Carrier Phase DGPS

João Miguel Duarte Mouro Alves de Andrade
joao.m.a.andrade@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2016

Abstract

Until GPS accuracy stopped being purposefully degraded in 2000, DGPS was used extensively to increase the positioning precision. These algorithms rely on a base station, whose position is known with great accuracy, to deduct the relative position – or baseline – of one or more rover receivers, by use of techniques to get rid of the common error sources. These techniques can be used on their own, or augmented through the use of Kalman Filtering or other algorithms. Receivers used to employ these techniques have to output a raw measurement of the pseudorange to each satellite, not just the position in geodetic or Cartesian coordinates. This type of device is known as a surveying receiver. Higher-end surveying receivers also track and output a measurement of the carrier signal phase sent by the satellite. Using this signal, having a much higher frequency, centimetre-level accuracy can be achieved. The use of carrier phase to implement DGPS is usually referred to as Real Time Kinematics (RTK). In this work, the development of a carrier phase DGPS solution will be presented. The goal is to implement a standalone, low cost system that delivers performance similar to that of commercial systems, using an accessible embedded platform.
Keywords: GNSS, Differential GPS, Real-Time Kinematic, Embedded platforms

1. Introduction

1.1. Motivation

When the Global Positioning System was approved for public use, its accuracy was deliberately degraded through a mechanism called Selective Availability (SA). To get around this situation, Differential GPS (DGPS) saw great development. Knowing the exact position of a receiver (known as the base station) and the position of the visible satellites, corrections to the code pseudoranges can be calculated and broadcast to nearby receivers (rovers), increasing positioning accuracy. As SA was disabled in 2000, this area lost some of its interest.

In its turn, modern higher-end GPS receivers can output not only regular code pseudoranges, but also a measurement of the carrier phase. Since these have accuracies in the order of tenths of a cycle, and the carrier phases have wavelengths of around 20 centimeters, centimetre-level accuracy can theoretically be achieved. This type of positioning is known as Real-Time-Kinematic (RTK) positioning. These receivers are also becoming smaller and more affordable, allowing new implementations such as in Unmanned Aerial Vehicles (UAVs).

The last few years also saw a rise in the popularity of the ”Maker” movement, and embedded platforms like the Arduino and the Raspberry Pi have become viable options for implementing computationally powerful, yet energy-saving and affordable systems.

1.2. State of the art

When the GPS was made available for civilian use, the system was the only one of its kind at full operating capacity, even if its accuracy was deliberately degraded for security reasons.

However, since satellite contribution to the observation error is the same for two close-by receivers, it can be mitigated by differencing two measurements that contain said error. This is known as single differencing. By differencing two single difference measurements that belong to the same set of receivers, but two different satellites, we can get rid of the two receiver clock errors. These are known as double differences and are fundamental to a collection of algorithms of DGPS.

Since the GPS is operated and maintained by the US Department of Defense and Department of Transportation, its civilian use can be restricted in exceptional cases, keeping the military functionality intact. Because of this possibility, several governments and international entities started their own GNSS programs. These include the Russian federation, with GLONASS, the European Com-

munity's Galileo, the Chinese system Beidou, the Japanese QZSS and Indian IRNSS.

Even if the end-user doesn't usually care for which system is available, as long as it works, the consequence of more systems in action is obvious: more data is accessible to process, potentially making positioning more precise. This thesis will stick to observing and processing GPS signals.

In its turn, the embedded system market has seen a tremendous rise in popularity since the early 2000s. The Arduino microcontroller platform, launched in 2005 [5], uses 8-bit Atmel AVR or 32-bit Atmel ARM processors and has sold over 700,000 official boards (derivative and clone boards are not accounted for).

In 2012, the first version of the Raspberry Pi was released. This is a low-cost, credit-card sized computer [6] with a full desktop operating system (several distributions of GNU/Linux are available, as is a Microsoft Windows version core),and uses a 32-bit ARM processor. As of February 2016, over eight million units have been sold [7], and the most recent version is the Raspberry Pi 3, which features a quad-core 1.2 GHz 64-bit processor and 1 GB of RAM.

These platforms are two examples of affordable systems that allow a user to develop their own solution, if so inclined. There are, however, drawbacks to both of them:

- The Arduino lacks computational power and memory, but has support for clock interrupts (on the AVR versions). It also lacks a Floating Point Unit (FPU), so its floating point operations are emulated, and does not support double precision;
- The Raspberry Pi does not run a real-time operating system, and so has to rely on operating system timing primitives. This is much less robust than the AVR counterpart, since the former uses registers to trigger interrupts. However, computational power is much more abundant. Storage and connectivity are also stronger points of the Raspberry Pi.

Since the processing does not require very precise timing (the GPS receivers already timestamp the data to be processed), and the lack of clock interrupt capabilities of the Raspberry Pi can be somehow mitigated with the faster clock speed and multi-core architecture, that was the embedded platform chosen for this work.

2. Background

In this section, we will present an overview of the reference coordinate systems used in this work, the segments that compose the Global positioning system, the measurements that can be deducted from the signals, and the error sources that pollute these observables.

2.1. Reference coordinate systems

Three reference coordinate systems are used throughout this work: Earth-Centered Inertial (ECI), Earth-Centered Earth Fixed (ECEF), and local coordinates. A brief explanation of each one follows.

The ECI frame is an inertial frame, conveniently placed to describe points on or near the Earth. It is defined with its origin being the center of mass of the Earth, the X axis pointing to the vernal equinox of J.2000 (Julian date of January 1, 2000), with the XY plane coinciding with the Earth's equatorial plane, and the Y and Z axes completing a right handed frame.

Since it is an inertial frame, the rotation motion of the Earth is not present, and a fixed point on the ground won't conserve its coordinates. However, satellite vehicles obey the laws of motion and gravitation on an inertial frame, where the coordinates are calculated, and then converted to the more usable system ECEF.

The ECEF frame is not an inertial frame, but, as stated above, it gains the convenience of including the rotational motion of the Earth, so a stationary point (such as a base station) conserves its coordinates over time. It is defined with its origin being the center of mass of the Earth, the Z axis being the mean rotational axis of the Earth, the X axis pointed in the direction of the Greenwich Meridian, and the Y axis completing a right-handed frame. From these XYZ coordinates, the latitude, longitude and altitude of a point can be easily deduced, but it is necessary to define the shape of the Earth. Different models for the shape of the Earth are called datums. The GPS uses the World Geodetic System 1984, the parameters of which are available in [2].

In certain applications, it becomes convenient to represent coordinates with respect to a base station, as opposed to the centre of the Earth. For example, when tracking satellites, the ones higher above the horizon have a shorter signal path through the ionosphere, so they'll provide less erroneous measurements. So, if it becomes necessary to establish one satellite as the reference one, it is generally the one with the highest elevation. It is also common to apply an elevation mask to discard satellites that are lower than a certain angle. Two local coordinate systems are used in this work.

The East, North, Up (ENU) frame is defined with the origin being the position of the reference station, the Up axis is normal to the geoid surface on the origin, while the East and North axes point, respectively, to East and North. This system will be

used to locate the rover receiver with respect to the base station.

Azimuth/Elevation coordinates can be derived from the ENU ones, and are particularly useful to locate satellites in the sky. The elevation angle is the angle the ENU vector makes with the EN plane (-90°$< \theta <$ 90°), while the Azimuth is the angle that the position vector makes with the NU plane (0°$\leq \psi <$ 360°). Azimuth angles are positive, going clockwise looking down on the EN plane. A third coordinate, distance (typically represented by ρ) can be added to make this a generic spherical coordinate system.

2.2. GPS Overview

The GPS is a Global Navigation Satellite System (GNSS) developed and maintained by the United States Department of Defense and Department of Transportation. It was originally created to provide a means for the military forces to continuously and accurately determine their position, velocity and time. However, in 1983, Korean Airlines Flight 007 diverted off-course, wandering into USSR airspace and being shot down by Soviet defences. After this incident, the GPS was made available for civilian use, with degraded accuracy, known as SA. Intentional errors were introduced in the pseudorange measurements, and in the satellite orbital information. SA was switched off in 2000.

The GPS consists of three segments:

- The space segment, which is the constellation of satellites. The original arrangement included 24 satellites, although today 31 are actively broadcasting signals, with several spares waiting in orbit, in case a malfunction occurs. The constellation is arranged so at least 4 satellites are visible over 10°anywhere on Earth, providing constant global coverage.

- The control segment is in charge of monitoring the satellites' health, and making sure the broadcast signals are accurate and up-to-date. Monitor stations throughout the planet track the visible satellites and relay the measurements to the master control station in Colorado, in the United States. These measurements are then processed and the parameters are uploaded to the satellites. These updates occur at least once a day.

- The user segment includes all the receivers that use the GPS signals to determine their position, velocity and time. Since there is no communication from the receivers to the satellites, the user capacity of the system is unlimited.

2.3. Observables

There are three types of measurements deducted from the time or phase differences between received, and receiver-generated signals:

- Code pseudoranges;

- Phase pseudoranges;

- Doppler measurements.

Code pseudoranges are formed using the travel time of the signal from the satellite to the receiver, multiplied by the speed of light. Since these are affected by receiver and satellite clock errors, as well as other sources, it is not a true range. It is the type of observable that the most common sort of GPS receiver uses to compute its position, usually through linearisation and a least squares algorithm.

Phase pseudoranges are much more accurate than their code counterparts, because the frequency is much higher, and can be measured more accurately. However, since the carrier phase is a periodical signal, the measured phase differs from the actual "phase range" by an integer number of cycles. This is known as an integer ambiguity, and it must be resolved if phase measurements are to be used as high-precision pseudoranges.

The Doppler shift of a wave is related to the radial velocity between the observer, and the source of the wave. In practice, this measurement of the deviation from the nominal frequency is used for velocity determination, or to aid in the resolution of integer ambiguities.

The error sources will now be presented. Code and phase pseudoranges introduced are affected by systematic errors, as well as random noise. Doppler data, on the other hand, is influenced by rates of change of the errors that affect the first two observables. These errors can be :

- Satellite-related: caused by the error in the satellite clock or its correction parameters, or in the broadcast orbital parameters;

- Propagation medium-related: the ionosphere is a dispersive medium for electromagnetic waves, which delays the signal. The longer the path through the ionosphere, the bigger the delay will be, so this can be treated as a function of the satellite elevation. In its turn, the troposphere, located in the lower 20 km of the atmosphere, is not a dispersive medium at GPS frequencies, but it is a refractive medium, bending the path of the signal as per Snell's law. Urban areas present one last source of

error: multipath. Buildings can reflect the signals and the same signal can arrive in the receiver antenna through different paths, at different times.

- Receiver-related: The signal acquisition process can introduce noise in the measurements, created by thermal noise of the electronic components. The signal path through the receiver may also introduce delays that must be accounted for.

The following is a table of the errors, and their contribution to the user equivalent range error (UERE).

Source	Nominal value [m]
Satellite clock	1.1
Ephemerides	0.8
Ionosphere	7
Troposphere	0.2
Multipath	0.2
Receiver	0.1
Total	7.1

Table 1: Pseudorange Error Budget, [4]

3. Implementation

The rover receivers (indicated by a subscript r) can be located with respect to to a base station (indicated by a subscript b), the location of which is known with great accuracy. The vector that points from the base station to the rover receiver is known as the baseline vector, defined as

$$b = X_r - X_b. \tag{1}$$

To compute this baseline vector, interferometric techniques are used. In short, measurements are subtracted from each other to get rid of the common sources of error.

3.1. Observables and double differences

Two types of observables are used: code and carrier phase measurements. As per [2], the measurement model is, for receiver k and satellite p:

$$\phi_k^p = \rho_k^p + \lambda N_k^p + c(t_k - t^p + T_k^p - I_k^p) + \epsilon_k^p, \tag{2}$$

$$PR_k^p = \rho_k^p + c(t_k - t^p + T_k^p - I_k^p) + \epsilon_k^p, \tag{3}$$

where ϕ_k^p and PR_k^p are code and carrier phase measurements (unit: meters), from receiver k to satellite p. ρ_k^p is the geometric distance between the receiver and the satellite, λ is the carrier wavelength, N_k^p is the unknown integer ambiguity (unit: cycles), c is the speed of light in vacuum, t_k and t^p are, respectively, the receiver and satellite clock offsets, T_k^p and I_k^p are, respectively, the tropospheric and ionospheric delays, and ϵ_k^p represents unmodelled sources of error, such as multipath and hardware interference.

Subtracting two carrier phase measurements pertaining to the same satellite and different receivers (let us call them k and m), we can get rid of the satellite clock offset, [4]. Assuming the baseline is short (under 10 km), the signal path between the satellite and the two receivers is the same, so the tropospheric and ionospheric delays are also cancelled. This is known as a single difference (SD):

$$\Delta\phi_{km}^p = \phi_k^p - \phi_m^p = \Delta\rho_{km}^p + \lambda\Delta N_{km}^p + c\Delta t_{km}^p + \Delta\epsilon_{km}^p. \tag{4}$$

The single differences of the receivers' clock bias are cancelled by double differencing, i.e., subtracting SD's from two different satellites (let the new satellite be q):

$$\begin{aligned}\nabla\Delta\phi_{km}^{pq} &= \Delta\phi_{km}^p - \Delta\phi_{km}^q \\ &= \nabla\Delta\rho_{km}^{pq} + \lambda\nabla\Delta N_{km}^{pq} + \nabla\Delta\epsilon_{km}^{pq}.\end{aligned} \tag{5}$$

For a given system, double differences are made always with respect to the same satellite, usually the one with the highest elevation, for reasons stated earlier. Following the same procedure for the code measurements yields the code double differences:

$$\nabla\Delta PR_{km}^{pq} = \nabla\Delta\rho_{km}^{pq} + \nabla\Delta\epsilon_{km}^{pq}. \tag{6}$$

The code double differences are much noisier, but that can be mitigated using a smoothing mechanism such as a Kalman Filter, [4].

3.2. Observables covariance

The noise of the measurements is assumed to have a Gaussian distribution, with expected value zero, and variance σ_ρ^2 and σ_ϕ^2, for code and carrier phase measurements respectively.

For a set of independent, identically distributed measurements given by a column vector z, and a disturbance column vector ϵ, the corresponding variance-covariance matrix is given by:

$$\begin{aligned}cov\,(z) = E\left\{(\epsilon - E\,(\epsilon))\,(\epsilon - E\,(\epsilon))^T\right\} = \\ E\left\{\epsilon\epsilon^T\right\} = \sigma_z^2 I_{n\times n}.\end{aligned} \tag{7}$$

For n measurements and where $I_{n\times n}$ is the identity matrix of order n.

For a single difference measurement (equation 4) for two receivers k and m (which results in the

SD error being $\Delta\epsilon_{km} = \epsilon_k - \epsilon_m$), the variance-covariance matrix for the SD vector is given by:

$$\begin{aligned} &cov\left(\Delta z_{km}\right) \\ &= E\left\{\left(\Delta\epsilon_{km} - E\left(\Delta\epsilon_{km}\right)\right)\left(\Delta\epsilon_{km} - E\left(\Delta\epsilon_{km}\right)\right)^T\right\} \\ &= E\left\{\left(\epsilon_k - \epsilon_m\right)\left(\epsilon_k - \epsilon_m\right)^T\right\} \\ &= E\left\{\epsilon_k\epsilon_k^T + \epsilon_m\epsilon_m^T\right\} = 2\sigma_z^2 I_{n\times n}. \end{aligned} \tag{8}$$

This results in a diagonal variance-covariance matrix, which means that single differences of a given measurement are uncorrelated.

Setting one of the measurements as the reference to build double differences (DD's), one uses a matrix to transform vector Δz_{km}, which has n single differences, into a vector with $n-1$ DD's:

$$\nabla\Delta z_{km} = M\Delta z_{km}, \tag{9}$$

where $M_{(n-1)\times n}$ is the transformation matrix:

$$M = \begin{bmatrix} 1 & & & -1 & & \\ & 1 & & -1 & & \\ & & \cdots & \vdots & \cdots & \\ & & & -1 & & 1 \\ & & & -1 & & & 1 \end{bmatrix}_{(n-1)\times n}. \tag{10}$$

We can now define the variance-covariance matrix of the double difference measurements:

$$\begin{aligned} &cov\left(\nabla\Delta z_{km}\right) = \\ &E\left\{\left(\nabla\Delta\epsilon_{km} - E\left(\nabla\Delta\epsilon_{km}\right)\right)\left(\nabla\Delta\epsilon_{km} - E\left(\nabla\Delta\epsilon_{km}\right)\right)^T\right\} \\ &= E\left\{M\Delta\epsilon_{km}\Delta\epsilon_{km}^T M^T\right\} = M\left(2\sigma_z^2 I_{n\times n}\right)M^T = \\ &2\sigma_z^2 MM^T. \end{aligned} \tag{11}$$

This method is used for pseudorange double differences, because pseudorange measurements are assumed to be independent and identically distributed.

Since for the carrier phase double differences, the ambiguity resolution algorithms are heavily dependant on the variance-covariance matrix of the weighted least squares estimation, a finer estimate of its variance is needed, so it is no longer assumed that the measurements are identically distributed.

For a single carrier phase measurement, we use a function of the satellite elevation for an estimate of variance from [9]:

$$\sigma_k^2 = \frac{a\left(b + \frac{b}{sin^2\theta_k}\right)}{\lambda^2}, \tag{12}$$

with θ_k being the elevation angle of satellite k, and a and b being two parameters set empirically. It follows from 8 that the variance-covariance matrix of the single difference measurements is given by:

$$cov\left(\Delta z_{km}\right) = \begin{bmatrix} 2\sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 2\sigma_n^2. \end{bmatrix}, \tag{13}$$

which we'll call σ_Δ. Applying the same process as 11, the variance-covariance matrix for the double difference measurements becomes:

$$\sigma_{\nabla\Delta} = M\sigma_\Delta M^T. \tag{14}$$

Regardless of defining the original observations as identically distributed or having a dependence on satellite elevation, the variance-covariance matrix of the double differences is not diagonal, which means these measurements are correlated. This is one of the biggest hurdles in the integer ambiguity resolution process.

3.3. Direction cosines

We define the direction cosine as being a unit vector pointing from receiver k to a given satellite p:

$$e_k^p = \frac{X_p - X_k}{\rho_k^p}e_x + \frac{Y_p - Y_k}{\rho_k^p}e_y + \frac{Z_p - Z_k}{\rho_k^p}e_z, \tag{15}$$

where (X_p, Y_p, Z_p) are the coordinates of an arbitrary point in the ECEF frame, and ρ_k^p is the true range between the two points.

Since dealing with short baselines and that the distances from the receivers to the satellites are of the order of 20,000 kilometers, the direction cosine for a given satellite is the same for two receivers close by. With this, we can define the single differences of true ranges as the dot product between the baseline vector and the direction cosine:

$$\Delta\rho_{km}^p = b \cdot e^p, \tag{16}$$

where e^p is now the LoS vector from the system to satellite p. The single differences of true ranges are then the projection of the baseline vector onto the LoS to the corresponding satellite, as explained in figure 1.

Differentiating 16 for two satellites, we then obtain the double differences of true ranges, expressed in terms of the baseline vector, and two LoS vectors:

$$\nabla\Delta\rho_{km}^{pq} = b \cdot \left(e^p - e^q\right) = b \cdot e^{pq}. \tag{17}$$

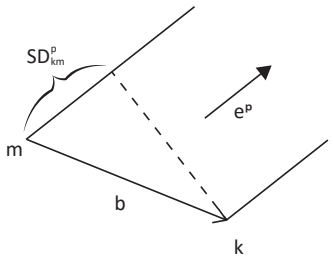

Figure 1: The interferometric technique, for one satellite.

3.4. Pseudorange-based differential positioning
By combining equations 6 and 17, for n satellites, one obtains the system:

$$y_{(n-1)\times 1} = B_{(n-1)\times 3} b_{3\times 1} + e_{(n-1)\times 1}, \quad (18)$$

where y is the measured double difference pseudorange vector, B is the system matrix for the baseline coordinates, b is the baseline vector, and e is the noise vector. The subscript in each matrix represents its dimension.

To solve for baseline, using a weighted least squares method, the definition of covariance matrix from equation 11 used to define the weighting matrix as its inverse:

$$W = cov\left(\nabla \Delta z_{km}\right)^{-1} = \left(2\sigma_z^2 M M^T\right)^{-1}. \quad (19)$$

And the baseline estimate is given by:

$$\hat{b} = \left(B^T W B\right)^{-1} B^T W y. \quad (20)$$

3.5. Carrier phase differential positioning
Building on 18 and adding the carrier phase double differences to the measurement vector:

$$y_{2(n-1)\times 1} = B_{2(n-1)\times 3} b_{3\times 1} + A_{2(n-1)\times(n-1)} a_{(n-1)\times 1} + e_{2(n-1)\times 1}. \quad (21)$$

Where the same notation as before applies, A is the matrix containing the ambiguity contribution to the phase double differences, and a is the DD ambiguity vector. As before, the subscript in each matrix represents its dimension.

Constructing an augmented system matrix $H = [A\ B]$ (its dimension being $2(n-1) \times ((n-1)+3)$), and the corresponding augmented state vector $x = \left[a^T\ b^T\right]^T$ (with dimension $((n-1)+3) \times 1$), the system is solvable if $2 \times (n-1) \geq (n-1)+3$, or $n \geq 4$.

As before, one starts with a weighted least squares method, defining the weight matrix from the observation variance-covariance matrix, then, the estimator is given by:

$$\begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} = \left([A\ B]^T W [A\ B]\right)^{-1} [A\ B]^T W y, \quad (22)$$

to which we will call the float solution, since it is known that the double difference ambiguities are integer, and this solution will be real valued. The next step is to fix the solution to integer values, for which we need the covariance matrix of the float solution. Defining the estimation error as $\tilde{x} = x - \hat{x}$:

$$\tilde{x} = -\left([A\ B]^T W [A\ B]\right)^{-1} [A\ B]^T W e, \quad (23)$$

e being the double differences noise. The covariance matrix for the estimation error is then:

$$\begin{aligned} Q_{\hat{x}} = Q_{\tilde{x}} &= E\left\{\left(\tilde{x} - E\left(\tilde{x}\right)\right)\left(\tilde{x} - E\left(\tilde{x}\right)\right)^T\right\} \\ &= \left([A\ B]^T W [A\ B]\right)^{-1} \\ &= \begin{bmatrix} A^T W A & A^T W B \\ B^T W A & B^T W B \end{bmatrix}^{-1} = \begin{bmatrix} Q_{\hat{a}} & Q_{\hat{a}\hat{b}} \\ Q_{\hat{b}\hat{a}} & Q_{\hat{b}} \end{bmatrix}. \end{aligned} \quad (24)$$

The float solution is a good starting point for search techniques, with the covariance matrix ($Q_{\hat{a}}$) defining the search space, [8]. The fixed solution $\check{a}$ is then given by the minimisation of the cost function:

$$\check{a} = arg\left(min_{a\in\mathbb{Z}} \|\hat{a} - a\|^2_{Q^{-1}_{\hat{a}}}\right). \quad (25)$$

Since the ambiguities are correlated, the search space in question is elliptical. In this case, the search space can be more elongated in one direction than another, which can result in erroneous search procedures. However, if we can transform the ambiguities to get rid of this correlation, and obtain a spherical search space, this problem will be mitigated. This is the goal of the LAMBDA method . A transformation is used to decorrelate the error, diagonalising the covariance matrix of the float solution. Because a true diagonalisation is hard to achieve, the focus will be in making the non-diagonal elements as close to zero as possible. Having this "nearly-diagonal" covariance matrix will result in a nearly spherical search space, making the search process faster and easier.

Defining the transformation matrix as Z, the covariance matrix of the transformed ambiguities is given by:

$$Q_{\hat{z}} = Z^T Q_{\hat{a}} Z. \quad (26)$$

It is necessary that both Z and Z^{-1} only have integer entries. This way, the original and transformed ambiguities remain integer.

Decomposing the covariance matrix of the float solution as

$$Q_{\hat{a}} = L^T D L, \tag{27}$$

and we can express $Q_{\hat{z}}$ as

$$Q_{\hat{z}} = Z^T L^T D L Z, \tag{28}$$

where $L_{(n-1)\times(n-1)}$ is a lower triangular matrix with every diagonal entry equal to 1, and $D_{(n-1)\times(n-1)}$ a diagonal matrix. Assuming Z is close to L^{-l} (making LZ and $L^T Z^T$ close to the identity matrix, and resulting in the intended "nearly-diagonal" covariance matrix $Q_{\hat{z}}$), the covariance matrix of the transformed ambiguities can be decomposed as in 27:

$$Q_{\hat{z}} = \tilde{L}^T \tilde{D} \tilde{L}, \tag{29}$$

where the non-diagonal elements of $\tilde{L}$, leading to the intended "nearly-diagonal" covariance matrix, [1].

Transforming the ambiguities using the generated Z matrix, the cost function is now given by

$$\check{z} = arg\left(min_z \|\hat{z} - z\|^2_{Q_{\hat{z}}^{-1}}\right), \tag{30}$$

with $z = Z^T a$ and $\hat{z} = Z^T \hat{a}$. Once the decorrelation is done, several techniques can be used to compute the transformed ambiguity set. Three are presented, sorted by increasing complexity.

- **Integer rounding** – with this method, the transformed ambiguity sets are simply rounded to the nearest integer. Along several epochs, it is expected that the most frequent rounded set is the actual transformed ambiguity set, since the double difference noise has been decorrelated and has null mean.
- **Volume search** – this method returns all ambiguity sets that verify the inequality

$$(\hat{z} - z)^T Q_{\hat{z}}^{-1} (\hat{z} - z) \leq \chi^2, \tag{31}$$

  which is the same as saying that it will return all ambiguity sets whose distance to the initial point ($\hat{z}$) is smaller than the search space volume χ^2, [1]. The search space volume must take in consideration the application at hand. Too few candidates and the true ambiguity set might not be found; too many and a real-time application won't be possible.
- **Search and Shrink** – the more elaborate algorithm returns a stipulated number of candidates, by computing conditional ambiguities and iterating until the residues are minimal.

After having the correct ambiguity set, the fixed baseline vector can be obtained directly, correcting the float baseline (from 22), [8], through

$$\check{b} = \hat{b} - Q_{\hat{b}\hat{a}} Q_{\hat{a}}^{-1} (\hat{a} - \check{a}). \tag{32}$$

Once the ambiguities are known, it might become advantageous to eliminate the pseudorange double differences from the baseline calculation. Removing the pseudorange component of equation 21, the resulting system is:

$$y_{(n-1)\times 1} = B_{(n-1)\times 3} b_{3\times 1} + A_{(n-1)\times(n-1)} a_{(n-1)\times 1} + e_{(n-1)\times 1}, \tag{33}$$

where once again the subscript indices represent the dimensions of the matrices and vectors involved.

Isolating the baseline terms and removing the error vector yields

$$Bb = y - Aa, \tag{34}$$

and using the definition of left pseudoinverse from [3], the weighted least squares solution is

$$b = \left(B^T B\right)^{-1} B^T \left(y - Aa\right), \tag{35}$$

which is a baseline vector computed using solely carrier phase measurements.

4. Results

In this section, an overview of the tests and results will be done. Four tests were made, each with different purposes. In each case, the resulting baseline in ENU coordinates is compared to the expected value, and the Circular Error Probable (CEP) accuracy is calculated. For Code DGPS and the LAMBDA solutions, 30 seconds averages and medians were taken in an effort to decrease variability and increase accuracy, at the expense of real-time positioning performance. These figures are also compared with the standalone, code-only solution, to assess the benefit of a differential setup.

4.1. Base test

The first test was performed with two stationary antennas present in the lab. It ran for one hour, and its purpose was to evaluate the validity of the algorithms and their performance with very short baselines. The results are compiled in table 2

The benefits of using differential positioning are apparent even using only code pseudoranges. Comparing with the announced optimal accuracy of the standalone u-blox receiver (2.5 metres CEP), the standalone accuracy was of 4.19 metres. Using differential positioning increases precision to 1.34 metres. The average and median operations

	Code	DGPS	LAMBDA	CP
Method		CEP [m]		
Raw	4.189	1.336	0.978	0.839
30s Av.		0.9317	0.813	
30s Med.		0.939	0.813	

Table 2: Precision comparison for the base test.

had the same effect, bringing it down to sub-metre precision.

Adding carrier phase to the baseline calculation also brings the accuracy to sub-metre level, with the unsmoothed LAMBDA baseline having a CEP of 97.8 centimetres. While the average and median operations, applied to the DGPS baseline, decreased the CEP value in 30 percent, applying them to the LAMBDA baseline only decreased it in 16 percent, to 81.3 centimetres in both cases. Using only carrier phase to calculate de baseline yielded a CEP of 83.9 centimetres, which is higher than the smoothed LAMBDA solutions, but better suited for real-time operations. This amounts to decimetre-level precision, where commercial solutions can offer centimetre-level precision.

4.2. Campus test

The second local test was made on campus, with a baseline length of around 100 metres. Using the same base antenna on every test, this was the first test performed with the mobile setup acting as the rover receiver, which is shown on figure 2, and it was done to test its feasibility.

	Code	DGPS	LAMBDA	CP
Method		CEP [m]		
Raw	7.720	3.252	2.696	3.146
30s Av.		3.132	2.753	
30s Med.		3.143	2.729	

Table 3: Precision comparison for the campus test.

The standalone precision was of 7.72 metres, compared to the 4.19 metres the previous test. Using differential positioning helped reduce this figure to 3.25 metres, which is 42% of the initial value. Introducing carrier phase observations helped the accuracy, with the CEP precision for LAMBDA being 2.696 metres, worse than the first test's sub-metre accuracy. The smoothing algorithms also did little to improve this, even deteriorating the accuracy. The carrier phase baseline also exhibited an accuracy worse than the LAMBDA one, on par with the DGPS solution.

The worse performance of this test may be due to three factors:

Figure 2: The solution developed for portable data collection: 1: Raspberry Pi and Wi-Fi adapter. 2: u-blox receiver. 3: Portable battery pack. 4: Breadboard with the LCD screen, button and LED's. The GPS antenna is not visible and sits at the top of the tripod, where a camera would usually be.

- The site was surrounded by two tall buildings, which blocked part of the sky and may have introduced some amount of multipath;
- The reference values retrieved from Google Earth may be off. It is possible that the equipment was a few metres to the East. This is corroborated by a constant error in the East component of the baseline. It is possible that the equipment was a few metres to the East;;
- The antenna used in this and the following tests was a different, smaller model, intended for portable applications. It is expected that it would have worse performance than the bigger model which includes a choke ring to reduce multipath interference.

4.3. Long baseline test

For the off-site test, a hill in a nearby park was chosen, since this site had good visibility of the sky, and sat at roughly 1600 metres from the base station.

The standalone precision was a bit better than the second test, which was to be expected since more of the sky was visible. These results are in line with the second test, and a limit to the precision seems to have been hit, with an apparent minimum of 2.7 metres. The sky was almost totally visible on this site, and the values show the same amount

	Code	DGPS	LAMBDA	CP
Method		CEP [m]		
Raw	6.888	3.081	2.722	2.723
30s Av.		2.840	2.696	
30s Med.		2.872	2.678	

Table 4: Precision comparison for the campus test.

of noise as the previous tests. Using average and median didn't have as much effect as in the first test. There are also biases on both the East and North components of the baseline vector. So, it is possible that, once again, the references are off by a few metres. For the East component, this is possible (there was a path that ran from east to west), but unlikely for the North component, which has a similar bias.

4.4. Final test

The final test was made to showcase the point logging capabilities of the setup. We returned to the park, which is filled with little paths and intersections, and pressed the logging button. The process was repeated for 14 points. Only carrier phase was used to calculate the coordinates. There were several points with multiple solutions, due to the integer ambiguity set changing over time, since the sets for each epoch were being averaged after 30 seconds with a consistent constellation.

Of the 26 measurements, the least accurate was 7.92 metres away from the true position. However, it is a considerable deviation from the average, since the mean error was of 3.64 metres. It should be noted that the average East and North errors, respectively 2.14 and 1.75 metres, should be much closer to zero, since the error in the signals observations have a null mean. This might be indicative of an error in the base receiver position.

4.5. Real-time applicability

For the final test, the real time applicability of the system was also tested. Each program (Code DGPS, LAMBDA, and carrier phase positioning) was ran three times, with the average times in seconds in table 5. In table 6, we included the performance of the system, measured in epochs processed per second.

	DGPS	LAMBDA	CP
Preparation	33.66	33.73	33.74
Processing	4.00	16.77	17.25
Total time	37.66	50.50	50.99

Table 5: Average running times in seconds, for the final test.

The phase labelled "Preparation" pertains to the initial part of the program, where the binary files from the base and rover are opened and decoded, and the double differences are formed. The "Processing" phase includes calculating the satellites' positions and direction cosines, and all the the remaining math to determine a baseline.

	DGPS	LAMBDA	CP
Preparation	49.38	49.27	49.26
Processing	415.5	99.1	96.35
Total	44.13	32.91	32.59

Table 6: Performance of each program, for the final test (in epochs per second).

We can see that the decoding of the information has the same performance in the three programs, since the data to be decoded is the same. This shows an average of 49 epochs of information prepared per second, or, that each epoch will take less than 25 milliseconds to prepare for processing, once the binary information is received.

For the processing phase, the Code DGPS algorithm is, as expected, the fastest of the three, attaining more than 400 epochs per second, so each epoch will be processed in less than 2.5 milliseconds. Adding the carrier phase double differences greatly decreases the performance of the system, due to the ambiguity resolution algorithms. Using the baseline vector output by the LAMBDA algorithm, the average performance was of 99 epochs per second, and using only carrier phase to calculate the baseline through weighted least squares brought the performance down to 96 epochs processed per second. It is then to be expected that the system be able to process one epoch of information in less than 20 milliseconds (50 epochs per second).

In total, we can see that the system is, on average, able to decode and process more than 25 epochs of information per second. This means that all the operations necessary, after the data has been received over serial port or over the internet, it will take less than 40 milliseconds to process and output a baseline. So, in a real time application, there is remaining computational power to implement the smoothing algorithms, be it the simple average and median shown in this chapter, or more sophisticated algorithms such as Kalman Filtering.

5. Conclusions

The main objective of this work was to develop a carrier phase DGPS solution that was implemented on an embedded platform, and that, with affordable components, could come close to the precisions offered by commercial solutions. While

the all the parts were not successfully implemented at the same time, all of them were, at one point, tested and successful on an embedded platform, namely:

- The communication, both over the internet with a base station, or with the local GPS receiver over USB;
- The data collection;
- The implementation of the algorithms on the platform, with promising results where the computational performance is regarded.

5.1. Future work

While the results were satisfactory, further work can be done to improve this solution. The considered approaches are:

- Solving the connectivity issues by successfully receiving and processing the serial data in real time.
- Improve the user interface by using more sophisticated means of user I/O. Relatively inexpensive touch screens are available for the Raspberry Pi and would solve the issues with user experience.
- Make the device more standalone like a true product. The developed prototype is not standalone. It requires a device to initially launch the programs, cancel the data collections, and provide internet access if applicable.
- Employ more sophisticated algorithms to the data processing. [9] adds Kalman Filtering to carrier phase positioning with satisfying results. These operations are, in principle, possible to implement on the Raspberry Pi without compromising real time operation. A rover receiver capable of L2 measurements would also highly increase solution precision.

References

[1] P. de Jonge and C. Tiberius. The lambda method for integer ambiguity estimation: implementation aspects. *Publications of the Delft Geodetic Computing Centre*, Aug. 1996.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle. *GNSS – Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. Springer, 1^{st} edition, 2008. ISBN:978-3-211-73012-6.

[3] M. James. The generalised inverse. *The Mathematical Gazette*, 62(420):109–114, 1978.

[4] E. Kaplan and C. Hegarty. *Understanding GPS – Principles and Aplications*. Artech House, 2^{nd} edition, 2005. ISBN:978-1-580-53894-7.

[5] D. Kushner. The making of Arduino. `http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino`, 2011. Accessed: 2015-07-07.

[6] Raspberry Pi Foundation. What is a raspberry pi? `https://www.raspberrypi.org/help/what-is-a-raspberry-pi/`, 2012. Accessed: 2015-07-07.

[7] Raspberry Pi Foundation. Raspberry pi 3 on sale now. `https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/`, 2016. Accessed: 2016-02-29.

[8] P. Teunissen. The least-squares ambiguity decorrelation adjustment: a method for fast gps integer ambiguity estimation. *Journal of Geodesy*, 70(1–2):65–82, Nov. 1995. doi:10.1007/BF00863419.

[9] S. Zhao, X. Cui, F. Guan, and M. Lu. A kalman filter-based short baseline rtk algorithm for single-frequency combination of gps and bds. *Sensors*, 14(8):15415–15433, Aug. 2014. doi:10.3390/s140815415.