing refracting effects (i.e., time delay and phase advance) proportional to the Total Electron Content (TEC) along propagation path. As a result, GPS signals present altered propagation velocity, which results in range error calculation [1].

Dual-frequency GPS receivers can compensate ionosphere effects by applying ionosphere-free linear combination of observed  $L_1$  and  $L_2$  phases to deduce TEC along propagation path and almost eliminate time delay by taking advantage of the dependence of delay on frequency [1]. However, single-frequency GPS receivers rely on the use of ionospheric behaviour models and estimation error techniques without reaching the accuracy level provided by dual-frequency GPS receivers.

### 1.1 Ionospheric Correction Algorithms

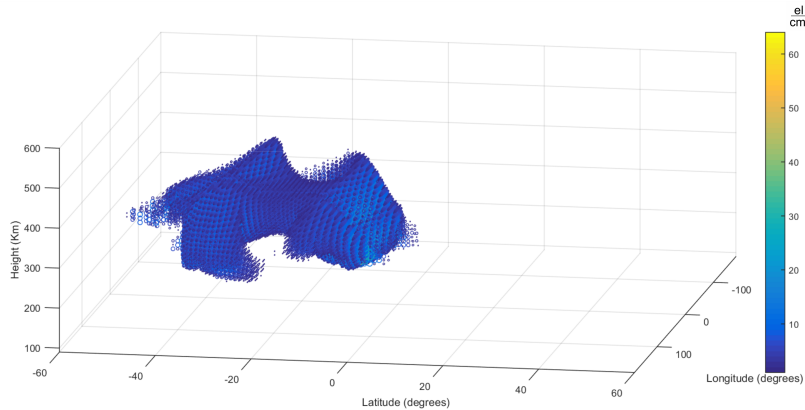
The solution for ionospheric errors can be simplified as determining the slant TEC along a GPS signal propagation path to compute and compensate time delay. However, the ionosphere is spatially inhomogeneous with TEC varying primarily as a function of local time and location as ionosphere ionization largely depends on sun radiation [2]. Additionally, satellite motion and user view angle may increase ionosphere errors [1].

Due to low cost and broad use of single-frequency GPS receivers, several methods were developed to model some level of ionosphere behavior to correct ionospheric errors. Bent [3] developed an empirical electron density model based on statistical analysis, with ionospheric electron density as a function of latitude, longitude, time, season, and solar radio flux, besides using ionospheric data. Klobuchar [4] presents a simple ionospheric model and correction algorithm consisting of a cosine representation of time delay

diurnal curve, which varies in period/amplitude and requires the user to approximate geomagnetic latitude, longitude, elevation angle and azimuth to each GPS satellite to discover the relative time delay.

## 1.2. Ionosphere simulation system

Space weather program at National Institute for Space Research (INPE) have developed an operational ionosphere dynamics prediction system [5] that provides a time-varying 3-dimensional grid with ion and electron concentration estimates in ionosphere. The system is based on the Sheffield University Plasmasphere-Ionosphere Model (SUPIM), that models several physical and chemical processes to describe the ionization in Earth's ionosphere and plasmasphere. A number of simulation runs provides 2-dimensional outputs aligned with Earth's magnetic field lines, which are approximated using International Geomagnetic Reference Field (IGRF-11). Post-processing includes the interpolation of simulation points to a 3-dimensional homogeneous grid with spatial resolution of 1 degree of latitude per 1 degree of longitude per 10 Km of altitude. Figure 1 shows a global ionosphere simulation for September 3rd, 2016 at 14:00h UT, where only high electronic concentration values are plotted. The system operation provides forecast data daily using 1 hour temporal resolution, and maps of TEC for South America region are available at <http://www2.inpe.br/climaespacial>.



**Figure 1** - Ionosphere simulation for September 3rd, 2016 at 14:00h UT.

## 2. Proposed approach

In this work, we investigate the possibility of improving the positioning accuracy using an Arduino-based single-frequency GPS receiver and iono-

sphere simulation data. The idea is to use an open hardware and inexpensive platform to provide high precision positioning, ideally similar to dual-frequency GPS receivers.

## 2.1 Arduino Platform and GPS module

Arduino is an open-source electronic platform based on easy-to-use hardware and software, created in Italy during year 2005 [6]. Because of its open source characteristic, anyone can have access to the hardware design, promote improvements and share it with community. For these applications it is necessary to plug-in Arduino shield in board. Those shields can add more features, like GPS, Ethernet, accelerometer, among other functionalities, what made Arduino boards to be used in several areas, including not only robotics, but education [7], renewable energy [8] and acquisition of meteorological data [9].

The GPS shield module used in this work is GY-NEO6MV2 that contains a NEO6M single frequency GPS chip [10] to provide positioning data. The NEO6 series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. The 50-channel u-blox 6 positioning engine have a Time-To-First-Fix (TTFF) of under 1 second in hot start and 27s in warm and cold starts. It is a surface-mount device (SMD) type 24-pin chip with a compact size.

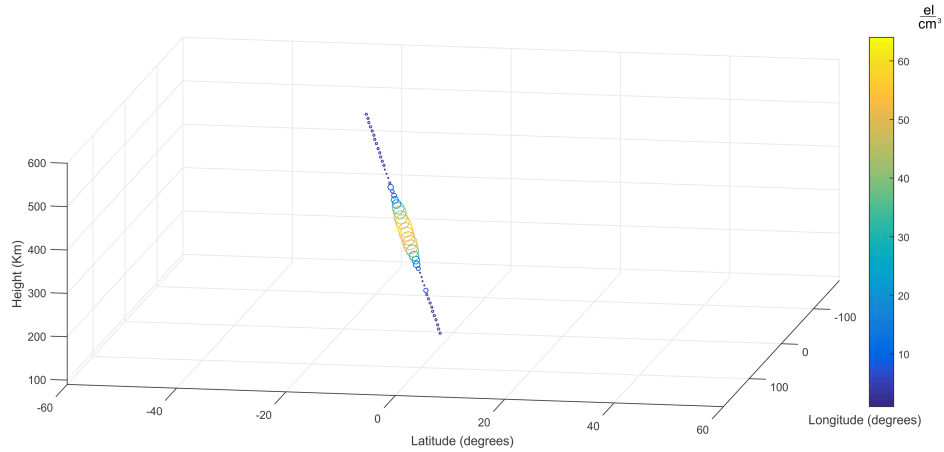
## 2.2 Slant TEC evaluation and correction algorithm

Since the approximate path from GPS receiver to a satellite used in positioning is available through Almanac data, it is possible to use the ionosphere simulation data to estimate the slant TEC in ionosphere from every satellite in view to the GPS single-frequency receiver, which is responsible for the time-delay in radio waves propagation. Figure 2 shows the electronic concentration, obtained from the simulation shown in Figure 1, only for the path from an hypothetical GPS receiver to a satellite with an elevation angle of 11.31 degrees. In the figure, the circles size are proportional to electronic concentration.

The ionospheric range error is inversely proportional to the square of carrier frequency (1575.42 MHz for L1), and directly proportional to the integral of electron density along the path. It is known that radio waves can be delayed in ionosphere by more than 300ns, which corresponds to range errors of more than 89.9m, considering the speed of radio waves in vacuum of  $299,792,458 \text{ m.s}^{-1}$ . The approximate path increase due to the effect of dispersion [11, 12] can be written as

$$error \approx \frac{K}{f^2} slantTEC, \quad (1)$$

where  $K = 40.3m^3.s^{-2}$ ,  $f$  is the signal frequency in Hz and  $slantTEC$  is the integral of electron density along the path from satellite to receiver, commonly expressed in TECU, where  $1 \text{ TECU} = 10^{16} \text{ electrons.m}^{-2}$ . For the case shown in Figure 2, the calculated slant TEC was 121.71 TECU, equivalent to a range error of 19.8 m. Such range errors can be estimated separately for every satellite used in GPS receiver and then regularly combined to produce the enhanced positioning.



**Figure 2** - Electronic concentration for the path between hypothetical GPS receiver and satellite.

### 3. Experiments and Preliminary Results

The embedded system hardware used in this work was composed by an Arduino UNO board, a GY-NEO6MV2 module and a SD card module. The microcontroller responsible to manage the hardware communication is an ATMEGA328 chip, that is an Atmel 8-bit AVR RISC-based microcontroller [13].

The GPS module contains NEO6M chip, an SMD voltage regulator MIC5205-3.3BM5, a SMD 432RP EEPROM memory, external antenna and a battery to ensure Real Time Clock (RTC) working on chip. The GPS-microcontroller communication used was a RS232-based serial protocol with adjustable baud rate.

The SD card module contains a SD connector, an AMS1117 SMD voltage regulator, filter capacitors and resistors to guarantee low signal-state in not used pins. The memory card was used as a datalogger, to receive and save all protocol information acquired via GPS. The communication protocol between SD card and ATMEGA328 is serial peripheral interface (SPI), a synchronous serial data protocol used by microcontrollers for communicating with one or more peripheral devices quickly over short distances [14].

### 3.1 Hardware Design

To improve hardware connection and become the system plug and play, avoiding malfunction due to movement, impacts and vibration, it was prototyped a 7x5cm board connected on Arduino headers. The additional hardware was created using an universal soldering prototype board, containing 2 male headers, 2 female headers and 2 100nF ceramic capacitors to enhance power source signal quality. The GPS RX pin was connected in Arduino digital pin 7, while TX pin was connected in Arduino digital pin 8. The SD SPI pins were connected in Arduino digital pins according to Table 1.

Table 1: SD SPI hardware connection

SD	Arduino
MOSI	11
MISO	12
SCK	13
SS	10

The Master Input Slave Output (MISO) pin, Master Output Slave Input (MOSI) pin and Standard Clock (SCK) pin are connected in correspondent Arduino UNO pins, as shown in Table 1, because they are the only ATMEGA328 pins that support SPI communication. However, Slave Select (SS) pin could be connected to any microcontroller general I/O pin, but it must have its connection explicitly declared in software and must be always in boolean low-state to avoid communication issues between external memory and microcontroller. The ATMEGA328 connections in board and the header connections developed in embedded system hardware schematic are represented in Figure 3. Moreover, a picture of hardware mounted and working is shown in Figure 4.

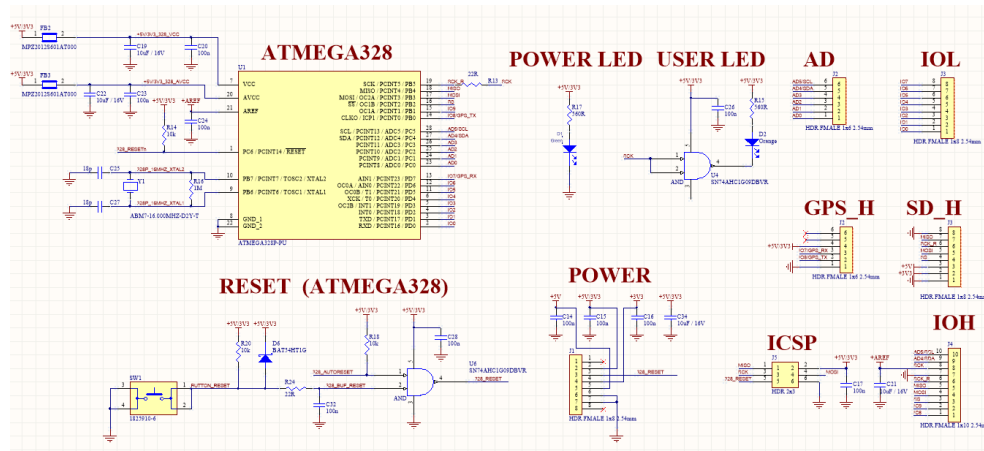


Figure 3 - Embedded system hardware schematic.

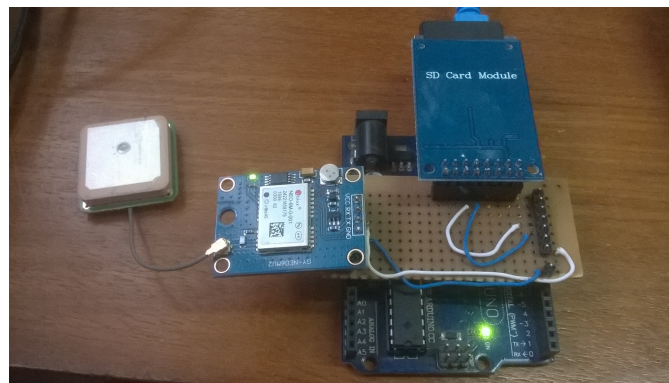


Figure 4 - Prototyped hardware mounted on Arduino board with GPS and SD card modules.

### 3.2 Software Design

The microcontroller programming language used to develop the software was C++, an object-oriented general-purpose programming language, that also provides access to low-level memory manipulation. The National Marine Electronics Association (NMEA) is a protocol for communication between marine electronics such as sonars, anemometers, GPS receivers and many other types of instruments [15]. The code developed aims at tracking NMEA sentences received by GPS module, save the information in SD card to posterior evaluation and display in computer screen. It was observed that NEO6M GPS receives easily generic and most used NMEA sentences, such

as GPS DOP and Active Satellites Data (GPGSA) and Global Positioning System Fix Data (GPGGA). In other hand, more specific NMEA sentences, like Global Positioning Satellites in view (GPGSV) were harder to observe. In several cases, when received, the sentences provided by GPS chip were in a corrupt format, with missing or unfit data, and so, unable to be used.

The developed software receives broadcasted satellite data in NMEA format via User Assynchronous Receiver/Transmitter (UART), stores all sentences in SD card and enables file scanning for specific sentences, like GPGSA or GPGSV. Table 2 shows different NMEA sentences acquired during a short period, while table 3 shows part of an organized GPGSA set of data acquired during a night observation.

Table 2: Example of NMEA sentences received

Sentences
\$GPRMC,023839.00,A,2940.90148,S,05348.87911,W,0.112,,110916,,A*77
\$GPGGA,023958.00,,,,,0,03,6.22,,,*
\$GPVTG,,T,,M,0.459,N,0.85,A6
\$G,5509090,3.8,15.,7,85,6PA,221,,,,,,3484
\$G,2402082,3.84,03347,5,6
\$G,,1011,06421,,3100,3
\$G,22602,84522507215527

Table 3: Part of the received GPGGA NMEA sentence

Sats	HDOP	Lat	Lon	Date	Time	Checksum
4	4.86	-29.681606	-53.814682	08/28/2016	02:37:58	717
4	4.86	-29.681604	-53.814685	08/28/2016	02:37:59	720
4	4.86	-29.681606	-53.814689	08/28/2016	02:38:00	721
4	4.86	-29.681604	-53.814693	08/28/2016	02:38:01	722
4	4.86	-29.681604	-53.814697	08/28/2016	02:38:02	725

On table 3, Sats refers to the number of satellites that GPS is receiving information from (not those in view), Horizontal Dilution of Precision (HDOP) is related to GPS error in horizontal position (in meters), latitude (LAT) and longitude (LON) provides hardware calculated position, and Checksum shows the total number of data packets that were lost during transmission. It can be seen that some packets were lost between one received data packet and another.



#### 4. Discussion

After developing the embedded system hardware and software, receiving and evaluating data, we realized that the acquired NMEA data is not sufficient to develop an algorithm capable of improving GPS accuracy with NEO6M single-frequency GPS. These problems are related to the fact that this specific GPS model does not allow user to access raw data, that is the unprocessed satellite data, before positioning estimation and not in NMEA format. NMEA sentences that provide satellite information - GPGSV do not supply sufficient information to be used in the positioning accuracy improvement algorithm.

We verified that GPS hardware models NEO6T or NEO6Q are able to provide, besides NMEA sentences, satellite raw data. These GPS chips are from the same family of NEO6M, that was used in this work. So, accessing raw data, it would be possible to evaluate timestamp (time when message was broadcasted by satellite), estimate satellite position, verify satellites in view received information, and correct ionosphere delay using ionosphere simulation system data. Using this approach, we verify that the positioning accuracy improvement algorithm would need to access and store the ionosphere simulation data periodically. Moreover, the ionic and electron concentration file data, generated by the simulation system, are large and would consume a large memory space. On the other hand, we expect an improve in positioning accuracy even higher than current ionosphere correction algorithms, since they simplify significantly the ionosphere dynamics modelling. The trade-off between complexity and accuracy has to be well evaluated. It is important to stress that the proposed approach could also be adapted to be applied to correct ionosphere delay for other positioning systems, like Glonass and Galileo, according to satellite received raw data.

#### References

- [1] Grewal, M.S., Weill, L.R., Andrews, A.P. Global positioning systems, inertial navigation, and integration. John Wiley & Sons, 2007.
- [2] Echer, E., Gonzalez, W.D., Guarnieri, F.L., Dal Lago, A., Vieira, L.E.A. Introduction to space weather. *Advances in Space Research*, 35(5), 855-865, 2005.
- [3] Bent, R.B., Llewellyn, S.K. Documentation and description of the Bent Ionospheric Model. Space & Missiles Organization, Los Angeles, USA, 1973.

- [4] Klobuchar, J.A. Ionospheric time-delay algorithm for single-frequency GPS users. *IEEE Transactions on aerospace and electronic systems*, (3), 325-331, 1987.
- [5] Petry, A., Souza, J.R., Velho, H.F.C., Pereira, A.G., Bailey, G.J. First Results of Operational Ionospheric Dynamics Prediction for the Brazilian Space Weather Program. *Advances in Space Research* 54: 22-36, 2014.
- [6] ARDUINO. We are Arduino.  
Available on: <https://www.arduino.cc/en/Main/AboutUs>. Accessed in: 20 Sep., 2016.
- [7] Araújo, A., Portugal, D., Couceiro, M.S., Rocha, R.P. Integrating Arduino-based educational mobile robots in ROS. *Journal of Intelligent & Robotic Systems* 77.2: 281-298, 2015.
- [8] Ramos, R.L., de Castro, L.F.O., Bertagnolli, S.C., Hbler, P.N. Uso da Plataforma Arduino para criar um dispositivo robótico otimizando o Aproveitamento de Energia Solar. *Computer on the Beach*: 406-408, 2015.
- [9] Torres, J.D., Monteiro, I.O., dos Santos, J.F., Ortiz, M.S. Aquisição de dados meteorológicos através da plataforma Arduino: construção de baixo custo e análise de dados. *Scientia Plena* 11.2, 2015.
- [10] NEO-6: U-blox 6 GPS Modules Datasheet, 2011.
- [11] Hoque, M.M., Jakowski, N. An alternative ionospheric correction model for global navigation satellite systems. *J Geod* 89:391-406, 2015.
- [12] Odijk, Dennis. Fast precise GPS positioning in the presence of ionospheric delays. NCG Nederlandse Commissie voor Geodesie Netherlands Geodetic Commission, 2002.
- [13] Atmel 8271J AVR ATmega Datasheet, 2015.
- [14] Davis, P.G., Huynh, Q.B., Peck, J.C. Serial peripheral interface. U.S. Patent No. 8,589,717. 19 Nov. 2013.
- [15] National Marine Electronics Association. NMEA 0183—Standard for interfacing marine electronic devices. NMEA, 2002.