ENHANCING GNSS RTK SOLUTIONS WITH RANDOM SAMPLE CONSENSUS

By

Angela Krebs

July, 2022

Director of Thesis: Dr. Zhen Zhu

Major Department: Department of Engineering

ABSTRACT

Precise positioning with Global Navigation Satellite Systems (GNSS) can be achieved by using the real-time kinematic (RTK) solution. A GNSS RTK solution was improved using an innovative approach based on Random Sample Consensus (RANSAC). RANSAC pre-filters carrier phase measurements based on the inherent consistency among GNSS satellites. This method improves the reliability and accuracy of an RTK system with low-cost receivers and antennas, especially in a GNSS-challenged environment. In this work, the integration of RANSAC and RTK is demonstrated with recorded live GNSS data.

ENHANCING GNSS RTK SOLUTIONS WITH RANDOM SAMPLE CONSENSUS

A Thesis

Presented to the Faculty of the Department of Engineering

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science

Mechanical Engineering

by

Angela Krebs

July, 2022

Enhancing GNSS RTK Solutions with Random Sample Consensus

By

Angela Krebs

Approved By:

Director of Thesis _____

Dr. Zhen Zhu

Committee Member _____

Dr. Faete Filho

Committee Member _____

Dr. George Wang

Chair of the Department of Engineering _____

Dr. Barbara Muller-Borer

Interim Dean of the Graduate School _____

Dr. Kathleen T. Cox

Table of Contents

LIST OF TABLES

LIST OF FIGURES

1. Introduction

The world is becoming more dependent on navigation technologies, such as the Global Positioning System (GPS), and more generally, Global Navigation Satellite Systems (GNSS). GNSS receivers can be the main navigational device of a large vehicle or run as an app on a cell phone. There are various available GNSS constellations launched by different countries. GPS from the United States, GLONASS from Russia, Galileo from Europe and BeiDou from China.

Among all the applications of GNSS, one area of particular interest in this thesis is navigation of autonomous vehicles. Figure 1 shows a good depiction of how autonomous ground vehicles could drive in the near future. Every car would have real-time precise location measurements and would be broadcasting them. Each car would also be equipped with other sensors, to sense the environment (such as a traffic jam). Collisions and other accidents may be minimized, and the safety of driving will be enhanced. To achieve this goal, autonomous vehicles require both accurate absolute and relative positioning, which is typically provided by GNSS.

Figure 1. The potential for autonomous vehicles on the road simultaneously [1].

GNSS receivers can help a vehicle locate itself relative to other vehicles or relative to the road. For example, automatic lane control can benefit from centimeter-level accuracy in positioning, especially when lane markers are not visible. Figure 2 shows that cars will be able to locate barriers and use accurate navigation input to maintain their lane without lane markers or other road signs.

It greatly improves not only safety, but also efficiency. Theoretically, if every car on the road could run with the same traffic information and with capable navigation devices, traffic jams would be eliminated as every car would always be driving with perfect awareness of other vehicles on the road.

Figure 2. Autonomous vehicles using lane control [2].

Another example is the control of vehicles when they enter an intersection with traffic lights. As this concept becomes more readily available, autonomous vehicles may be able to pace the speed to minimize time at red lights. This is one further step towards complete autonomous vehicles on the road. Figure 3 shows a diagram of what a potential intersection might look like.

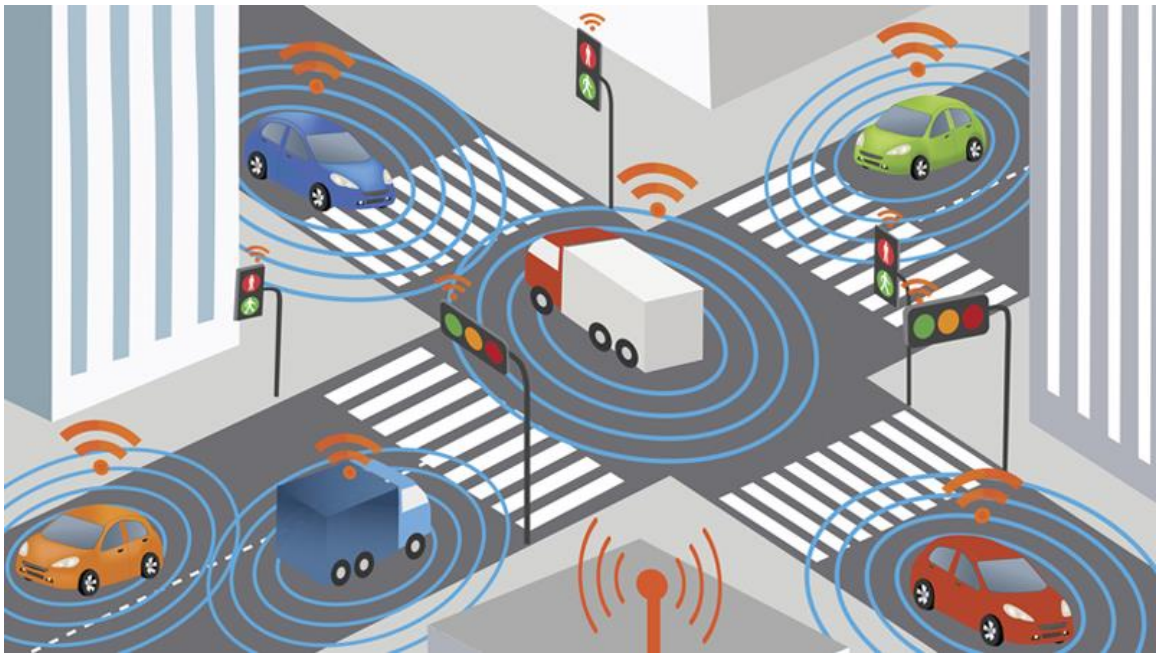

Figure 3. Traffic signals sent to vehicles within the vicinity [3].

In addition to ground vehicles, GNSS receivers are widely used on Unmanned Aerial Vehicles (UAVs). UAVs can be used in mapping, surveying, monitoring, delivery, and many other applications. Some small UAVs are equipped with precise relative positioning capability, such as the one shown in Figure 4.

Figure 4. A DJI Phantom 4 Drone with precise relative positioning capability [4].

Another booming area of research and applications of GNSS is in precision agriculture, which is a way to implement new technologies into farming to increase both the profits of the farm and the productivity [5]. This is done by decreasing the manual inputs required to grow and harvest certain crops. One example of precision agriculture is autonomous tractors, shown in Figure 5.

Figure 5. An autonomous tractor [6].

In order to support all the applications in different industries, the GNSS solution must be accurate and reliable. This thesis focuses on improving the GNSS solutions in kinematic environments by excluding erroneous satellite measurements and data using the random sample consensus (RANSAC) algorithm.

2. Background

Normally, a GNSS receiver calculates its own position by using the distances from the antenna to multiple satellites. The distance can be estimated with the code phase of GNSS satellites [7], which is based on the phase of a pseudorandom code generated by the satellites. For GPS, it could be C/A (course-acquisition) code, P (precise) code or other types of codes. Error of code phase-based ranging is usually on the order of 1-5 meters [7]. The accuracy is primarily limited by the noise and errors on the code phase measurements.

2.1 Carrier-Based Techniques with Precise Baseline Determination in Real Time

The range between an antenna and a satellite can also be measured with the carrier phase of a GNSS signal. As illustrated in Figure 6, when the carrier signal from a GNSS satellite is received, its frequency and phase can be precisely measured. The magnitude of noise and error on GNSS carrier phase is much lower than that of code phase. Therefore, carrier phase-based ranging is often used in applications to achieve better precision.

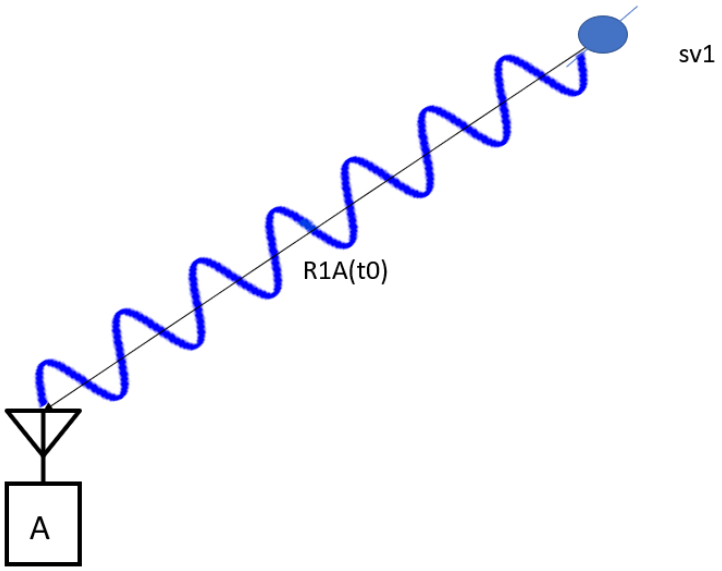

Figure 6. Diagram of an introduction to an alternative method [8].

For example, precise positioning with GNSS can be achieved by using the real-time kinematic (RTK) solution. As the name implies, RTK is a solution that finds the location of a rover (user) in real time with respect to a base. This solution uses carrier phase-based range measurements to calculate a vector pointing from the base to the rover. RTK, though computationally complex, offers a relative positioning solution that is a few orders of magnitude better than that of stand-alone GNSS position solutions. The RTK base station typically has a GNSS antenna installed on a fixed location, whereas another GNSS antenna would be installed on the rover vehicle that is free to move within the proximity of the base. In some cases, the rover can be stationary as well.

RTK achieves its accuracy via a differencing technique, Double-Differencing Carrier Phase. Differencing is a type of technique often used in GNSS to reduce errors, which ranges from single to triple differencing [9]. A single-differenced solution can be formed by differencing two GPS receivers with respect to one satellite or differencing two satellites with respect to one receiver. If two GNSS receivers are differenced, the errors common to the same satellite will be canceled, such as the satellite clock/orbital errors. Likewise, two satellites are differenced to cancel common receiver errors, such as the receiver clock errors. Therefore, a double-differenced solution can further improve the accuracy by removing most of the error sources common to satellites and receivers. Triple differencing refers to the process of a further differencing step over time. Any residual error source that is constant over time will be cancelled in a triple-differenced solution. However, it measures the velocity instead of position of the rover, which makes it less popular than the double-differenced solution. A diagram of the double-differenced solution is shown in Figure 7.

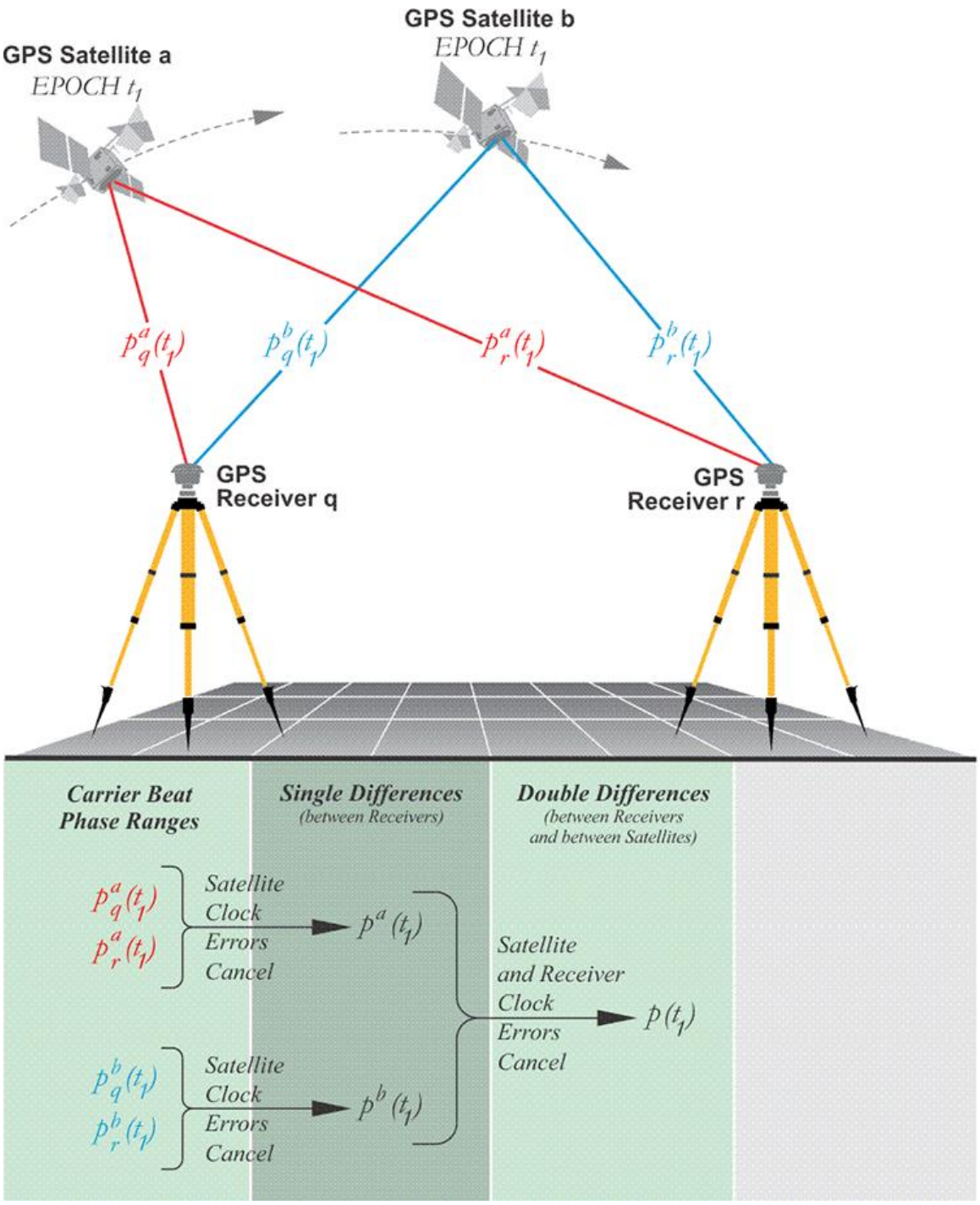

Figure 7. Double difference carrier phase solution [10].

Assume that the GNSS signal is received by the antenna at a known constant frequency, which corresponds to a constant wavelength. Therefore, the true range would contain some number of integer cycles and a fractional part of a cycle. Let N represent the integer number of cycles, and $frac$ be the fractional value. If both N and $frac$ values are known with the length of the cycle, the distance may be precisely known. In practice, the range measured with carrier phase can achieve millimeter-level accuracy.

As a result, a relative position solution between the base and the rover using double differenced carrier phase solution can also achieve millimeter-level precision. Double differenced carrier phase solutions can be used in real time and in post-processing. As shown in Figure 8, the range between a satellite (sv_1) and a GNSS receiver (A) is measured at time stamp t_0, ($R_{1A}(t_0)$).

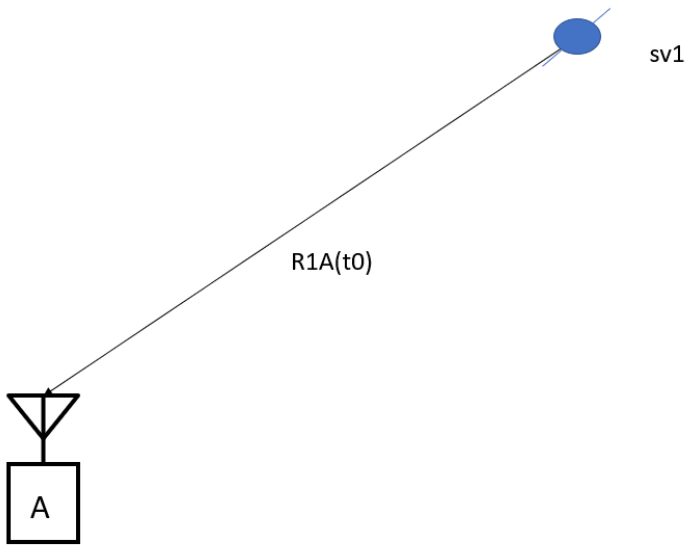

Figure 8. Diagram of one satellite and one receiver [8].

In essence, the GNSS carrier phase measurement is based on the Doppler effect due to the relative motion between the receiver and the satellite. The GNSS signal is transmitted at a nominal frequency. However, when it is received by the receiver antenna, the signal frequency and wavelength would be varied due to satellite and receiver motion. As the receiver antenna and the satellite move closer to each other along their line of sight (LOS), a positive Doppler frequency shift will be observed. If they are moving away from each other, there will be a negative Doppler frequency shift. However, the carrier phase measurement is more than just the frequency shift measured at any instant. Instead, a receiver would accumulate the frequency shift over time to precisely estimate the change of range. Therefore, a receiver can precisely determine $frac$, but the absolute value of N could not be directly estimated in the receiver.

A double-differenced solution requires two satellites and two GNSS receivers as shown in Figure 9. It is assumed that signals from both satellites were transmitted at the same frequency in this configuration.

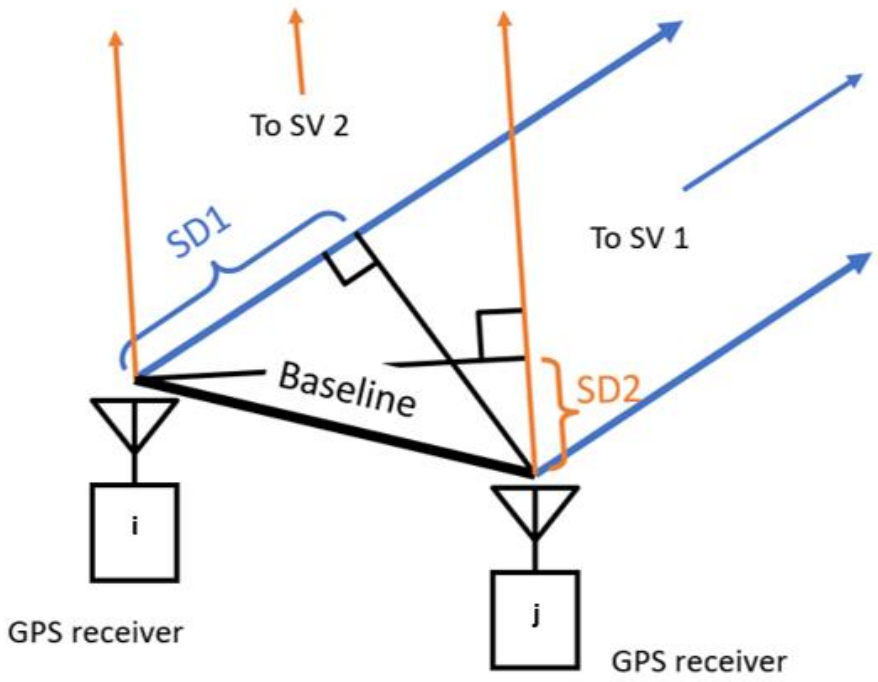

Figure 9. A diagram of the basis required for a double difference carrier phase solution [8].

First, two single difference solutions will be calculated using Equations 1 and 2 where the phase differences, $\phi_{i,1} - \phi_{i,2}$ and $\phi_{j,1} - \phi_{j,2}$ are the difference in the $frac$ values from both satellites, measured by each of the two receivers. Single differencing between two satellites removes the errors common to a receiver. The integer differences between both satellites, N_i and N_j, are unknown to either receiver at this stage. The difference in satellite clock errors, $c\delta t_{12}$, is also included in both equations. However, since it is common to both receivers, it would be eliminated in the next step, a double differenced solution.

$$SD_i = b \cdot e_i = \phi_{i,1} - \phi_{i,2} + N_i\lambda + c\delta t_{12} \quad (1)$$

$$SD_j = b \cdot e_j = \phi_{j,1} - \phi_{j,2} + N_j\lambda + c\delta t_{12} \quad (2)$$

The double differenced carrier phase is calculated simply with Equation 3, which is just the difference of the two single differenced carrier phases shown above.

$$\begin{aligned} DD_{ij} &= SD_i - SD_j \\ &= \phi_{i,1} - \phi_{i,2} - \left(\phi_{j,1} - \phi_{j,2}\right) + N_{ij}\lambda \end{aligned} \quad (3)$$

It only includes the measured phase values, a known wavelength λ and an integer $N_{ij} = N_i - N_j$. Since both N_i and N_j are still unknown, N_{ij} is also treated as an unknown integer in this equation. It is called the "ambiguity" in a double differenced solution. DD_{ij} is related to the relative position between the receivers, and between the receivers and the satellites. As shown in Equation 4, it is the dot product of the baseline between both receivers, denoted with vector b, and the difference between two-unit vectors (e_i and e_j).

$$DD_{ij} = b \cdot \left(e_i - e_j\right) \quad (4)$$

Both unit vectors e_i and e_j can be calculated using rough estimates of user location, or from an iterative solution. Therefore, a successful double differenced solution will produce estimates for the baseline b and ambiguity N_{ij}. Equation 4 can be formulated with multiple satellites. However, each equation can only reduce the degree of freedom by one. Therefore, it would be impossible to directly solve for b and ambiguity N_{ij} from all the satellites.

It can be formulated with satellites that transmit more than one frequency as well. For example, Equations 5 and 6 show the formation of equations for GPS L1 and L2 frequencies.

$$(e_2{}^T - e_1{}^T)(x_{b,real} - x_a) + N_{L1}\lambda_1 = \phi_{L12A,meas} - \phi_{L12B,meas} - (\phi_{L11A,meas} - \phi_{L11B,meas}) \quad (5)$$

$$(e_2{}^T - e_1{}^T)(x_{b,real} - x_a) + N_{L2}\lambda_2 = \phi_{L22A,meas} - \phi_{L22B,meas} - (\phi_{L21A,meas} - \phi_{L21B,meas}) \quad (6)$$

Notice that each frequency has its unique ambiguity. Therefore, the use of multiple frequencies does not provide a direct solution of the ambiguities either.

2.2 Ambiguity Solution

The key to providing high-precision and speedy GNSS parameter estimations is held within carrier-cycle ambiguity resolution, using double differencing techniques. One method includes examining the residuals [9] after calculating the least-squares solution at each iteration of the double difference through the range of carrier wavelengths. By going epoch-to-epoch, the integer ambiguity sets that have residuals close to zero would be saved as potential candidates for the double difference iteration. Notice that due to noise, the residuals would not be exactly zero even for the correct integer ambiguity set. Over time, the number of potential candidate sets would decrease until only one remained. However, this method is not recommended as the number of least-square solutions that would have to be iterated for each epoch would become quickly unmanageable and is, overall, inefficient.

Another approach uses predetermined criteria and only examines the data that falls within these criteria. There are three main techniques that follow this method including the fast ambiguity resolution approach (FARA), the ambiguity function method (AFM), and the least-squares ambiguity search technique (LSAST) [9]. All three of these methods provide further applications when attempting to find a GNSS solution in real time or in dynamic situations.

In most modern solutions, a method called the Least-Squares Ambiguity Decorrelation Adjustment (LAMBDA) is often used. As to be shown in this section, the ambiguity can be solved as an Integer Least-Squares problem. The LAMBDA method uses a two-step process. First, it transforms the ambiguity to decorrelate the integer set. This reshapes the search space into a new search domain [9]. This new domain is intended to have an integer nature and to preserve volume.

To understand this, assume that the ambiguity search space is two-dimensional in the shape of an ellipse. The LAMBDA method would then transform the search space into a circular shape, which is much easier for calculations.

In the second step, LAMBDA will search for candidate solution.

The exact implementation of double differencing in each individual software varies based on the user and the desired result. Thus, depending on the target application, different parts of each technique (FARA, AFM, LSAST, LAMBDA, etc.) may be used. However, most techniques do include LAMBDA, since it is more likely to contain the desired solution after ambiguity transformation [9].

Methods involving QR factorization can eliminate the need to calculate the least-squares solution. Instead, the least-squares solution space is able to be separated from the least-squares residual, in which the residuals may be used to isolate the integer ambiguities by investigating the double difference carrier-phase, DD_{cp}, measurement using Equation 7 below where ϕ_{DD} is the double difference fractional phase, which is taken from the receiver measurements, $\hat{n}$ is the double difference ambiguity, which is unknown, R_b is both the inherent receiver channel bias and the residual propagation delays, S is the noise, and λ is a conversion factor based on wavelength [9]. This equation takes the double difference and outputs a result in units of length. Multipath, though technically not noise, is included in the variable for noise, S, as it is difficult to separate from other sources of noise.

$$DD_{cp} = (\phi_{DD} + \hat{n} + R_b + S)\lambda \tag{7}$$

Equation 7 may be rewritten using smoothed-code double differences, DD_s, as shown below in Equation 8 [9]. Now, instead of using the double difference fractional phase, which is taken from the receiver measurements, the geometric distance (measured in carrier-cycles) is used, ρ_{DD}, within the noise bound. Also, the unknown double difference ambiguity is replaced by an integer ambiguity that is known to be bounded between -11 and +11 wavelengths at L1. This adjustment in equations may be used with the knowledge that the ambiguity or uncertainty of the values are bounded.

$$DD_s = (\rho_{DD} + \tilde{n} + R_b + S)\lambda \tag{8}$$

Using the least-squares solution, the resolution of the unknown, but bounded double difference ambiguity, $\tilde{n}$, can now be determined by Equation 9 below [9]. In this equation, the $\tilde{n}$ represents an applicable double difference wavelength ambiguity number. q is the least-squares residual vector. The elements of q do not change until further measurements are gathered, which means that q remains constant within an epoch. The value η is the measurement inconsistency. In a perfect world where there is no noise, this value would be zero.

$$\begin{aligned}\mathbf{qDD}_s = \big[&q_1\big(\rho_{DD_1} + \tilde{n}_1 + R_{b_1} + S_1\big) + q_2\big(\rho_{DD_2} + \tilde{n}_2 + R_{b_2} + S_2\big) \\ &+ q_3\big(\rho_{DD_3} + \tilde{n}_3 + R_{b_3} + S_3\big) + q_4\big(\rho_{DD_4} + \tilde{n}_4 + R_{b_4} + S_4\big)\big]\lambda = \eta\end{aligned} \tag{9}$$

By identifying the unknown carrier-cycle integer ambiguity from the smoothed-code double differences, the only sources of error that remain are noise and multipath [9]. As discussed above, multipath, though not noise, acts as noise over an extended period of time. One difference between noise and multipath is that multipath may not tend to a zero-value mean. With these thoughts, Equation 9 may be rewritten as Equation 10.

$$q_1[DD_{s1} - \tilde{n}_1\lambda] + q_2[DD_{s2} - \tilde{n}_2\lambda] + q_3[DD_{s3} - \tilde{n}_3\lambda] + q_4[DD_{s4} - \tilde{n}_4\lambda] = \gamma \qquad (10)$$

The value for the double difference ambiguity can be brought close to zero over some prearranged boundary or threshold constraint, γ. The iterations to solve for the possible candidates of the fixed baseline solution are not unreasonable as the computation cost becomes manageable at this stage.

There are a few different ways of extracting the fixed solution in practice. As mentioned previously, a receiver can choose a small set of candidates and observe it over a period of time. It is also possible to observe the difference between the best candidate and the second best [9].

2.3 Wide-Lane Solution

There are more efficient methods, however, by examining the wide-lane solution, to further reduce the computation cost [9].

The wide-lane frequency and wavelength are defined in Equations 11 and 12, respectively. The frequency is identified as the beat frequency between L1 and L2 [9].

$$f_{wl} = 1{,}575.42\ \text{MHz} - 1{,}227.6\ \text{MHz} = 347.82\ \text{MHz} \tag{11}$$

$$\lambda_{wl} = 86.25\ \text{cm} \tag{12}$$

The wide-lane wavelength is much greater than L1, which results in a multi-magnitude decrease in the number of potential candidate residuals. Using the wide-lane wavelength, the search space could be further limited to an even smaller number of λ_{wl}. However, the tradeoff for this efficiency in analysis comes in the form of increased noise, which is measured using Equation 13. On the bright side, current receiver technology is well equipped to handle this increase in noise.

$$S_{wl} = \lambda_{wl}\sqrt{\left(\frac{S_{L1}}{\lambda_1}\right)^2 + \left(\frac{S_{L2}}{\lambda_2}\right)^2} \tag{13}$$

The wide-lane carrier phase may be calculated by the simple equation shown in Equation 14 [9].

$$\phi_{wl} = \phi_{L1} - \phi_{L2} \tag{14}$$

This is based on dual-frequency tracking (collecting data from both L1 and L2 simultaneously) and requires the use of the P(Y) code in a GPS receiver. Thus, a regular C/A code-based GPS receiver cannot be used for wide-lane solution. There are additional benefits from the P(Y) code. For example, the C/A code also has a much lower chipping rate than the P(Y) code, which results in greater noise and errors. With L2 and P(Y) code, the ionospheric path delay may be eliminated, or at the very least, determined [9]. By combining the difference of L1 and L2 signals, the wide-lane metric is determined. The opposite is also true. By combining the sum of L1 and L2 signals, the narrow-lane metric is determined. As for both metrics, errors that are independent of frequency are unchanged. These errors include ephemeris, troposphere, and clock errors. However, errors that are dependent on frequency are not the same. This means that both narrow-lane pseudo-range observables and wide-lane carrier phase observables must be paired together to understand the effects of frequency dependent variables.

2.4. Software Implementation

RTKLIB [11] is a GNSS software toolkit to perform precise positioning and capable of processing recorded data from most GNSS constellations, including GPS, GLONASS, Galileo, and BeiDou [12]. This open-source software uses an extended Kalman filter (EKF) to compute the final solution in differential GNSS (DGNSS) in three different modes: static, kinematic, and moving-baseline [13].

The EKF may be modelled with the following three equations (Equations 15, 16, and 17), which estimate the solution as a state vector, x, and its covariance matrix, P [13]. $\hat{x}_k$ is the estimated state vector, and P_k is the estimated covariance matrix at a specific time epoch, k. These solutions may be estimated using y_k, the linear or linearized measurement model, H, and R_k, the covariance matrix of measurement errors. The $(+)$ and $(-)$ values represent the point in time, with $(+)$ referring to the next instant in time and $(-)$ referring to the previous instant in time.

$$\hat{x}_k(+) = \hat{x}_k(-) + K_k(y_k - h(\hat{x}_k(-))) \tag{15}$$

$$P_k(+) = (I - K_k H(\hat{x}_k(-)))P_k(-) \tag{16}$$

$$K_k = P_k(-)H(\hat{x}_k(-))(H\big(\hat{x}_k(-)\big)P_k(-)H\big(\hat{x}_k(-)\big)^T + R_k)^{-1} \tag{17}$$

Equations 18 and 19 are used to propagate the state vector and the covariance matrix over time respectively. ${F_k}^{k+1}$ is the transition matrix and ${Q_k}^{k+1}$ is the covariance matrix representing the system noise.

$$\hat{x}_{k+1}(-) = {F_k}^{k+1}\hat{x}_k(+) \tag{18}$$

$$P_{k+1}(-) = {F_k}^{k+1}P_k(+){F_k}^{k+1T} + {Q_k}^{k+1} \tag{19}$$

By using double difference measurement models over a short distance, a myriad of error sources can be eliminated. For example, biases from both the satellite and receiver clocks as well as the effects of the ionosphere and the troposphere when the distance between the base-station and the rover are less than 10 km from each other. This is considered a short-baseline solution [13].

In practice, RTKLIB tracks the single difference carrier-phase biases rather than the double difference as described in this section, although the double differenced solution is computed by the software. It is a feature of RTKLIB designed to avoid the "hand-over" of different reference satellites [13]. One additional benefit of using the single difference biases is that they also help solve integer ambiguities that arise from GLONASS FDMA (frequency division multiple access) signals.

Using EKF for short baselines, the state vector includes the position of the rover antenna, the bias of velocity and the single difference carrier-phase. Equation 20 shows a non-linear measurement model, $h(x)$, which includes the measurement models of code phase and carrier phase at L1, L2, and L5 frequencies. Equation 21 shows the linearized version of it, $H(x)$, based on partial derivatives. Equation 22 provides the covariance matrix of measurement errors or R.

$$h(x) = \left({h_{\phi,1}}^T, {h_{\phi,2}}^T, {h_{\phi,5}}^T, {h_{P,1}}^T, {h_{P,2}}^T, {h_{P,5}}^T\right)^T \tag{20}$$

$$H(x) = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}} = \begin{pmatrix} -DE & 0 & \lambda_1 D & 0 & 0 \\ -DE & 0 & 0 & \lambda_2 D & 0 \\ -DE & 0 & 0 & 0 & \lambda_5 D \\ -DE & 0 & 0 & 0 & 0 \\ -DE & 0 & 0 & 0 & 0 \\ -DE & 0 & 0 & 0 & 0 \end{pmatrix} \tag{21}$$

$$R = \begin{pmatrix} DR_{\phi,1}D^T & & & & & & \\ & DR_{\phi,2}D^T & & & & & \\ & & DR_{\phi,5}D^T & & & & \\ & & & DR_{P,1}D^T & & & \\ & & & & DR_{P,2}D^T & & \\ & & & & & DR_{P,5}D^T & \end{pmatrix} \quad (22)$$

The values of $h_{\phi,i}$ (carrier phase) and $h_{P,i}$ (code phase) are shown in Equations 23 and 24, respectively.

$$h_{\phi,i} = \begin{pmatrix} \rho_{\gamma B}^{12} + \lambda_i\left(B_{\gamma b}^1 - B_{\gamma b}^2\right) \\ \rho_{\gamma B}^{13} + \lambda_i\left(B_{\gamma b}^1 - B_{\gamma b}^3\right) \\ M \\ \rho_{\gamma B}^{1m} + \lambda_i\left(B_{\gamma b}^1 - B_{\gamma b}^m\right) \end{pmatrix} \quad (23)$$

$$h_{P,i} = \begin{pmatrix} \rho_{\gamma B}^{12} \\ \rho_{\gamma B}^{13} \\ M \\ \rho_{\gamma B}^{1m} \end{pmatrix} \quad (24)$$

In these equations, D is the single difference matrix, defined in Equation 25. L is the phase-range measurement error in meters and M is the value found from the carrier phase and the code phase in Equations 23 and 24. Error is also found in the matrix of partial derivatives and is defined in Equation 26.

$$D = \begin{pmatrix} 1 & -1 & 0 & L & 0 \\ 1 & 0 & -1 & L & 0 \\ M & M & M & O & M \\ 1 & 0 & 0 & L & -1 \end{pmatrix} \tag{25}$$

$$E = (e_r^1, e_r^2, \dots, e_r^m)^T \tag{26}$$

The transition matrix and the covariance matrix representing the system noise are defined in Equations 27 and 28, respectively. They are used to propagate the filter over time.

$$F_k^{k+1} = \begin{pmatrix} I_{3\times3} & I_{3\times3}\tau_\gamma & \\ & I_{3\times3} & \\ & & I_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{27}$$

$$Q_k^{k+1} = \begin{pmatrix} 0_{3\times3} & & \\ & Q_v & \\ & & 0_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{28}$$

Notice that in this model, it is assumed that the rover may be moving, and the system noise in velocity (east, north, and up) is considered in both matrices. The filter can be constructed without considering the receiver dynamics. In that case, matrices defined in Equations 29 and 30 will be used instead. At every epoch, the receiver position is reset to the initial values to avoid numerical instability. Large process noises are simply added to the variance that RTKLIB handles. The initial receiver position is defined by the single point positioning process previously used. This allows avoidance of various non-linear iterations from the measurement updates.

$$F_k^{k+1} = \begin{pmatrix} I_{3\times3} & & \\ & I_{3\times3} & \\ & & I_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{29}$$

$$Q_k^{k+1} = \begin{pmatrix} \infty_{3\times3} & & \\ & 0_{3\times3} & \\ & & 0_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{30}$$

When the rover is stationary, Equations 31 and 32 will be used in RTKLIB to propagate the filter. In the static mode, the rover velocity noise is simply set to zero. The data collected in this thesis will be processed in this mode.

$$F_k^{k+1} = \begin{pmatrix} I_{3\times3} & & \\ & I_{3\times3} & \\ & & I_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{31}$$

$$Q_k^{k+1} = \begin{pmatrix} 0_{3\times3} & & \\ & 0_{3\times3} & \\ & & 0_{(3m-3)\times(3m-3)} \end{pmatrix} \tag{32}$$

Furthermore, the best candidate of integer ambiguities is compared to the second best at every epoch in RTKLIB. The result is reported as a ratio between their residuals, which allows for a direct observation of the solution quality.

2.5 Ongoing Research on RTK

Current research of GNSS RTK focuses on three main areas: improving the measurement accuracy and error model, increasing the efficiency of the solution, and finding new applications of RTK. All these areas are tied together in their efforts to minimize the computational cost and initialization time.

There have been many studies focused on improving the accuracy of the solution [14], [15], [16]. Two researchers discovered ways to reduce the temporal and spatial decorrelation errors [14]. They wanted to minimize the errors produced as the system is changing both environmental factors over a distance with respect to time. Further research has been conducted to minimize the inter-channel biases [15], which results in fewer iterations to solve for the ambiguities [16]. Huge improvements have been made in the last ten years alone, but there is still further to go in this area [17].

Other areas of research in more recent years have included increasing the efficiency of the solution [18], [19]. As more satellites have become available, another area of RTK solution research focuses on increasing the availability of RTK solution [18]. This is also done by minimizing the iterations required to solve for the ambiguity of the solution. With greater availability of GNSS satellites, current research also focuses on achieving a better solution under varying conditions [19].

More areas of research include applying the solution to new technologies [20], [21]. With more options and a better understanding of how to solve for position, there is now research aimed at improving the efficiency of GPS modules for low-cost scientific use [20]. Others have researched ways in which to increase performance with low-cost systems [21].

However, despite all this research into improving the solution, the ambiguity resolution still maintains a low success rate in a GNSS-challenged environment, such as urban canyons, which results in a lower reliability of the positioning. It is an even greater challenge to achieve the goal with low-cost GNSS receivers and antennas.

3. Methodology

The research project focused on improving the GNSS RTK solution using low-cost receivers and antennas. The proposed method is used to pre-filter carrier phase data measurements based on the inherent consistency among GNSS satellites. This work is based on an existing project that studied the consistency of carrier phase among GNSS satellites [22]. The improved RTK solution will be demonstrated with recorded live GNSS data.

3.1 Processing Software

A few preliminary trials were conducted using data recorded with the UBLOX F9P board and the RTKLIB software. An example of the original standalone solution (without using RTK) of a stationary rover is shown in Figure 10. The figure shows meter-level accuracy of the standalone positioning solution.

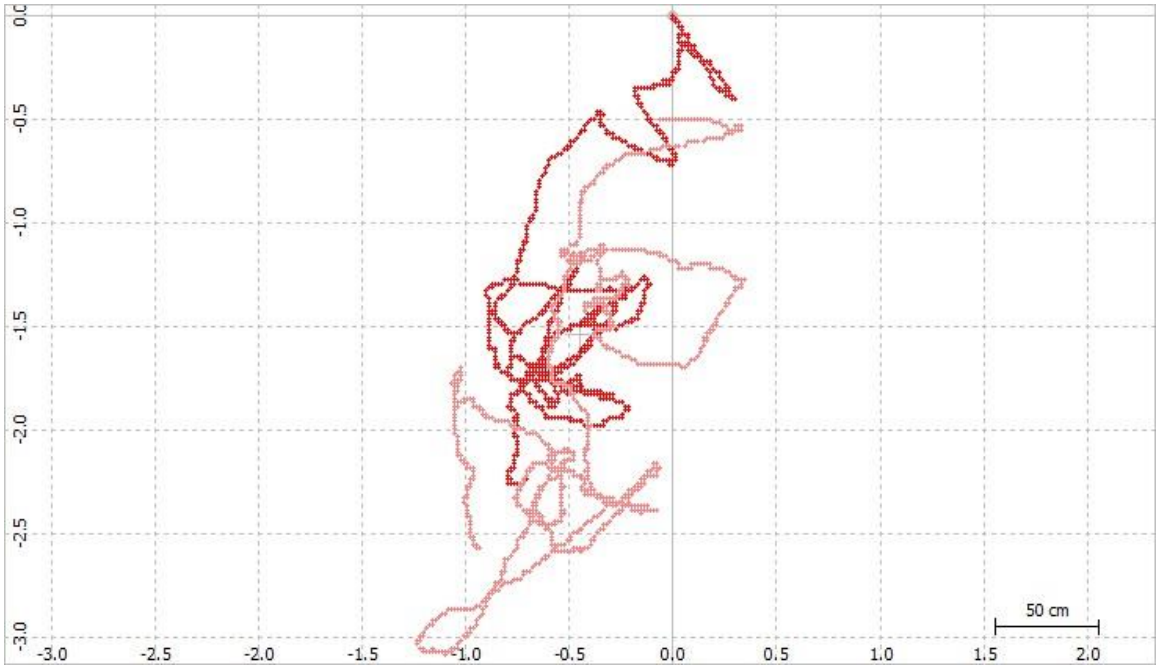

Figure 10. The standalone positioning solution of a rover calculated in the RTKLIB software. (Courtesy of Nicholas Hill.)

The position can be further corrected using post-processing RTK from RTKLIB. An example of the corrected solution is shown in Figure 11. The yellow part of the plot shows the solution before the integer ambiguity is solved, and the green part for after the ambiguity solution. The solution is at cm-level accuracy after the integer ambiguity solution.

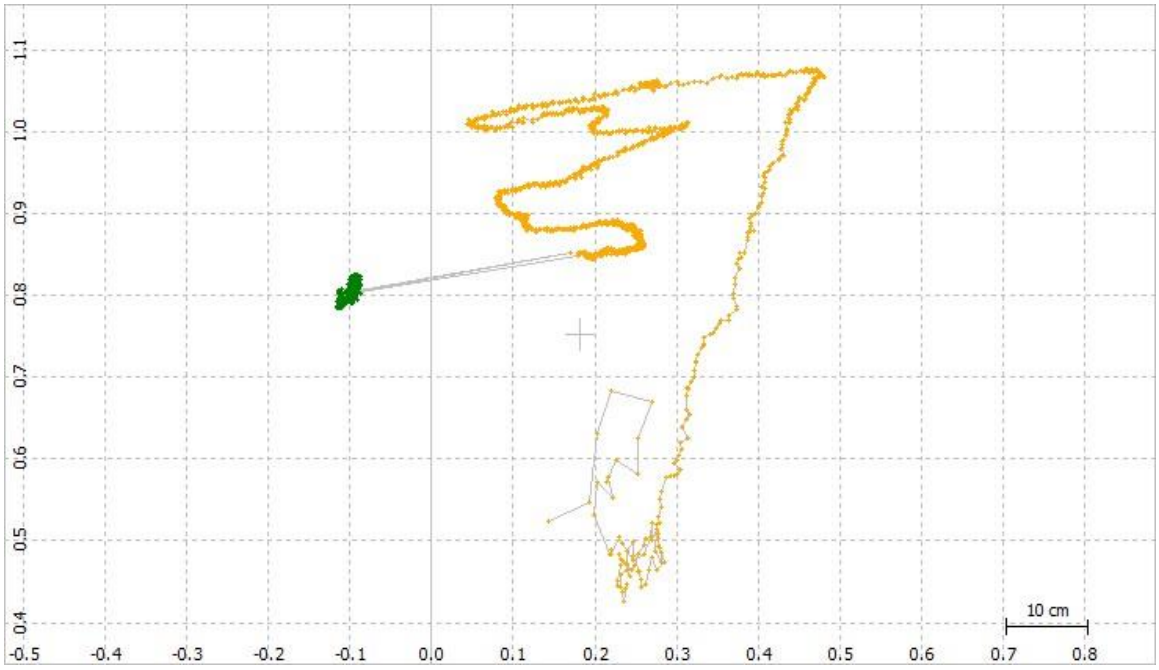

Figure 11. The position solution from a successful RTK trial using the RTKLIB software. (Courtesy of Nicholas Hill.)

In RTKLIB, satellites from these constellations will be screened based on preset thresholds of signal strength and elevation angle, which is a common practice in GNSS receivers. However, signal strength and satellite elevation angle do not always directly reflect the quality of carrier phase measurements. In a complex operational environment such as an urban area, signal multipath, shadowing, interference, jamming, and spoofing may affect the carrier phase measurements. It is desirable to directly observe the quality of carrier phase measurements in this case. However, such direct observation typically requires a truth reference, such as a high-quality navigation device, which is not available to small vehicles during normal operations. An alternative approach is to measure how well the carrier phase measurements agree with each other. Instead of a truth reference, each measurement is compared against a consensus among all available measurements. This approach leads to a new satellite pre-selection method for RTK, based on random sample consensus (RANSAC).

RANSAC was originally developed with an iterative method to discover an underlying math model from noisy samples, despite any data outliers [23]. These outliers should have little influence over the estimate values in data pre-selected with RANSAC. It has been widely used in computer vision and robotics. Figure 12 shows an example line of best fit that can be generated using a RANSAC algorithm to exclude the outliers.

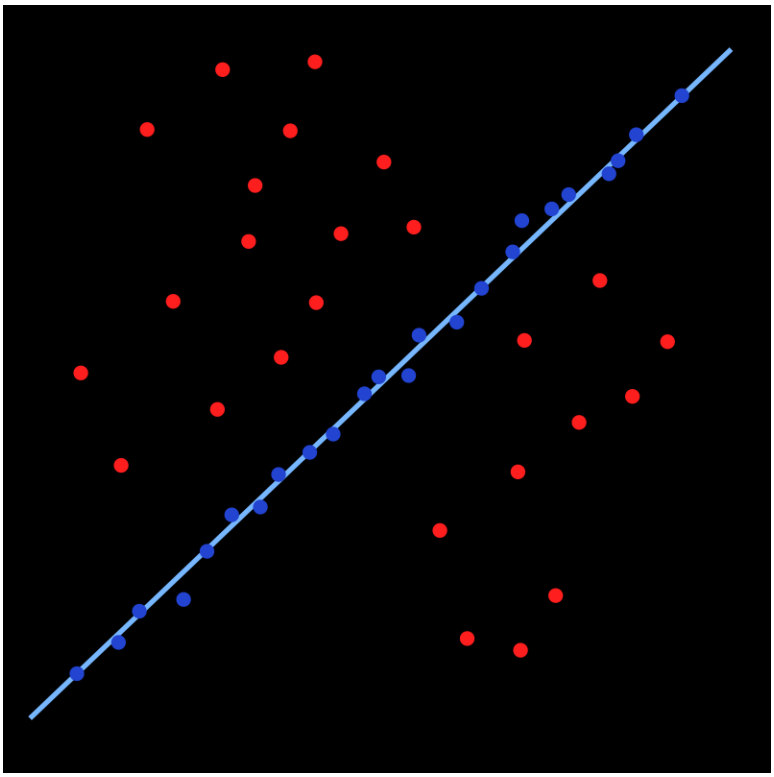

Figure 12. A line of best fit produced with a RANSAC algorithm [24].

RANSAC is used to identify the *outliers* in a set of raw measurements. In the context of RTK, it is used to flag carrier phase measurements from erroneous satellites. These satellites will then be excluded from RTK in the pre-selection step, which is expected to improve the accuracy and reliability. RANSAC is able to handle multiple concurrent outlier satellites, which the system considers “faulty” information, whereas the inliers are “fault-free” [22].

In a GNSS-challenged environment, such as an urban canyon, it is often hard to precisely model errors and outliers in the raw measurements. RANSAC is not dependent on an accurate error model for GNSS, which makes it robust in this type of environment [22]. Instead of individual error models, RANSAC focuses on the lack of consistency among the measurements of faulty satellite data and is able to reliably separate them from good measurements. It is very efficient and can be implemented for post-processing or real-time operations.

In this work, real-time software was developed in C in a Linux operating system, which implements RANSAC on GNSS carrier phase. The software component has been integrated into the pre-selection step of RTKLIB. Currently, it is a component of the customized post-processing RTK function in RTKLIB. The feasibility of RTK-RANSAC integration will be demonstrated using recorded live GNSS data.

3.2 Hardware Setup and Data Recording

The low-cost receiver used in this work is the UBLOX C099-F9P Application Board (Figure 13), with the high-quality NovAtel PwrPak 7 (Figure 14) receivers used as truth references. The UBLOX receiver is compact and low-cost, which makes it a good candidate for target applications such as UAVs and autonomous vehicles. However, both UBLOX and NovAtel receivers have been rated with centimeter-level accuracy in RTK solutions.

Figure 13. The UBLOX C099-F9P Application Board [25].

Figure 14. The NovAtel PwrPak 7 [26].

Live GNSS data will be recorded using two identical setups, each equipped with a UBLOX receiver and a NovAtel receiver. Each receiver is connected with its own antenna. A complete data collection setup is shown in Figure 15. The UBLOX receiver is powered by a 3.7 V lithium-ion battery. The NovAtel receiver is powered by an 11.1 V lithium-ion battery, which also powers a Raspberry Pi III computer that records raw data from the UBLOX receiver. Raw data from the NovAtel receiver is saved on its internal data storage.

Most of the datasets demonstrated in this work are collected in a suburban area. As a proof of concept, the setups have not been installed on any platform (such as UAVs) yet. Instead, both are placed on the ground for stationary tests. Typically, both setups are placed in an open field with no immediate obstructions overhead (such as trees or power lines). There will be houses and trees in the surroundings, such as the environments shown below in Figures 16 and 17. However, the data collection and processing techniques presented in this paper are also applicable to receivers in a kinematic application, and in an urban canyon.

After each data collection trip, raw data from both receivers are transferred onto local computer for storage. A flow chart in Figure 18 summarizes this process.

Figure 15. The complete hardware setup with a UBLOX and a NovAtel receiver.

Figure 16. A common data collection location with no obstructions that might hinder the receivers' view of satellites.

Figure 17. A field used for data collection in Washington, NC.

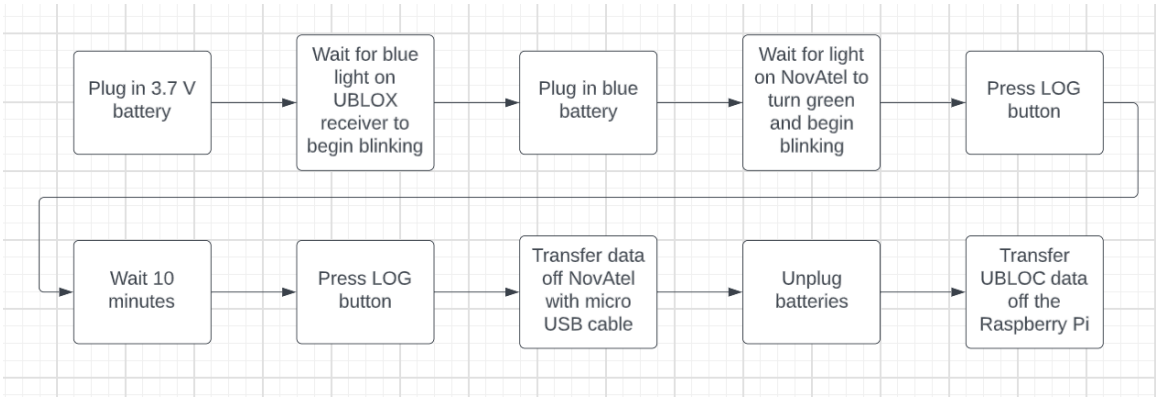

Figure 18. Flowchart of data recording.

3.3 Data Processing

Both RTKLIB and RANSAC have been compiled in Raspbian, which is a light-weighted distribution of Linux customized for a Raspberry Pi computer. An offboard Raspberry Pi III computer is used to process raw data recorded by both UBLOX receivers. Notice that since receivers were stationary in this work, no dynamic truth reference was needed. Therefore, the data recorded with the NovAtel receiver were not processed.

The binary data files from both UBLOX receivers were converted into Receiver Independent Exchange (RINEX) format. RTKLIB provides a function “convbin” for this conversion. Notice that RTKLIB is capable of processing UBLOX data in its native format. However, it is more convenient to use a standard file format such as RINEX, which is independent of the receiver manufacturer in post-processing. Different software components, such as RTKLIB and RANSAC can easily share the same data files. The software developed in this work can also be used with receivers from other manufacturers in the future.

Next, the post-processing RTK function in RTKLIB, “rnx2rtkp” is used to compute the double differenced carrier phase solution. This function has been modified such that it will report the solution with and without RANSAC. The navigation solution and status over the entire data collection are reported in two text-based log files (log.txt and log.txt2 in the current implementation), with and without RANSAC, respectively. The log files are the final products of the Linux-based software developed in this work.

Although some user interaction is still required in the current implementation, it is rather limited and can be easily automated in a future version. An embedded computer such as Raspberry Pi III can easily handle real-time processing of “convbin” and “rnx2rtkp”.

Both log files are copied onto a Windows computer for further analysis in MATLAB. A summary of this process is shown in the flowchart in Figure 19.

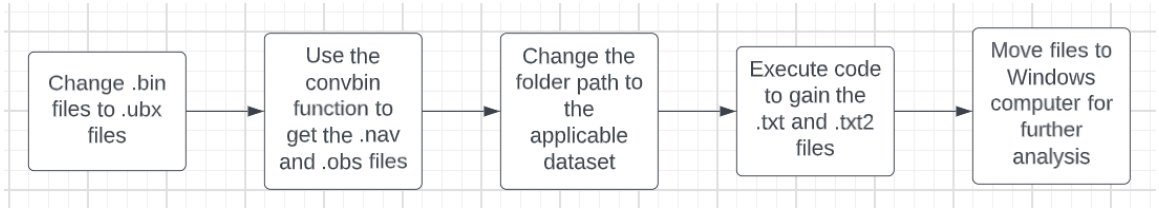

Figure 19. Flowchart of processing data.

Quality and status of the double differenced solution are compared in MATLAB, with and without RANSAC. Obviously, it is not part of the real-time processing. Appendices A, B, C, and D show the code used to analyze this data. Appendix E shows the step-by-step process of how to analyze the data.

4. Data Analysis Results

As aforementioned, the datasets have been collected with partially open sky in a suburban environment. Some of the satellites may be affected by trees and houses in the surroundings, but most satellites would be normal in these datasets. RANSAC is not expected to make a large difference in this type of scenario.

4.1 Improved RTK with RANSAC

One of the datasets (Dataset 2), however, showed a distinct difference in the quality of solution. Without pre-selecting satellites using RANSAC, the double differenced solution in RTKLIB struggled to compute a precise relative position solution between both antennas. The relative position (baseline) in the east, north, and up directions is shown below in Figure 20 without RANSAC.

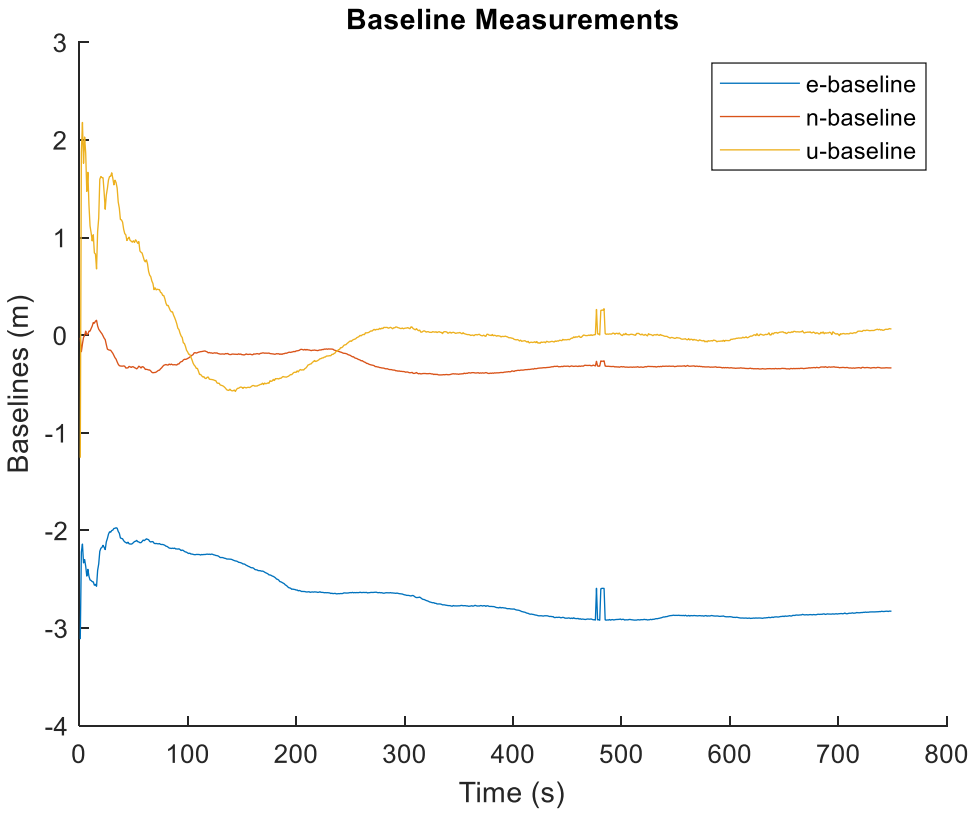

Figure 20. The baseline measurements of Dataset 2 without RANSAC.

The standard deviations of these baseline measurements are shown below in Figure 21.

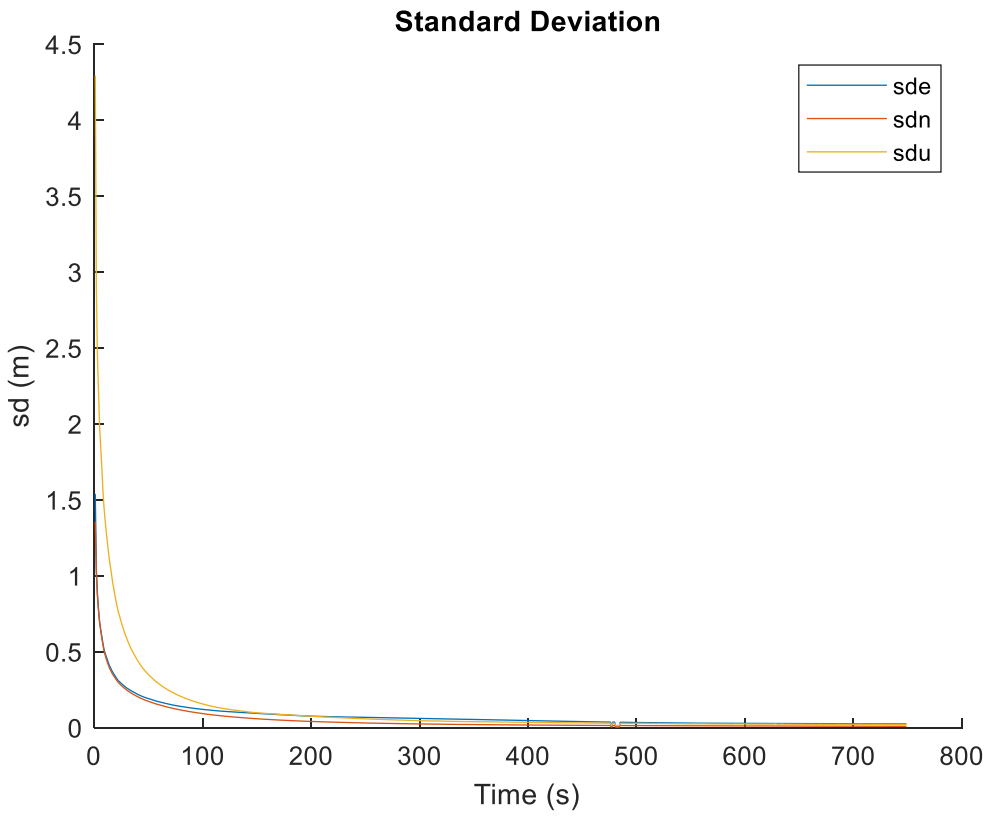

Figure 21. The standard deviation measurements of Dataset 2 without RANSAC.

The quality metric “q” of the solution reported by RTKLIB is shown below in Figure 22. In this dataset, the value of q is either 1 or 2. Status 1 means the system has successfully solved the integer ambiguities with an RTK solution, whereas Status 2 means that an RTK solution is available, but without the integer ambiguity. As shown in Figure 22, other than for a very brief span (5 seconds), the system was stuck in Status 2, which means that the integer ambiguities were not solved. The value “ns” is the number of satellites currently being tracked. On average, about 12 satellites were used in this dataset without RANSAC. The value “ratio” is defined as the ratio between residuals of the second best over the best solution in each epoch. As aforementioned it also reflects the quality of solution. When the integer ambiguities are correctly solved, this ratio is expected to be higher. Therefore, higher ratio value indicates correct ambiguities and better RTK solution. Notice that for the brief 5 seconds of time in Status 1 (using RANSAC), the ratio value peaks, showing a higher quality dataset.

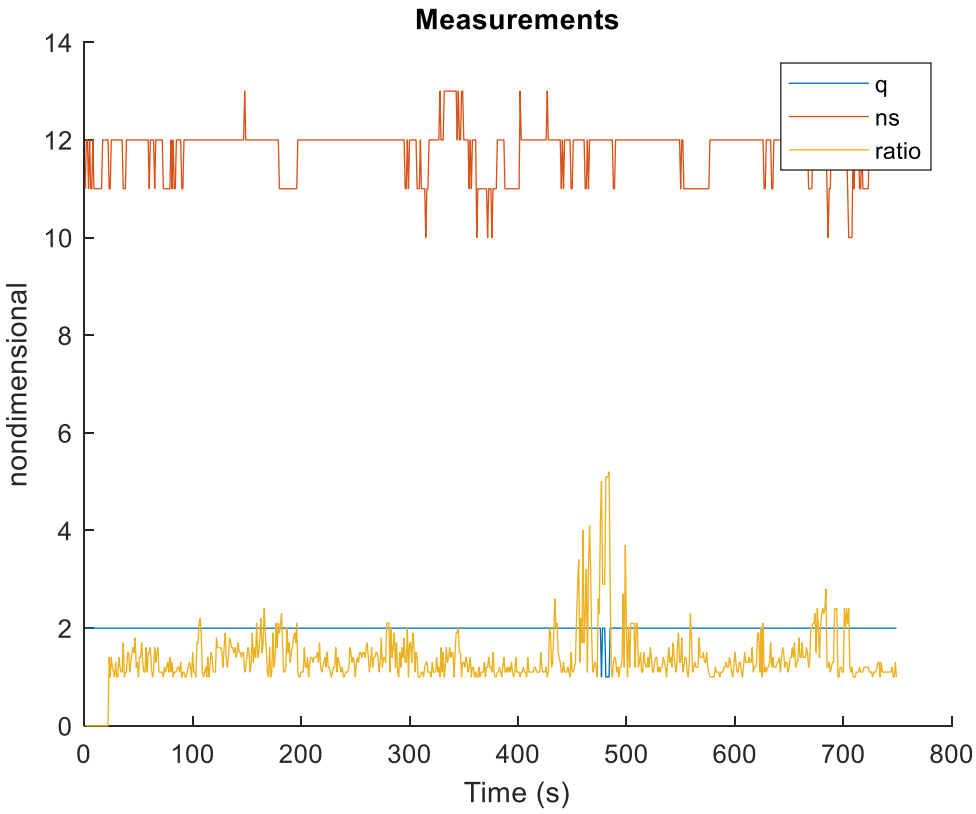

Figure 22. The quality measurements of Dataset 2 without RANSAC.

The actual error in the RTK solution can be observed by comparing the estimated length of baseline against a truth reference (tape-measured). Figure 23 shows that during the five seconds when Status 1 was achieved, the error is at cm-level. With Status 2, the error remains high (around 0.3 m).

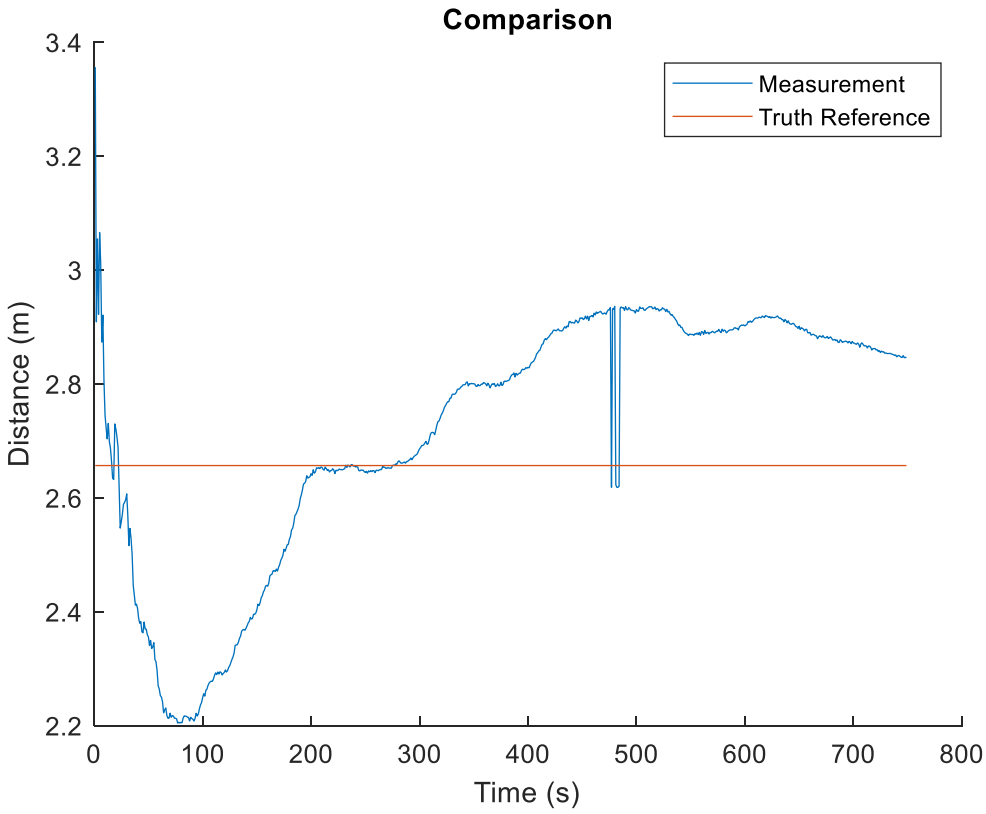

Figure 23. The measurement of distance with a truth reference comparison of Dataset 2 without RANSAC.

This dataset is a good representation of what RANSAC can accomplish to improve the quality of RTK. RANSAC was implemented using only the Rover receiver in this case. Since it was assumed that both receivers are close to each other (short baseline), it would be sufficient to run RANSAC on one of them. The baseline measurements are shown below in Figure 24. Notice that at a time a little before 400 s, the system seems to flatten out and provide a solid baseline value for the three directions. This is when the integer ambiguities were solved correctly and remained unchanged.

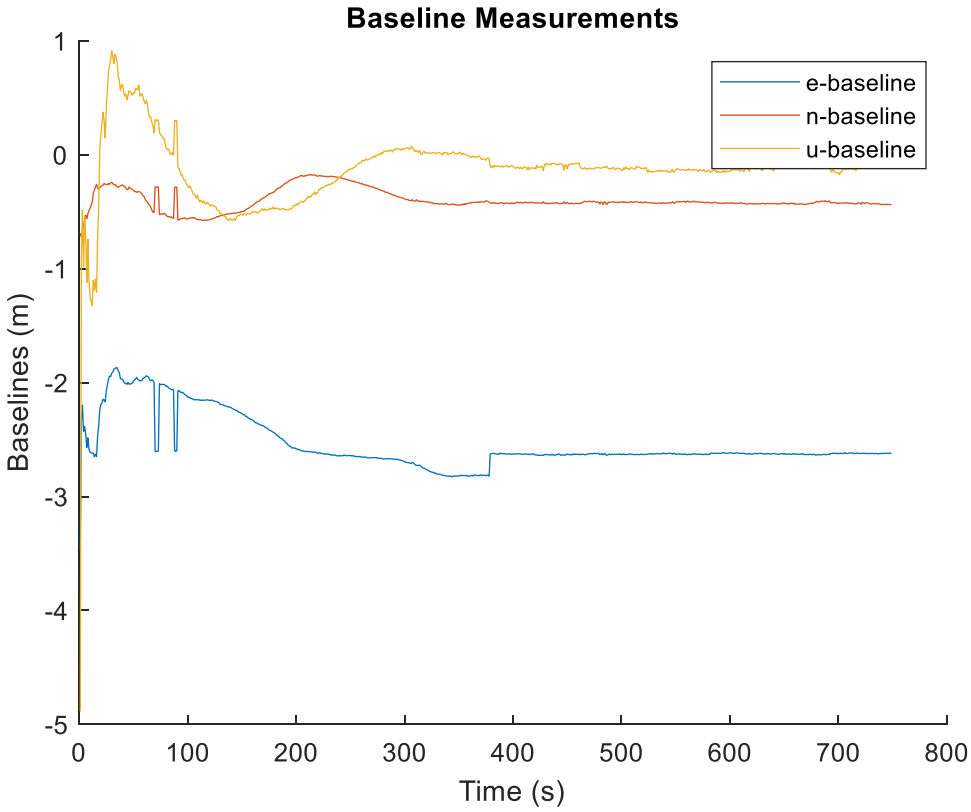

Figure 24. The baseline measurements of Dataset 2 with RANSAC.

The standard deviation values of the baseline measurements with RANSAC are shown below in Figure 25. They are reduced to very low levels after 400 s.

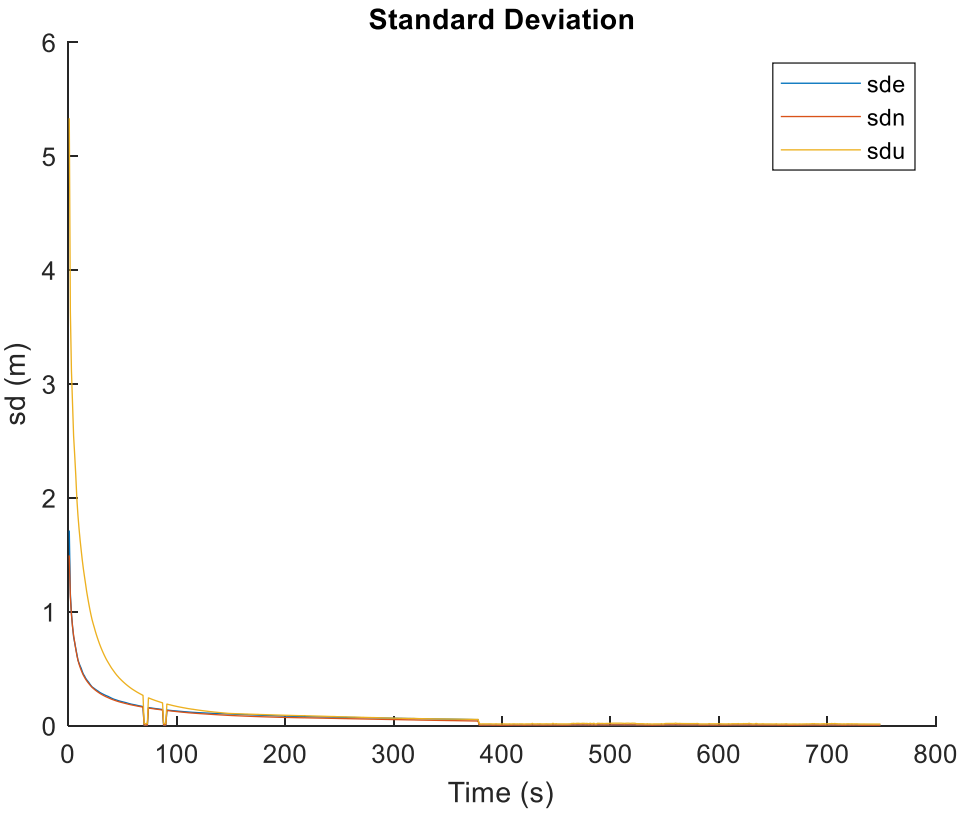

Figure 25. The standard deviation measurements of Dataset 2 with RANSAC.

The quality metrics with RANSAC are shown below in Figure 26. The status (q) bounces between 1 and 2 for a while, but eventually settles into Status 1. Notice that once again, when in Status 1, the ratio value peaks and remains high, showing the highest quality of data. The number of satellites also being used is fewer, as expected, averaging around 10 satellites versus the 12 being seen without RANSAC. This means the RANSAC algorithm correctly identified an erroneous carrier phase that may be present in either or both of the satellites excluded from the solution.

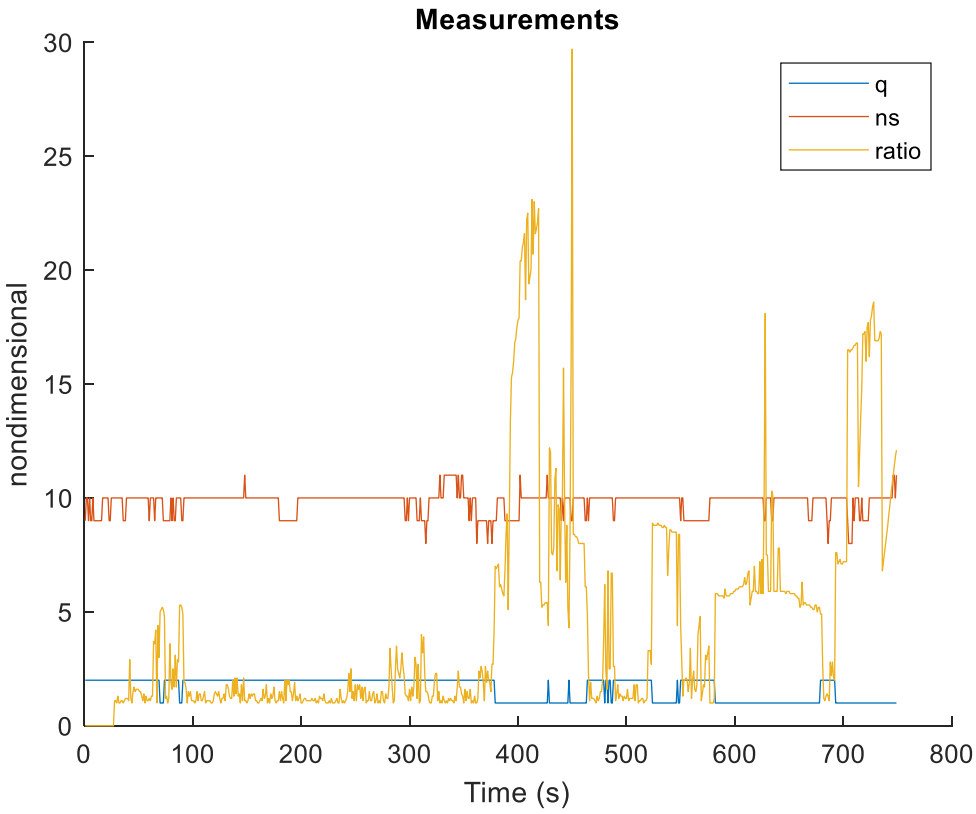

Figure 26. The quality measurements of Dataset 2 with RANSAC.

The comparison of the estimated baseline and the truth reference are shown below in Figure 27. In this case, where RANSAC is present, the measurement is much more accurate, right along with the truth reference line.

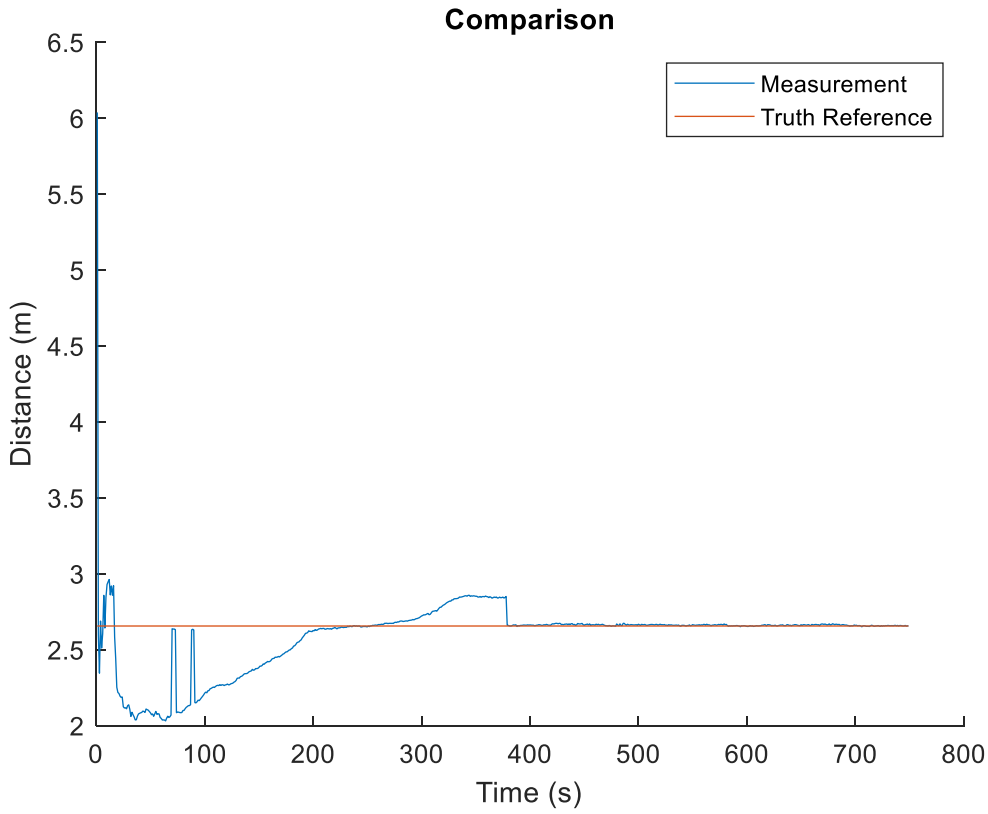

Figure 27. The measurement of distance with a truth reference comparison of Dataset 2 with RANSAC.

The baseline solutions from this dataset with and without RANSAC are compared against each other in east, north, and up directions, respectively. As shown in Figures 28, 29, and 30, the solution with RANSAC remains close to the truth reference in all three directions. Centimeter-level errors were observed with RANSAC, whereas the solution without RANSAC clearly deviates in all three directions.

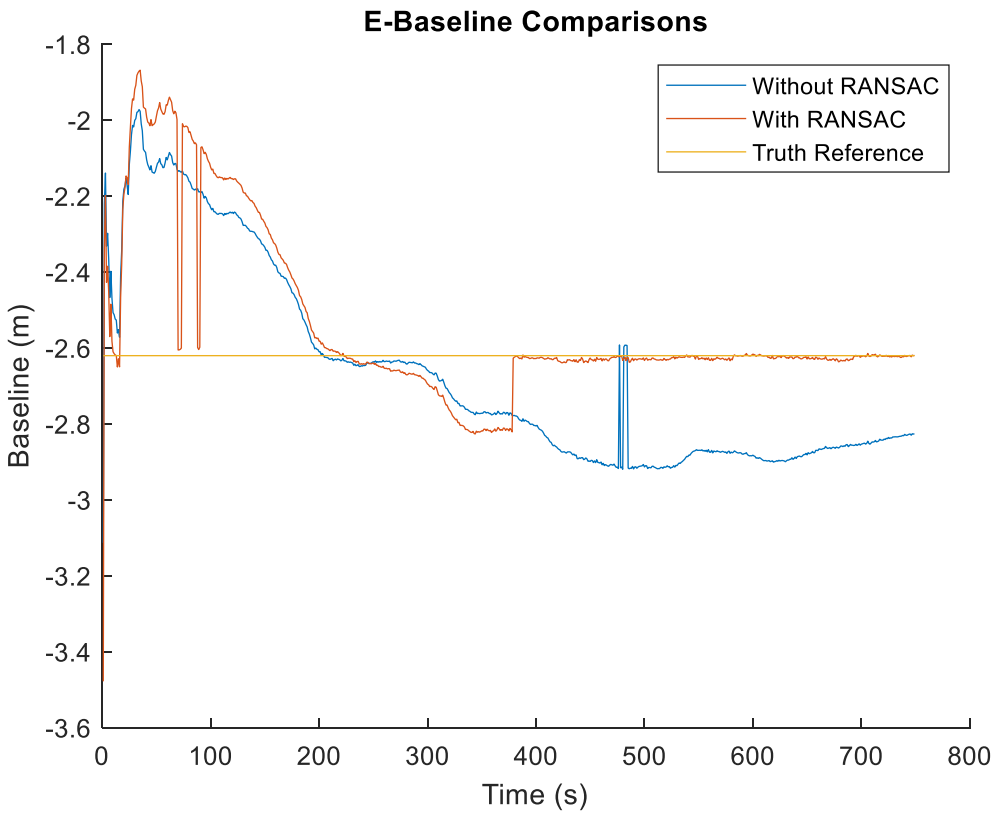

Figure 28. The E-Baseline Comparison of Dataset 2.

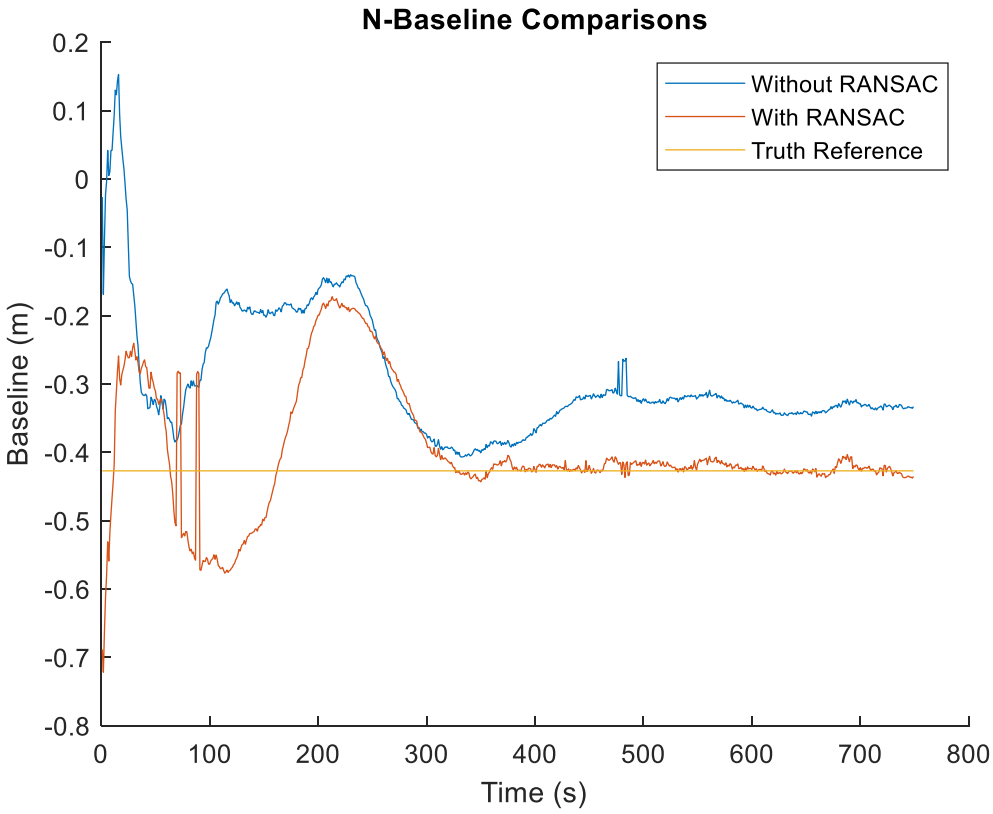

Figure 29. The N-Baseline Comparison of Dataset 2.

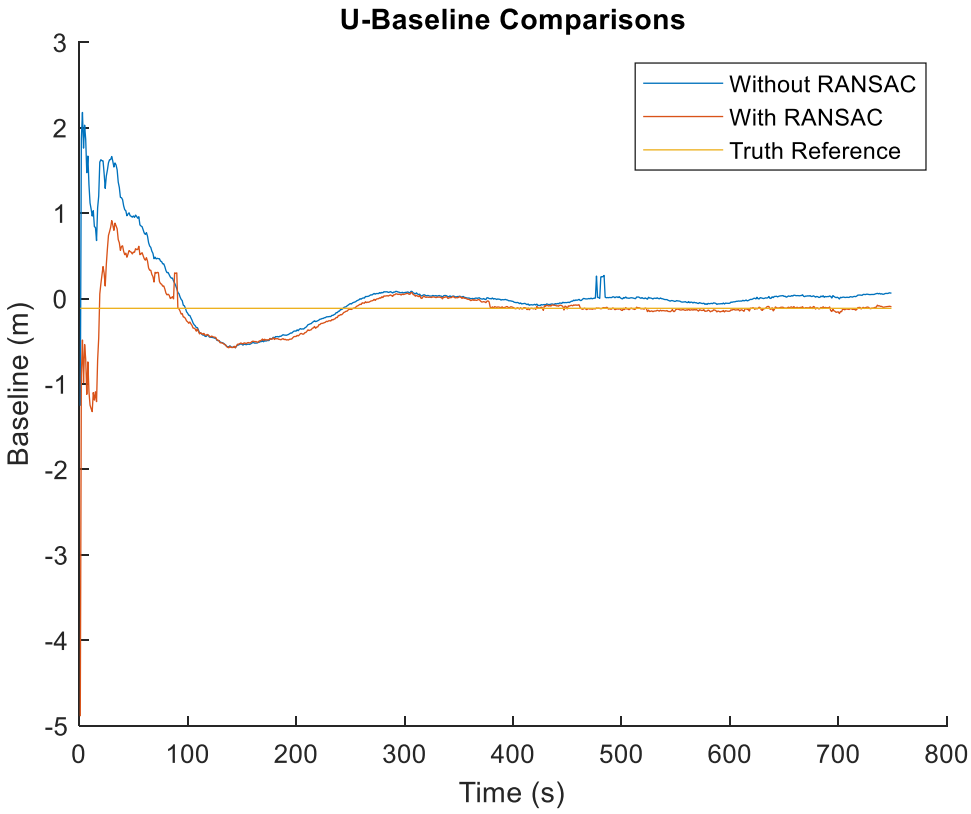

Figure 30. The U-Baseline Comparison of Dataset 2.

In Figure 31 below, the comparison with and without RANSAC is shown for the e-standard deviation.

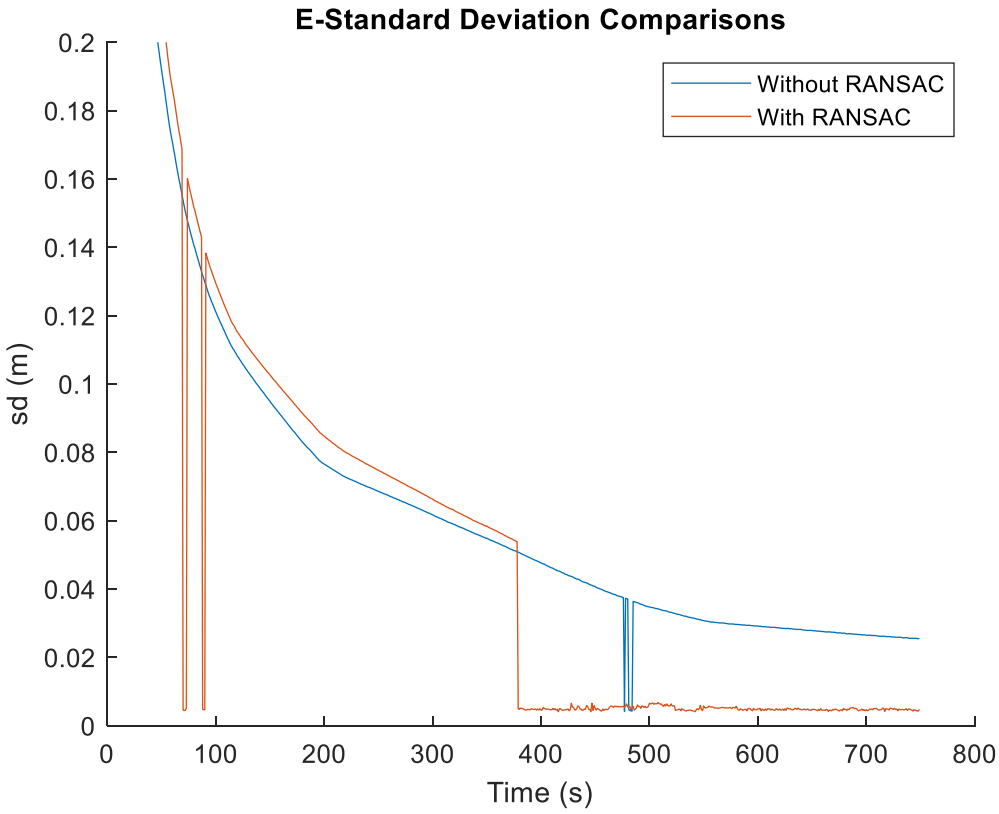

Figure 31. The E-Standard Deviation Comparison of Dataset 2.

In Figure 32 below, the comparison with and without RANSAC is shown for the n-standard deviation.

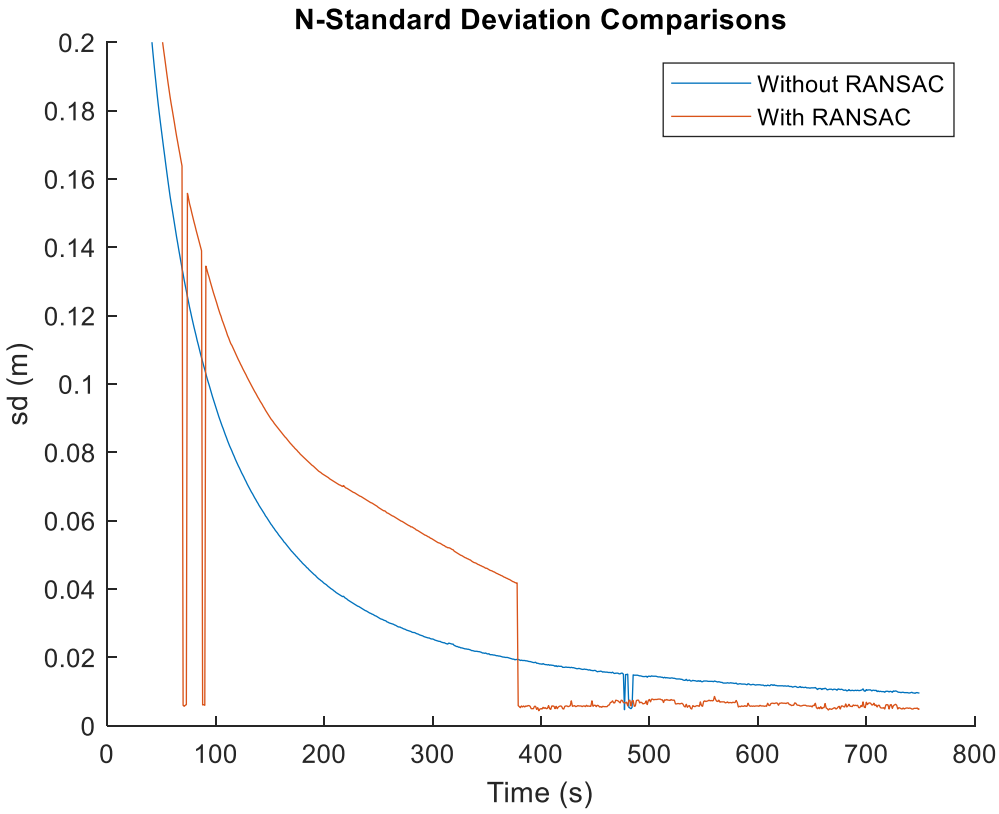

Figure 32. The N-Standard Deviation Comparison of Dataset 2.

In Figure 33 below, the comparison with and without RANSAC is shown for the u-standard deviation.

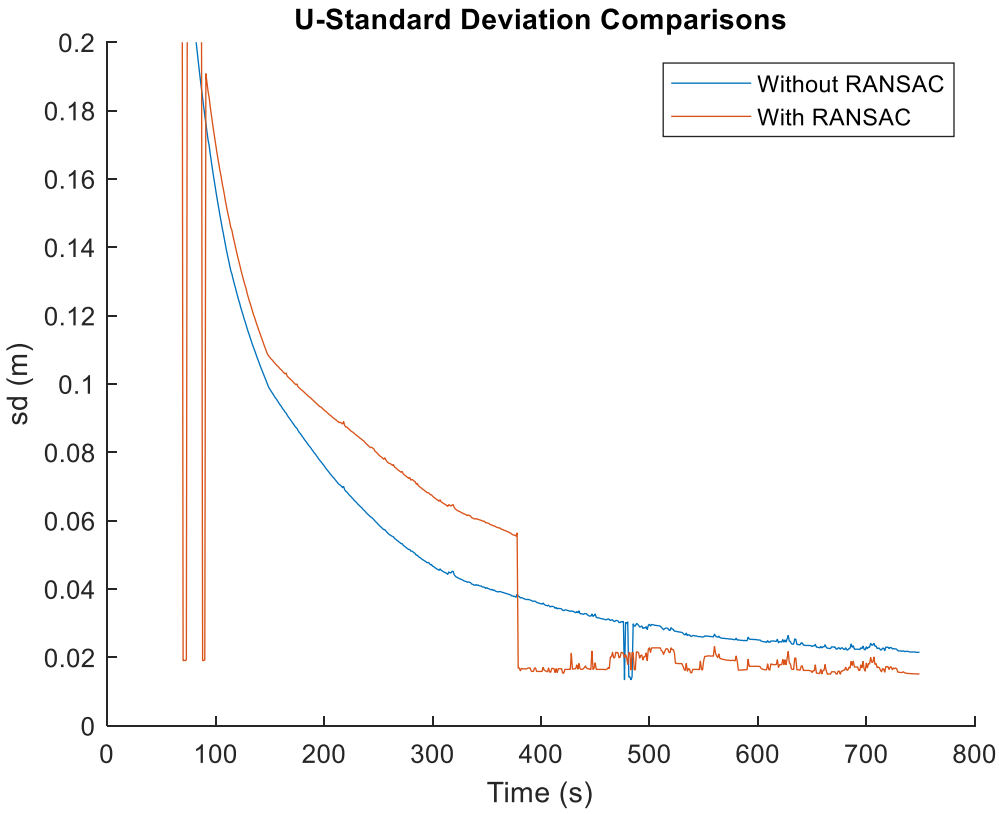

Figure 33. The U-Standard Deviation Comparison of Dataset 2.

In Figure 34 below, the comparison with and without RANSAC is shown for the status (or q value). As can be seen, with RANSAC, the system spends a great deal more time in Status 1.

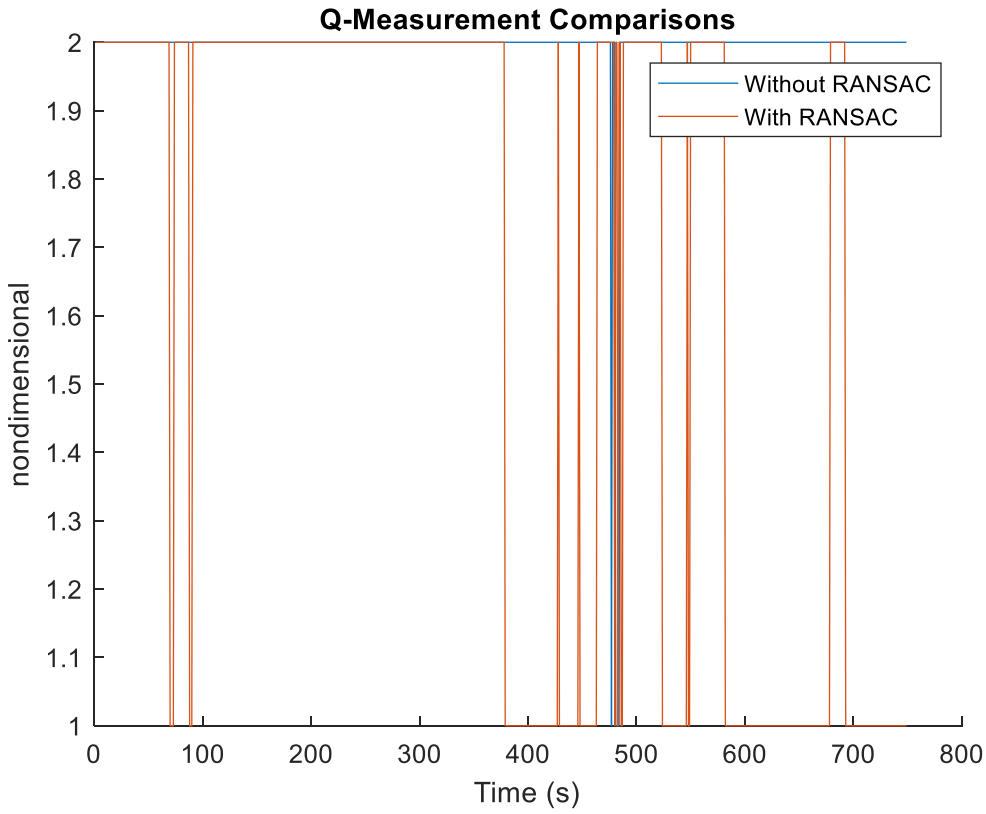

Figure 34. The Q-Measurement Comparison of Dataset 2.

In Figure 35 below, the comparison with and without RANSAC is shown for the number of satellites. As expected, the RANSAC algorithm has removed erroneous satellites using a lower number of satellites to identify its solution, however, these satellites are of higher quality.

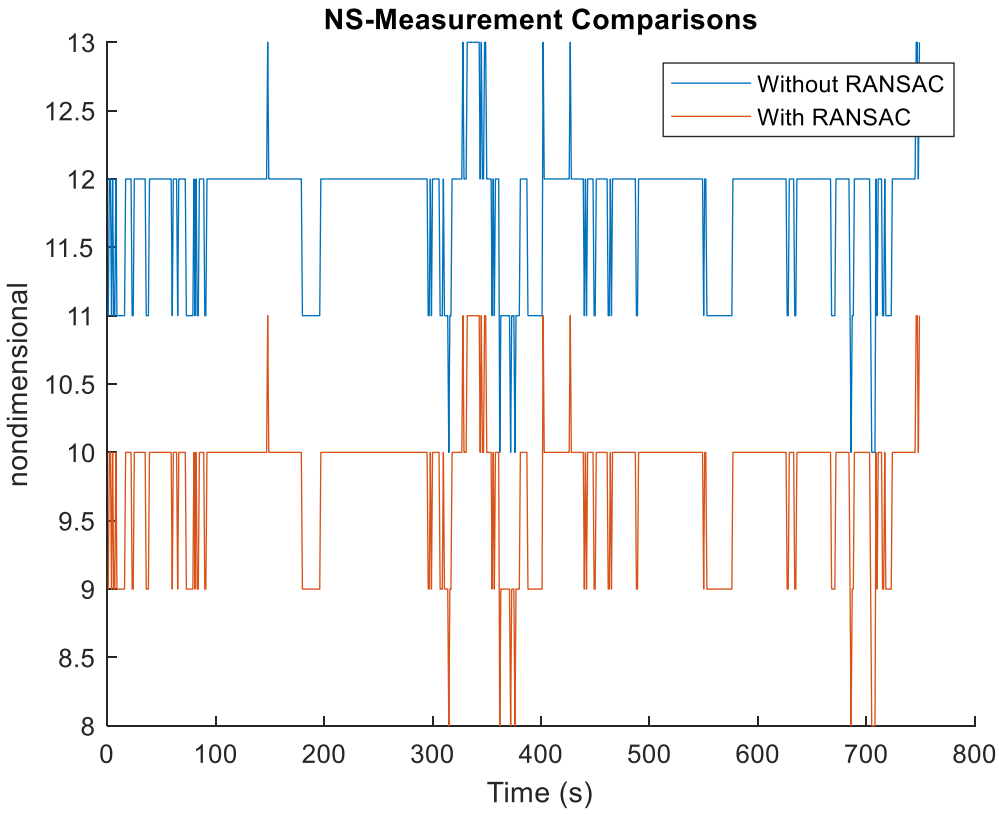

Figure 35. The NS-Measurement Comparison of Dataset 2.

In Figure 36 below, the comparison with and without RANSAC is shown for the ratio. Once more, the data is reflecting expectations. With RANSAC, there is a higher quality and confidence in the data as reflected by the ratio value. The higher the ratio, the greater confidence in the solution.

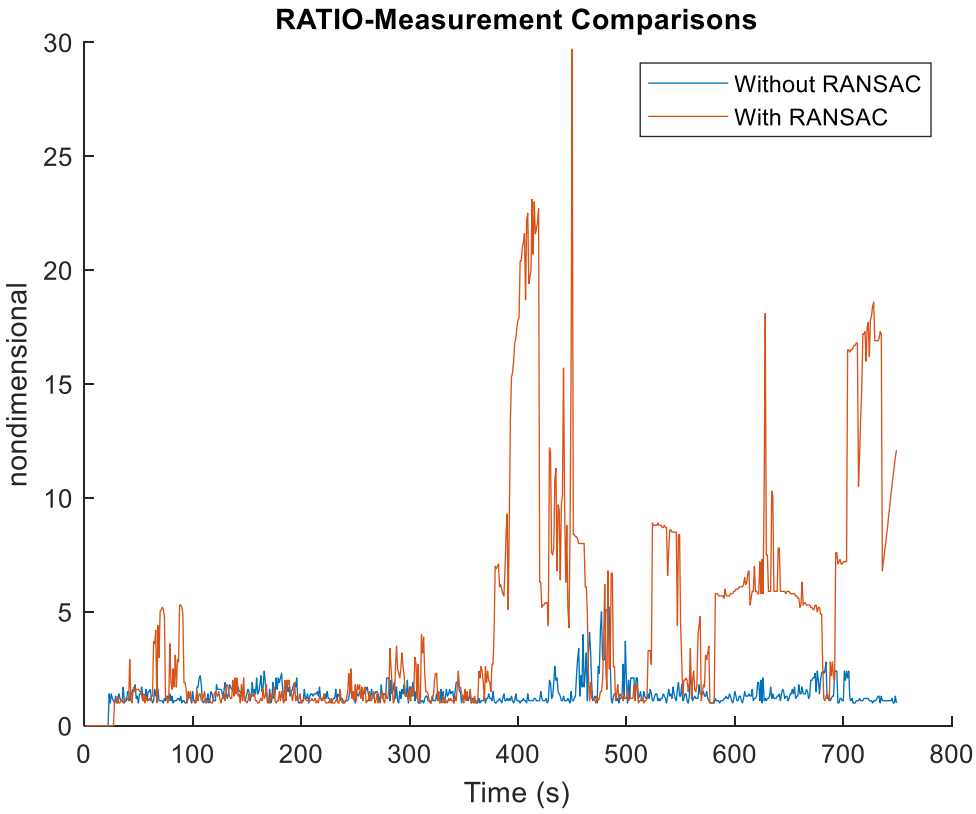

Figure 36. The RATIO-Measurement Comparison of Dataset 2.

Thus, the solution comparisons with and without RANSAC are shown below in Figure 37 with a truth reference. It clearly shows that RANSAC produces a better set of data when there are erroneous satellites.

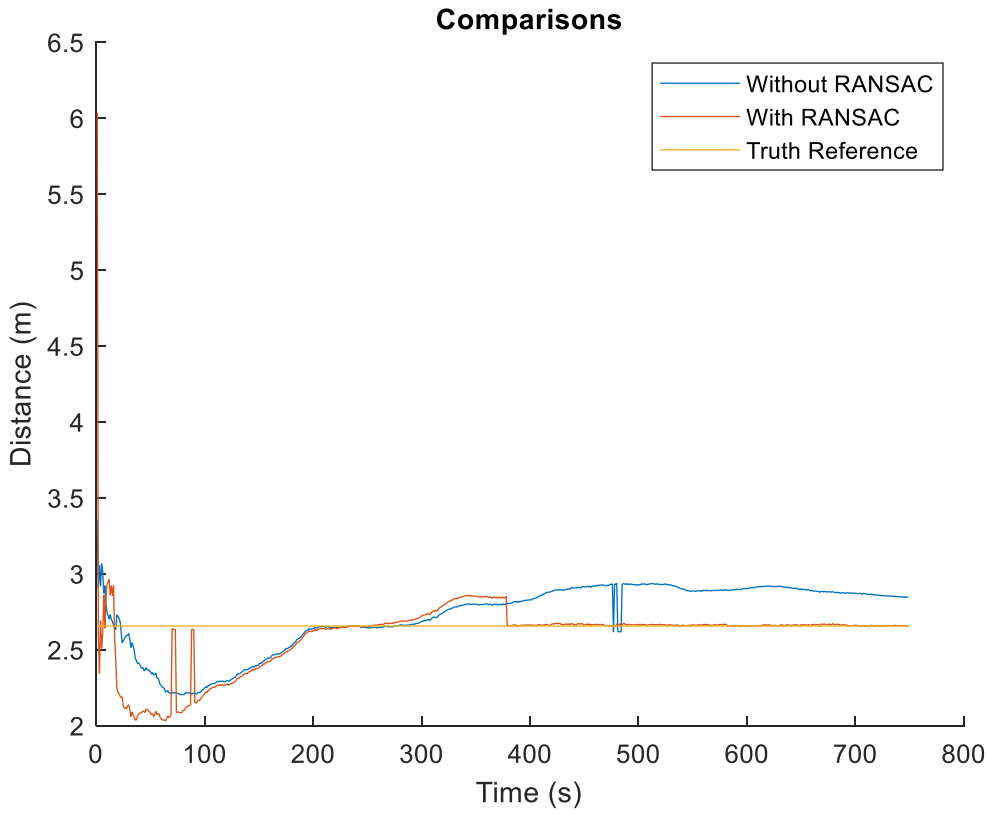

Figure 37. The measurement of distance with and without RANSAC with a truth reference of Dataset 2.

4.2 Other Datasets

Not all datasets were as good as Dataset 2. The RANSAC algorithm is only expected to improve RTK if there are erroneous carrier phase measurements from some GNSS satellites. In most of the datasets collected in an open-sky environment, these errors were not observed. One such example occurs in Dataset 4. The baseline measurements without RANSAC are shown below in Figure 38.

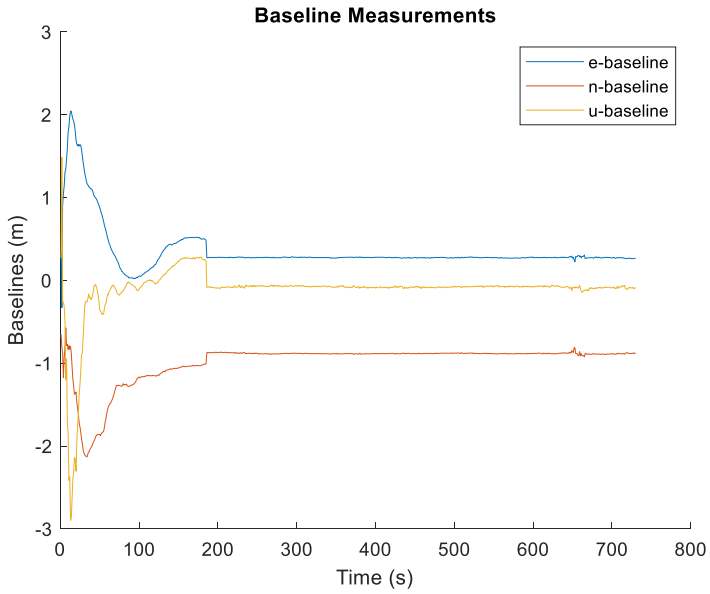

Figure 38. The baseline measurements of Dataset 4 without RANSAC.

Figure 39 shows nearly identical results with RANSAC.

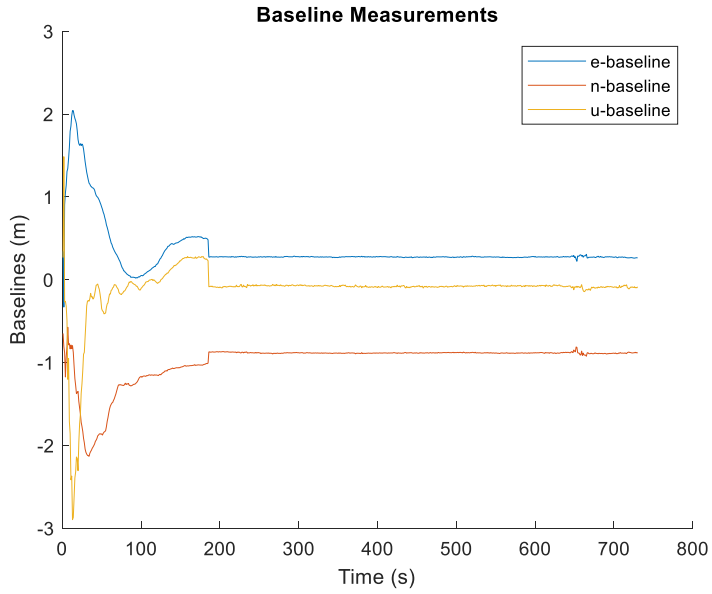

Figure 39. The baseline measurements of Dataset 4 with RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 40. There is no visual difference from Figure 41, showing the standard deviation measurements with RANSAC.

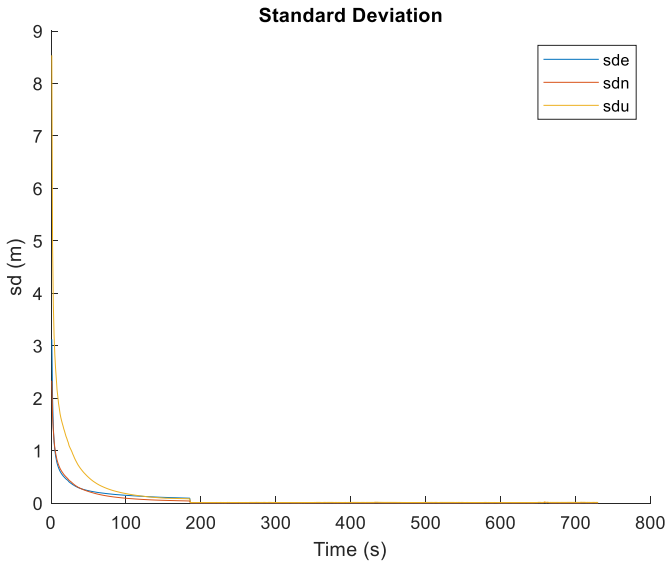

Figure 40. The standard deviation measurements of Dataset 4 without RANSAC.

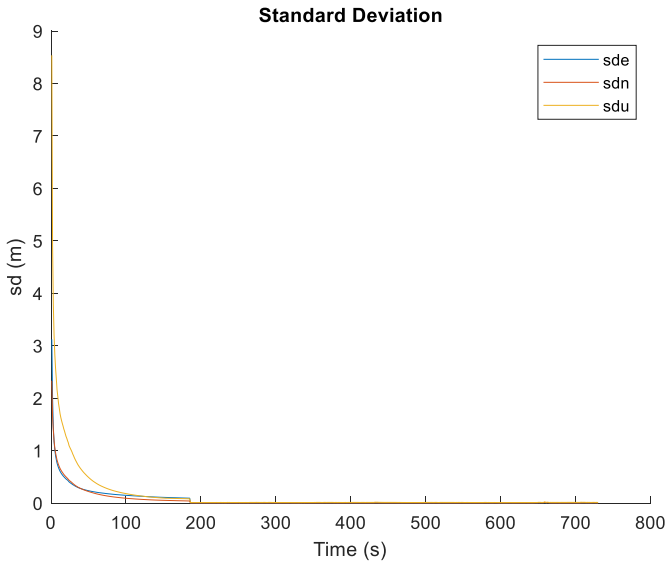

Figure 41. The standard deviation measurements of Dataset 4 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 42. Again, it shows virtually no difference from Figure 43, with RANSAC.

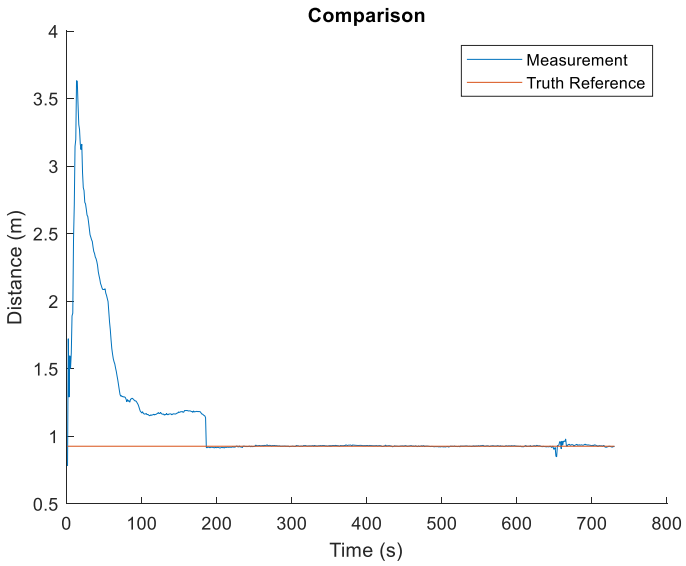

Figure 42. The distance measurements between receivers with a truth reference of Dataset 4 without RANSAC.

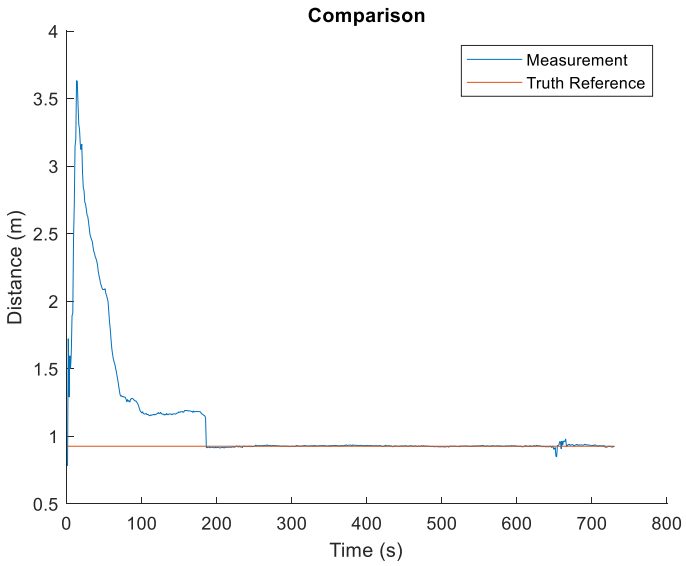

Figure 43. The distance measurements between receivers with a truth reference of Dataset 4 with RANSAC.

The quality measurements without RANSAC are shown below in Figure 44. In some cases, the number of satellites (ns) is slightly higher than that reported with RANSAC, in Figure 45. That is because RANSAC could raise a flag with health GNSS satellites, which is likely triggered by noise. These false alarms could have a small impact on the system performance.

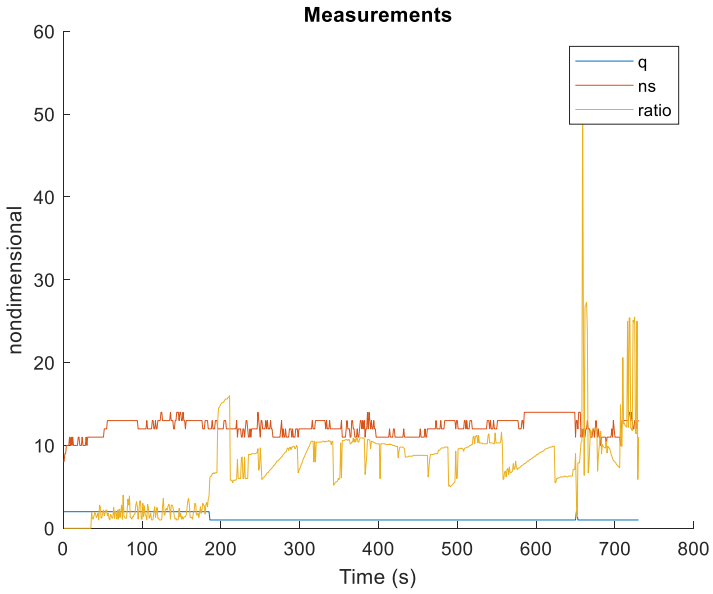

Figure 44. The quality measurements of Dataset 4 without RANSAC.

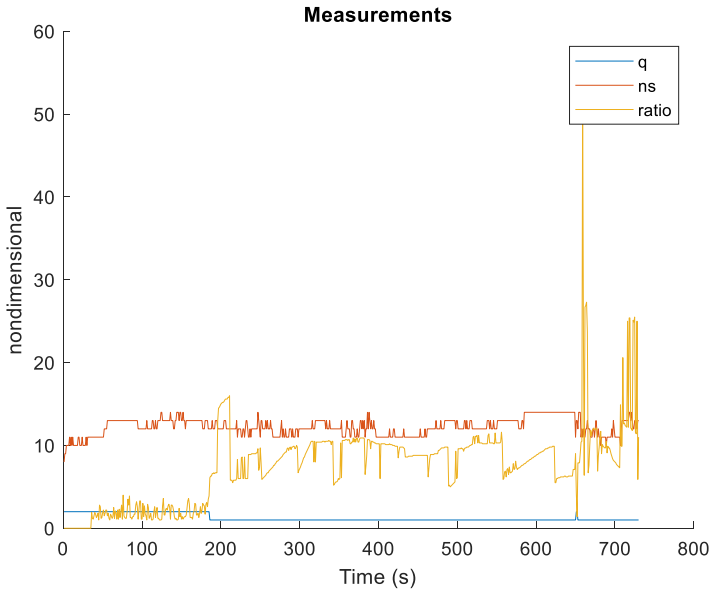

Figure 45. The quality measurements of Dataset 4 with RANSAC.

Table 1 below shows a summary of performance for all datasets with and without using RANSAC. It includes the time it takes for RTKLIB to first reach Status 1, the percentage the data stayed in Status 1 after first reaching it, and the number of satellites. As expected, RANSAC always decreased the number of satellites (if not the same number). Except for dataset 2, the excluded satellites are most likely due to false alarms in RANSAC. Therefore, excluding satellites does not necessarily improve the solution. In fact, it could impact the system performance if there is an insufficient number of satellites.

Table 1. Analysis of time to Status 1 and time spent in Status 1.

Date	1st Time to Status 1 without RANSAC (s)	Percentage in Status 1 After First Time without RANSAC	Number of Satellites	1st Time to Status 1 with RANSAC (s)	Percentage in Status 1 After First Time with RANSAC	Number of Satellites
Dataset 1	123	78.42 %	11	317	79.56 %	9
Dataset 2	477	1.47 %	12	70	40.06 %	10
Dataset 3	334	27.38 %	15	330	27.91 %	14
Dataset 4	186	99.63 %	12	186	99.63 %	12
Dataset 5	316	77.85 %	11	587	46.91 %	11
Dataset 6	456	100.00 %	13	458	100.00 %	13
Dataset 7	253	100.00 %	13	253	100.00 %	13
Dataset 8	91	84.20 %	12	91	84.20 %	12
Average	**279.5**	**71.12 %**	**12.4**	**286.5**	**72.28 %**	**11.8**

Results from additional datasets can be seen in Appendix F.

5. Discussions and Conclusions

The GNSS RTK solution was successfully improved by prefiltering carrier phase measurements using RANSAC, which is based on the inherent consistency among GNSS satellites. Using live data recorded with low-cost GNSS receivers (UBLOX C099-F9P), RANSAC was implemented as an external component of RTKLIB postprocessing to exclude erroneous satellites.

Since the datasets shown in this thesis were collected in an open-sky condition as a proof of concept, RANSAC was not expected to make a significant difference. However, one of the datasets did provide the proof that RANSAC could indeed improve GNSS RTK positioning solutions. It was shown in this dataset that if one or few erroneous satellites are preset, they will be detected by RANSAC. Once they are removed from RTK, the positioning solution is greatly improved. On the other side, it is still possible for the RANSAC algorithm to misinterpret the measurements and exclude a healthy satellite due to false alarms. It could also slightly decrease the performance of the solution in open-sky conditions.

In an actual urban environment with autonomous operations, the receivers will face more erroneous satellite measurements. In these applications, RANSAC would be able to help RTKLIB detect and exclude these satellites. Further refinement of the RANSAC algorithm and software will be required to reduce the likelihood of false alarms.

References

[1] “Demystifying the future of connected and autonomous vehicles,” *Argonne National Laboratory*, 10-May-2018. [Online]. Available: https://www.anl.gov/es/article/demystifying-the-future-of-connected-and-autonomous-vehicles. [Accessed: 23-Apr-2021].

[2] “Application fields and solutions,” *QST Corporation*, 2020. [Online]. Available: https://www.qstcorp.com/yylyfa. [Accessed: 23-Apr-2021].

[3] J. Plungis, “FCC Moves Plan Forward to Chop Up Vehicle Safety Airwaves,” *Yahoo! News*, 12-Dec-2019. [Online]. Available: https://news.yahoo.com/fcc-moves-plan-forward-chop-225248977.html?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAAVjqxrTLNFp6ynLPP_x5rKEYazREXT82pBpC3hcX26O426_XKde3vC-sPy5dOyN4C8NIpFK4e90TfPTZhbYlhEHmzGtkWE1XtL3DtMWPuvhYsVnPwpshxjG-KrzW8m_HzjwYxSVkTB-A4o7Z8PCZO0ujemCUloYL0IxdkIsqHRX. [Accessed: 23-Apr-2021].

[4] “Phantom 4 RTK,” *DJI*. [Online]. Available: https://www.dji.com/phantom-4-rtk. [Accessed: 05-May-2021].

[5] N. Rogers, “What Is Precision Agriculture?,” *Sustainable America*, 10-Jan-2014. [Online]. Available: https://sustainableamerica.org/blog/what-is-precision-agriculture/#:~:text=Precision%20agriculture%20seeks%20to%20use,using%20less%20to%20grow%20more. [Accessed: 06-May-2021].

[6] L. Dormehl, “Case IH's Self-Driving Tractors Are the Terminators of the Agricultural World,” *Digital Trends,* 02-Sep-2016. [Online]. Available: https://www.digitaltrends.com/cool-tech/self-driving-tractors/. [Accessed: 29-Apr-2021].

[7] “Differential Correction.” [Online]. Available: https://www.nrem.iastate.edu/files/wk12-GPS_DiffCorr_CarriervsCode_COStateParks.pdf. [Accessed: 06-May-2021].

[8] Z. Zhu, “notes0401,” in *MENG6700,* 01-Apr-2021.

[9] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: Principles and applications*. Boston: Artech House, 2006.

[10] “Differencing,” *PennState Department of Geography,* 2020. [Online]. Available: https://www.e-education.psu.edu/geog862/node/1727. [Accessed: 27-Apr-2021].

[11] “Where the world builds software,” *GitHub*. [Online]. Available: https://github.com/. [Accessed: 19-May-2021].

[12] M. Choudhary, “What are the various GNSS systems?,” *Geospatial World,* 20-Nov-2019. [Online]. Available: https://www.geospatialworld.net/blogs/what-are-the-various-gnss-systems/. [Accessed: 19-May-2021].

[13] T. Takasu, *RTKLIB Demo5 Manual*. 2020.

[14] B. Park and C. Kee, “The Compact Network RTK Method: An Effective Solution to Reduce GNSS Temporal and Spatial Decorrelation Error,” *Journal of Navigation*, vol. 63, *(2)*, pp. 343-362, 2010.

[15] A. Al-Shaery, S. Zhang and C. Rizos, “An enhanced calibration method of GLONASS inter-channel bias for GNSS RTK,” *GPS Solution*, vol. 17, *(2)*, pp. 165-173, 2012.

[16] B. Li *et al*, “GNSS ambiguity resolution with controllable failure rate for long baseline network RTK,” *Journal of Geodesy*, vol. 88, *(2)*, pp. 99-112, 2013.

[17] A. Brack, “Parkinson Award Nomination Form,” *GeoForschungsZentrum, Postdam*. .

[18] A. Parkins, “Increasing GNSS RTK availability with a new single-epoch batch partial ambiguity resolution algorithm,” *GPS Solutions*, vol. 15, *(4)*, pp. 391-402, 2011.

[19] K. Wang, P. Chen and P. J. G. Teunissen, “Single-Epoch, Single-Frequency Multi-GNSS l5 RTK under High-Elevation Masking,” *Sensors (Basel, Switzerland)*, vol. 19, *(5)*, pp. 1066, 2019.

[20] M. S. Garrido-Carretero *et al*, “Low-cost GNSS receiver in RTK positioning under the standard ISO-17123-8: A feasible option in geomatics,” *Measurement: Journal of the International Measurement Confederation*, vol. 137, pp. 168-178, 2019.

[21] T. Li *et al*, “Tightly-Coupled Integration of Multi-GNSS Single-Frequency RTK and MEMS-IMU for Enhanced Positioning Performance,” *Sensors (Basel, Switzerland)*, vol. 17, *(11)*, pp. 2462, 2017.

[22] Z. Zhu, E. Vinande, J. Pontious, B. Hall, and N. Shah, “A Robust Multi-Constellation Time Differenced Carrier Phase Solution.”

[23] R. B. Fisher, “The RANSAC (Random Sample Consensus) Algorithm,” 06-May-2002. [Online]. Available: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FISHER/RANSAC/. [Accessed: 27-Apr-2021].

[24] “Random sample consensus,” *Wikipedia*, 14-Oct-2020. [Online]. Available: https://en.wikipedia.org/wiki/Random_sample_consensus. [Accessed: 19-May-2021].

[25] “C099-F9P application board,” *ublox*, 15-Dec-2020. [Online]. Available: https://www.u-blox.com/en/product/c099-f9p-application-board. [Accessed: 27-Apr-2021].

[26] “PWRPAK7®,” NovAtel. [Online]. Available: https://novatel.com/products/receivers/enclosures/pwrpak7. [Accessed: 25-Mar-2022].

Appendix A

The Read.m file is shown below.

```
clear all; close all; clc


filename = 'log.txt';
fileID = fopen(filename);
n = 1;


for c = 1:25
    tline = fgetl(fileID);
end


while 1
    try
        tline = str2num(fgets(fileID));
        tline(:,1) = [];
        %tline(1) = n;
        R{n} = tline;
        n = n + 1;
    catch
        break
    end
```

```
end

filename = 'log.txt2';
fileID = fopen(filename);
n = 1;

for c = 1:25
    tline = fgetl(fileID);
end

while 1
    try
        tline = str2num(fgets(fileID));
        tline(:,1) = [];
        %tline(1) = n;
        S{n} = tline;
        n = n + 1;
    catch
        break
    end
end

%%
```

```
x = 1;

y = 1;


A = size(R);

a = A(2);

C = size(R{1});

c = C(2);


d{x} = [1:14];

d{x}(y) = R{x}(y) - S{x}(y);


while x < a
   while y < c
      y = y + 1;
      d{x}(y) = R{x}(y) - S{x}(y);
   end
   x = x + 1;
   d{x} = [1:14];
   y = 1;
   d{x}(y) = R{x}(y) - S{x}(y);
   C = size(R{x});
   c = C(2);
```

```
end


while y < c
    y = y + 1;
    d{x}(y) = R{x}(y) - S{x}(y);
end
```

Appendix B

The RData.m file is shown below.

```
close all


i = 1;
while i < n
   I{i} = i;
   if length(R{i}) > 13
      qu = length(R{i});
      diff = qu - 13;
      z = R{i};
      for u = 1:(diff)
         z(1) = [];
      end
      R{i} = z;
   end
   e{i} = R{i}(1);
   k{i} = R{i}(2);
   o{i} = R{i}(3);
   h{i} = R{i}(4);
   ns{i} = R{i}(5);
   sde{i} = R{i}(6);
```

```
    sdn{i} = R{i}(7);
    sdu{i} = R{i}(8);
    ratio{i} = R{i}(13);
    i = i + 1;
end

i = 1;
for u = 1:length(e)
    E(i) = e{i};
    K(i) = k{i};
    O(i) = o{i};
    SDE(i) = sde{i};
    SDN(i) = sdn{i};
    SDU(i) = sdu{i};
    Q(i) = h{i};
    NS(i) = ns{i};
    RATIO(i) = ratio{i};
    i = i + 1;
end

hold on
plot(E)
plot(K)
```

```
plot(O)
title('Baseline Measurements')
xlabel('Time (s)')
ylabel('Baselines (m)')
legend({'e-baseline','n-baseline','u-baseline'},'Location','northeast')

figure
hold on
plot(SDE)
plot(SDN)
plot(SDU)
title('SD Measurements')
xlabel('Time (s)')
ylabel('sd (m)')
legend({'sde','sdn','sdu'},'Location','northeast')

figure
hold on
plot(Q)
plot(NS)
plot(RATIO)
title('Measurements')
xlabel('Time (s)')
```

```
ylabel('nondimensional')
legend({'q','ns','ratio'},'Location','northeast')


i = 1;
for u = 1:length(R)
    v = [R{i}(1) R{i}(2) R{i}(3)];
    V(i) = norm(v);
    Act(i) = 0.93;
    i = i + 1;
end

figure
hold on
plot(V)
plot(Act)
```

Appendix C

The SData.m file is shown below.

```
close all

i = 1;
while i < n
   I{i} = i;
   if length(S{i}) > 13
      qu = length(S{i});
      diff = qu - 13;
      z = S{i};
      for u = 1:(diff)
         z(1) = [];
      end
      S{i} = z;
   end
   e{i} = S{i}(1);
   k{i} = S{i}(2);
   o{i} = S{i}(3);
   h{i} = S{i}(4);
   ns{i} = S{i}(5);
```

```
    sde{i} = S{i}(6);
    sdn{i} = S{i}(7);
    sdu{i} = S{i}(8);
    ratio{i} = S{i}(13);
    i = i + 1;
end

i = 1;
for u = 1:length(e)
    E1(i) = e{i};
    K1(i) = k{i};
    O1(i) = o{i};
    SDE1(i) = sde{i};
    SDN1(i) = sdn{i};
    SDU1(i) = sdu{i};
    Q1(i) = h{i};
    NS1(i) = ns{i};
    RATIO1(i) = ratio{i};
    Diff(i) = (E1(i) + K1(i) + O1(i))/3;
    Act1(i) = 0.9271;
    i = i + 1;
end
```

```
figure
hold on
plot(E1)
plot(K1)
plot(O1)
title('Baseline Measurements')
xlabel('Time (s)')
ylabel('Baselines (m)')
legend({'e-baseline','n-baseline','u-baseline'},'Location','northeast')

figure
hold on
plot(SDE1)
plot(SDN1)
plot(SDU1)
title('Standard Deviation')
xlabel('Time (s)')
ylabel('sd (m)')
legend({'sde','sdn','sdu'},'Location','northeast')

figure
hold on
plot(Q1)
```

```
plot(NS1)
plot(RATIO1)
title('Measurements')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'q','ns','ratio'},'Location','northeast')



i = 1;
for u = 1:length(S)
    v = [S{i}(1) S{i}(2) S{i}(3)];
    V1(i) = norm(v);
    Act1(i) = 0.9271;
    truth1(i) = -2.62;
    truth2(i) = -0.427;
    truth3(i) = -0.114;
    i = i + 1;
end

figure
hold on
plot(V1)
plot(Act1)
```

```
title('Comparison')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'Measurement','Truth Reference'},'Location','northeast')

%%
figure
hold on
plot(E)
plot(E1)
plot(truth1)
title('E-Baseline Comparisons')
xlabel('Time (s)')
ylabel('Baseline (m)')
legend({'Without RANSAC','With RANSAC','Truth Reference'},'Location','northeast')

figure
hold on
plot(K)
plot(K1)
plot(truth2)
title('N-Baseline Comparisons')
xlabel('Time (s)')
```

```
ylabel('Baseline (m)')
legend({'Without RANSAC','With RANSAC','Truth Reference'},'Location','northeast')

figure
hold on
plot(O)
plot(O1)
plot(truth3)
title('U-Baseline Comparisons')
xlabel('Time (s)')
ylabel('Baseline (m)')
legend({'Without RANSAC','With RANSAC','Truth Reference'},'Location','northeast')

%%
figure
hold on
plot(SDE)
plot(SDE1)
title('E-Standard Deviation Comparisons')
xlabel('Time (s)')
ylabel('sd (m)')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')
```

```
figure
hold on
plot(SDN)
plot(SDN1)
title('N-Standard Deviation Comparisons')
xlabel('Time (s)')
ylabel('sd (m)')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')

figure
hold on
plot(SDU)
plot(SDU1)
title('U-Standard Deviation Comparisons')
xlabel('Time (s)')
ylabel('sd (m)')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')

%%
figure
hold on
plot(Q)
plot(Q1)
```

```
title('Q-Measurement Comparisons')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')

figure
hold on
plot(NS)
plot(NS1)
title('NS-Measurement Comparisons')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')

figure
hold on
plot(RATIO)
plot(RATIO1)
title('RATIO-Measurement Comparisons')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'Without RANSAC','With RANSAC'},'Location','northeast')
```

```
%%
figure
hold on
plot(V)
plot(V1)
plot(Act)
title('Comparisons')
xlabel('Time (s)')
ylabel('nondimensional')
legend({'Without RANSAC','With RANSAC','Truth Reference'},'Location','northeast')
```

Appendix D

The Data.m file is shown below.

```
close all; clc

i = 1;
f = 1;
g = 1;

while i < n;
   if Q(i) == 1;
      F(f) = i;
      f = f + 1;
   end
    if Q1(i) ==1;
       G(g) = i;
       g = g + 1;
    end
   i = i +1;
end

B = size(F);
```

```
b = B(2) - 1;

rem = (n - 1) - F(1);

per = (b/rem)*100;


P = size(G);

p = P(2) - 1;

REM = (n - 1) - G(1);

PER = (p/REM)*100;



fprintf('\nThe first time the system reaches status 1 without RANSAC is at %d s.\n',F(1))

fprintf('The percentage of time the system stays in status 1 without RANSAC after first reaching
it is %.2f percent.\n',per)


fprintf('\nThe first time the system reaches status 1 with RANSAC is at %d s.\n',G(1))

fprintf('The percentage of time the system stays in status 1 with RANSAC after first reaching it
is %.2f percent.\n\n',PER)
```

Appendix E

Steps to Analyze Data

1. Rename data file to RecA.ubx and RecB.ubx from their .bin files.
2. Move files to /home/pi/RTKLIB-demo5-working/app/convbin/gcc.
3. Run "./convbin -os RecA.ubx" and "./convbin -os RecB.ubx".
    a. Once run, there should be a .nav, .obs, .sbs, and .ubx file for each receiver (8 files total).
4. Move these 8 files into their own folder named after the date (i.e. 02_28_2022).
5. Move this folder to /home/pi/datafiles.
6. Use the readme.txt found under /home/pi/datafiles/ransac for next steps.
    a. Change the folder path to the applicable data set for the RecA.obs, RecB.obs, RecA.nav, and log.txt files.
    b. DO NOT change the folder path for the opts1.conf file.
7. Copy this line of code into the terminal window and execute.
    a. There are three stages to the analysis. Five folders will be made log.txt, log.txt.stat, log.txt2, log.txt2.stat, and log_events.pos.
    b. May need to be in ~/RTKLIB-demo5-working/app/rnx2rtkp/gcc to execute.
8. Use flash drive to move this folder to Windows for postprocessing in Matlab.

Appendix F

F.1 Dataset 1

One such case where the system actually ran more efficiently before RANSAC was included was Dataset 1. The baseline measurements without RANSAC are shown below in Figure 46.

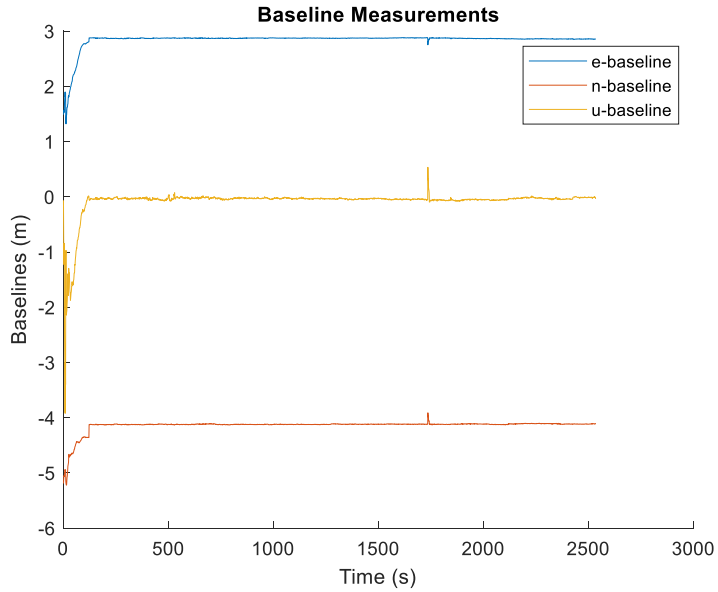

Figure 46. The baseline measurements of Dataset 1 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 47.

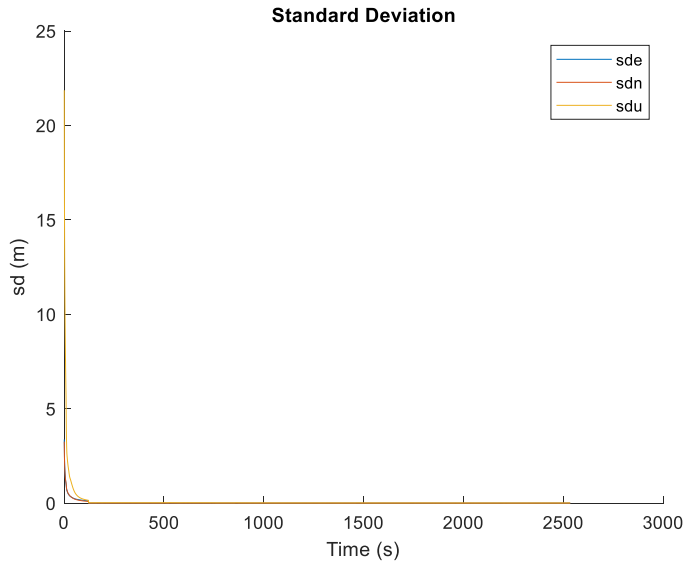

Figure 47. The standard deviation measurements of Dataset 1 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 48.

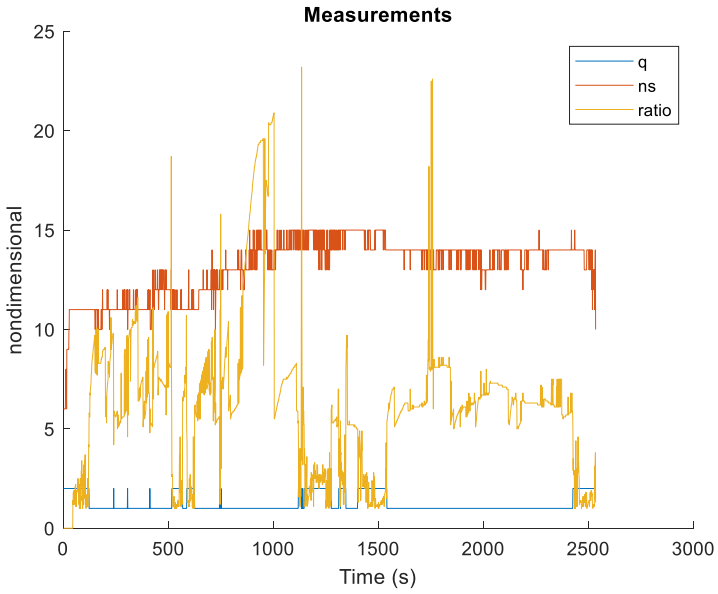

Figure 48. The quality measurements of Dataset 1 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 49.

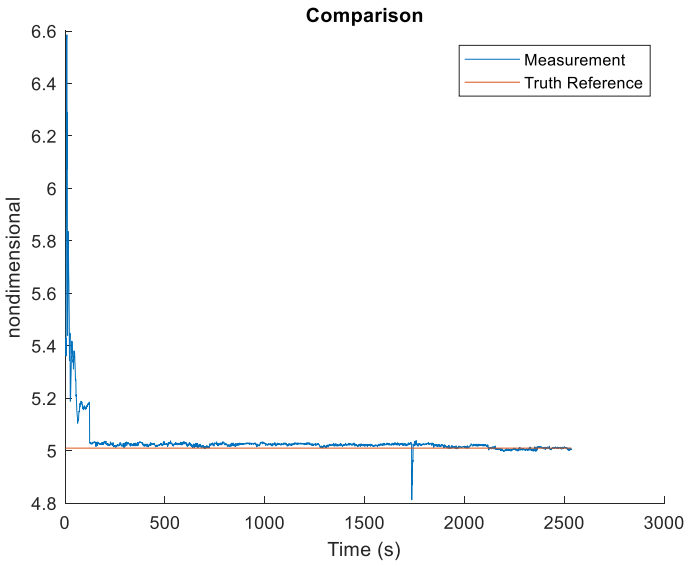

Figure 49. The distance measurements between receivers with a truth reference of Dataset 1 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 50.

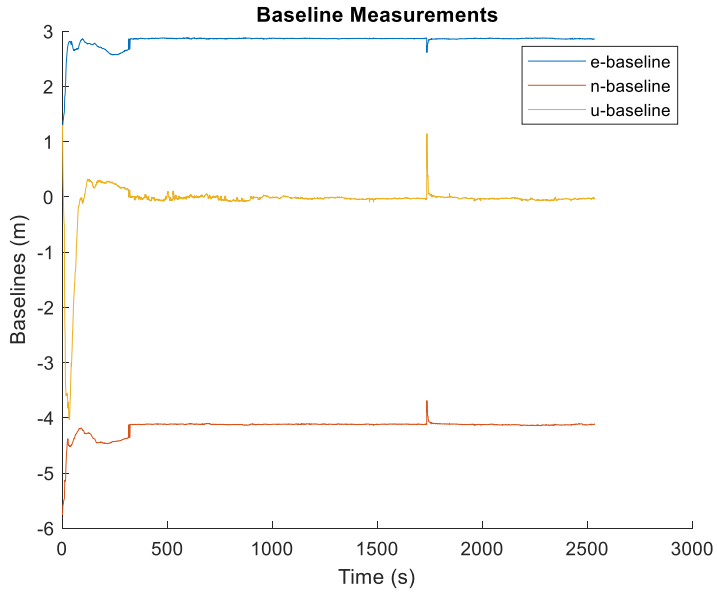

Figure 50. The baseline measurements of Dataset 1 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 51.

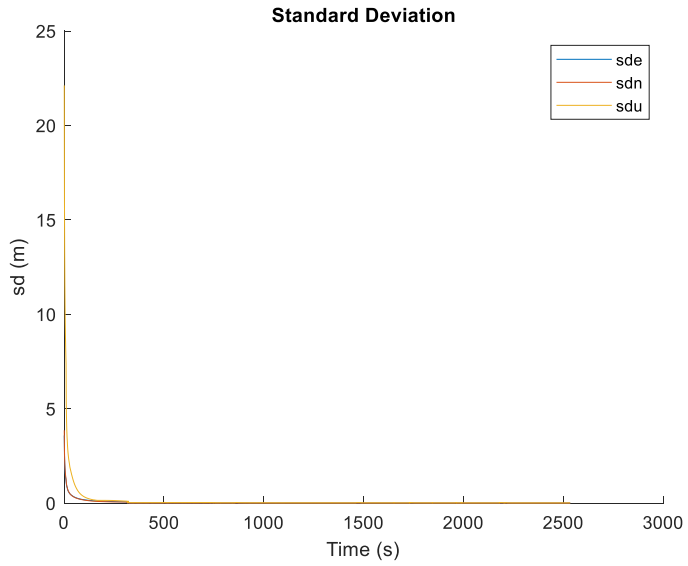

Figure 51. The standard deviation measurements of Dataset 1 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 52.

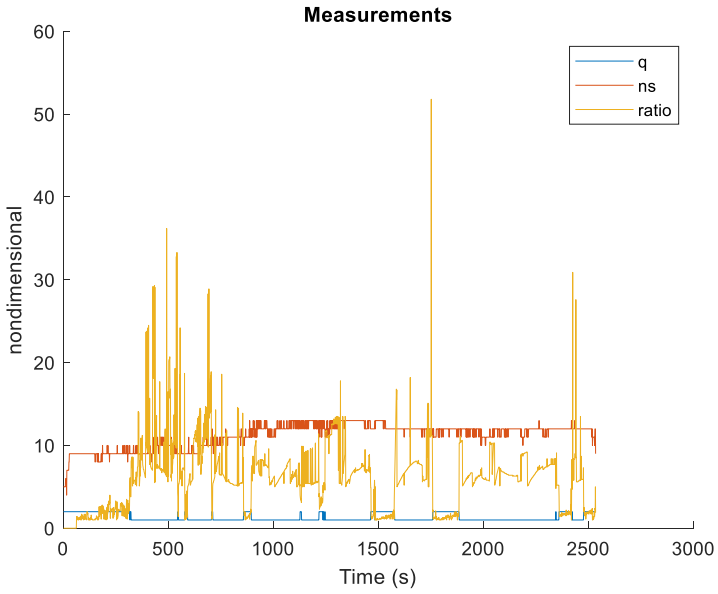

Figure 52. The quality measurements of Dataset 1 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 53.

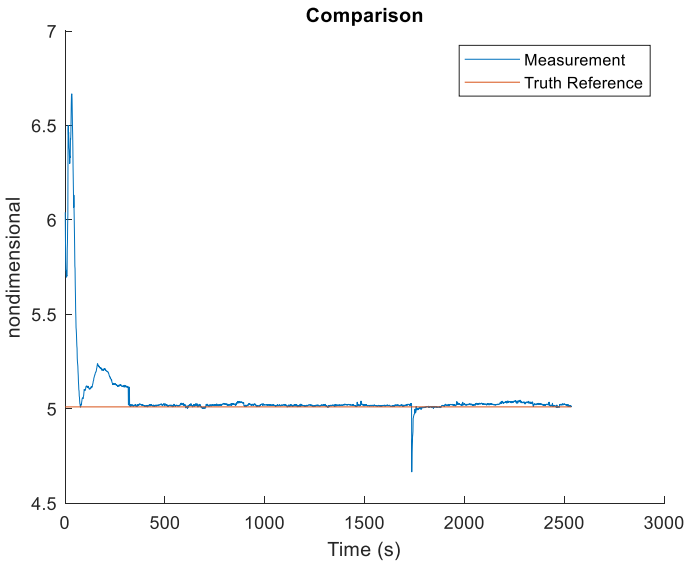

Figure 53. The distance measurements between receivers with a truth reference of Dataset 1 with RANSAC.

F.2 Dataset 3

The baseline measurements without RANSAC are shown below in Figure 54.

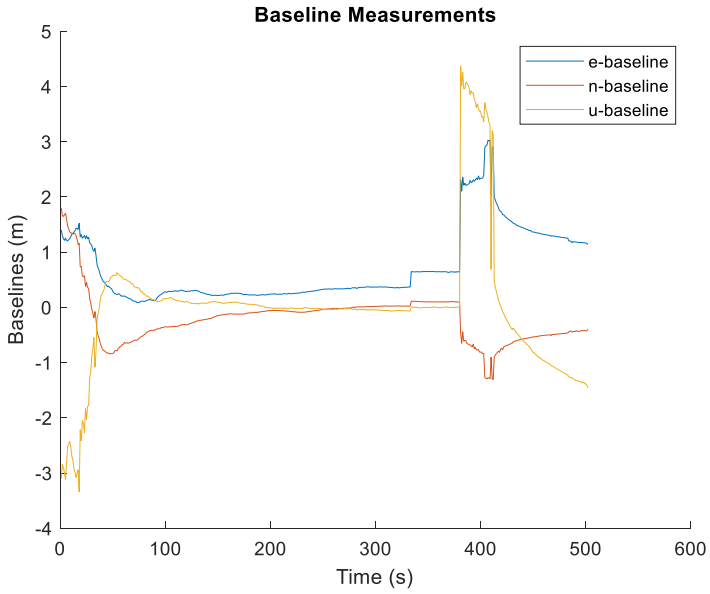

Figure 54. The baseline measurements of Dataset 3 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 55.

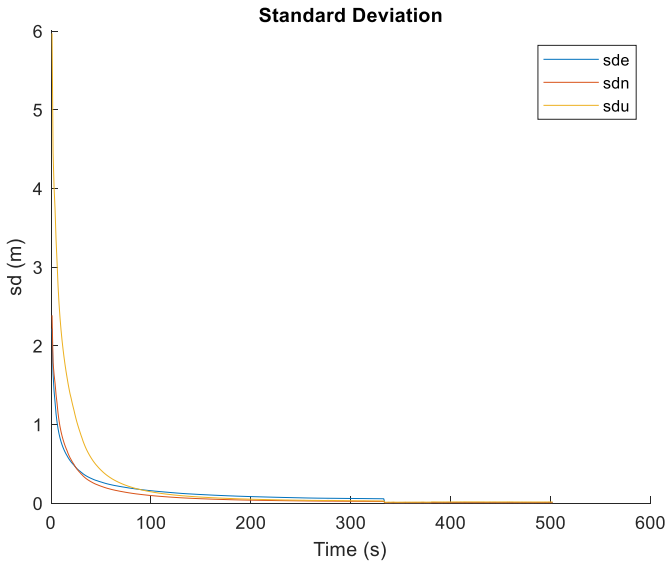

Figure 55. The standard deviation measurements of Dataset 3 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 56.

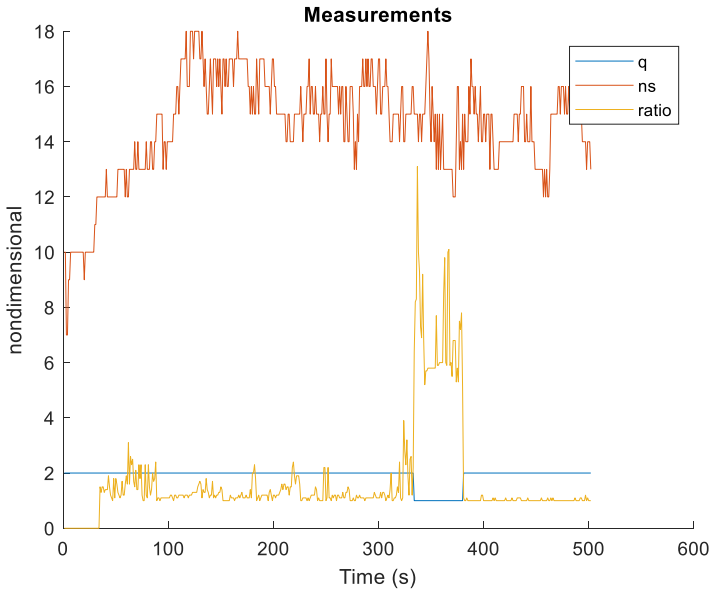

Figure 56. The quality measurements of Dataset 3 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 57.

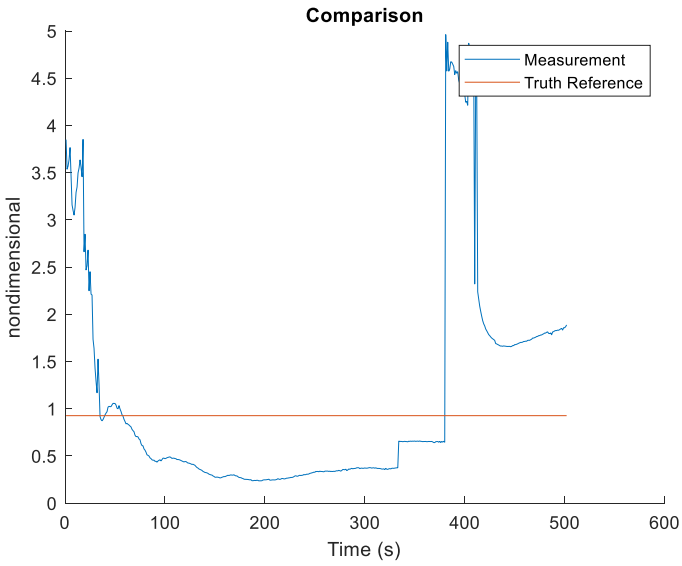

Figure 57. The distance measurements between receivers with a truth reference of Dataset 3 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 58.

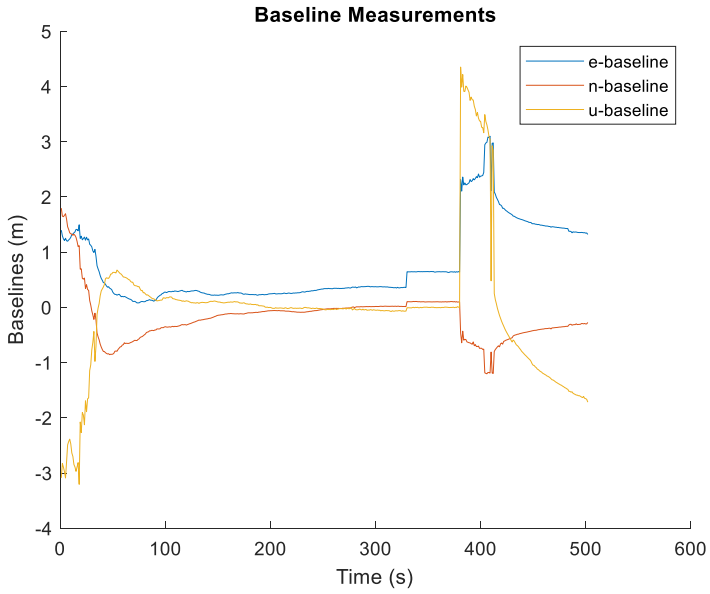

Figure 58. The baseline measurements of Dataset 3 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 59.

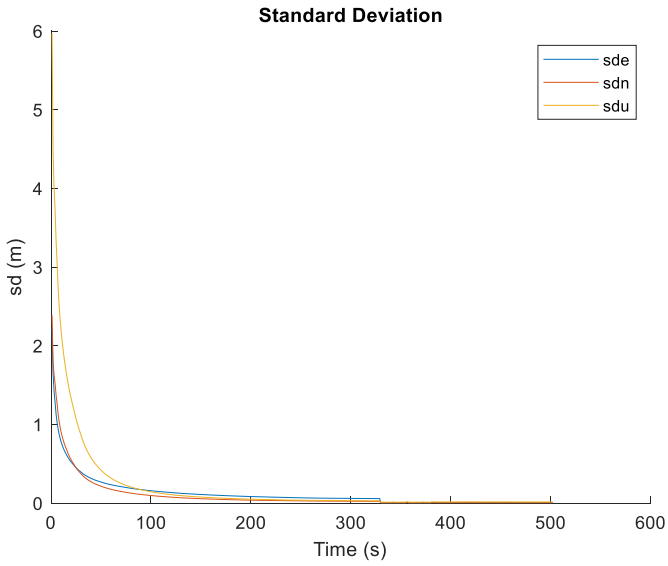

Figure 59. The standard deviation measurements of Dataset 3 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 60.

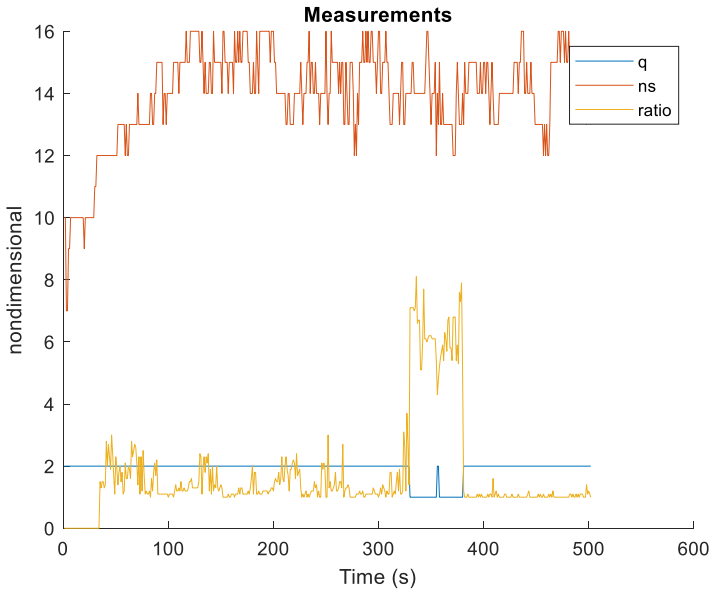

Figure 60. The quality measurements of Dataset 3 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 61.

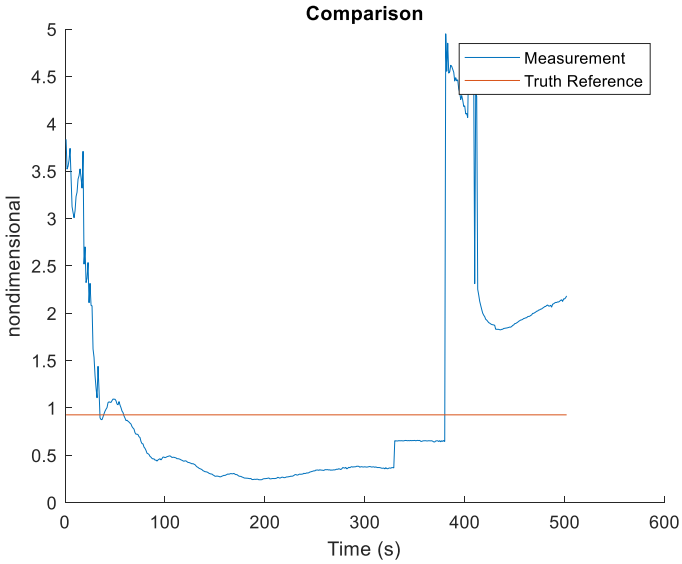

Figure 61. The distance measurements between receivers with a truth reference of Dataset 3 with RANSAC.

F.3 Dataset 5

The baseline measurements without RANSAC are shown below in Figure 62.

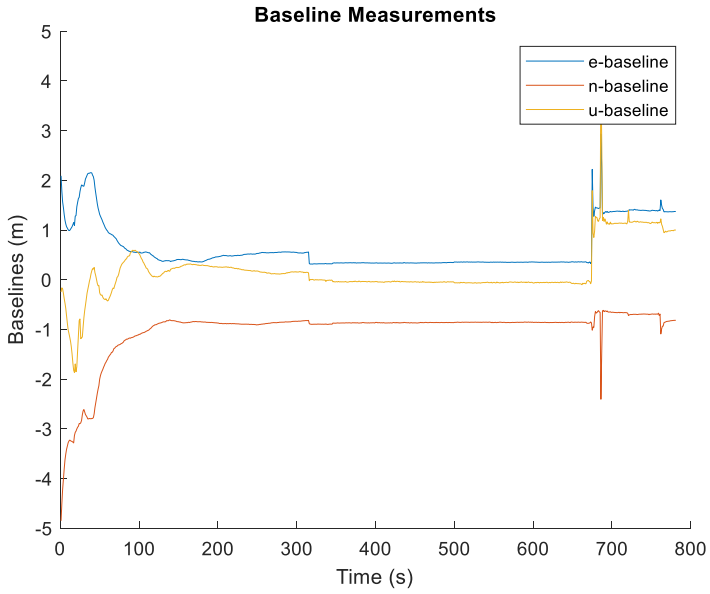

Figure 62. The baseline measurements of Dataset 5 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 63.

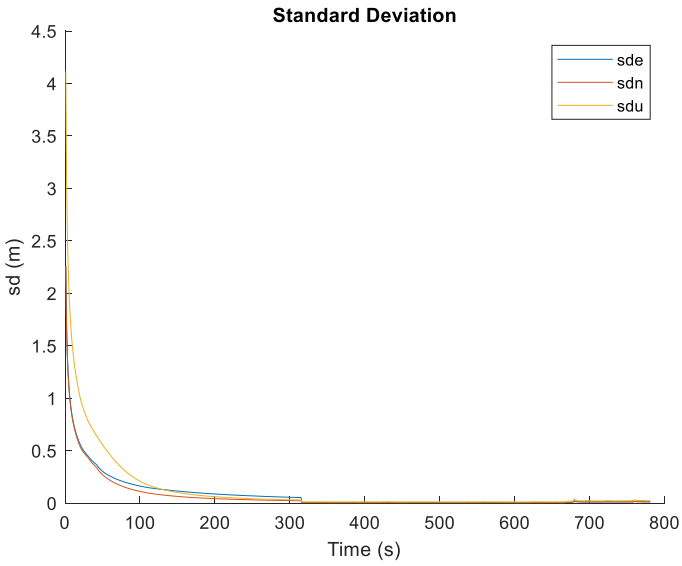

Figure 63. The standard deviation measurements of Dataset 5 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 64.

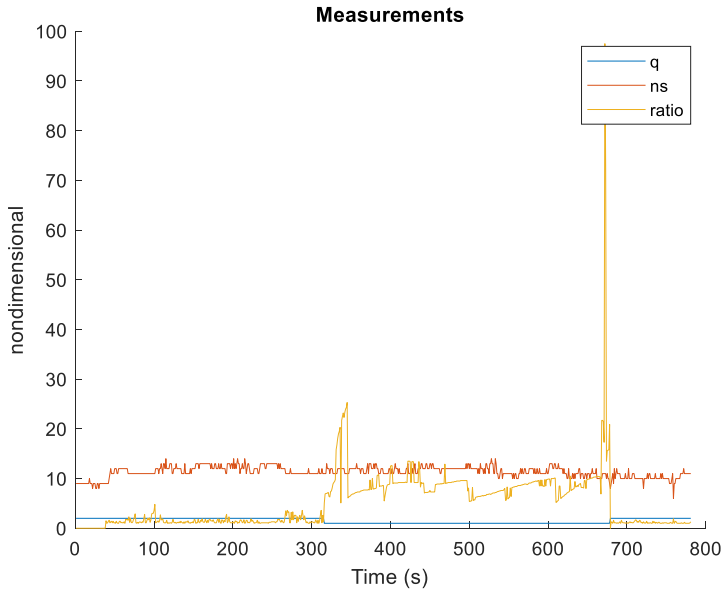

Figure 64. The quality measurements of Dataset 5 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 65.

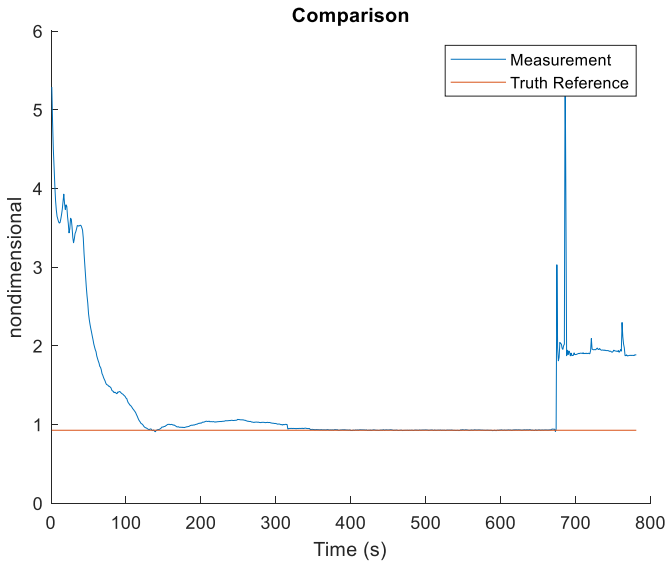

Figure 65. The distance measurements between receivers with a truth reference of Dataset 5 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 66.

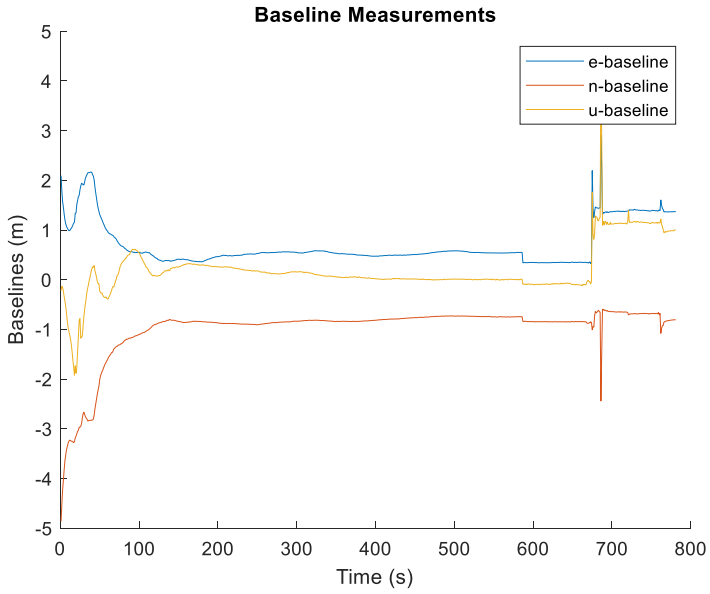

Figure 66. The baseline measurements of Dataset 5 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 67.

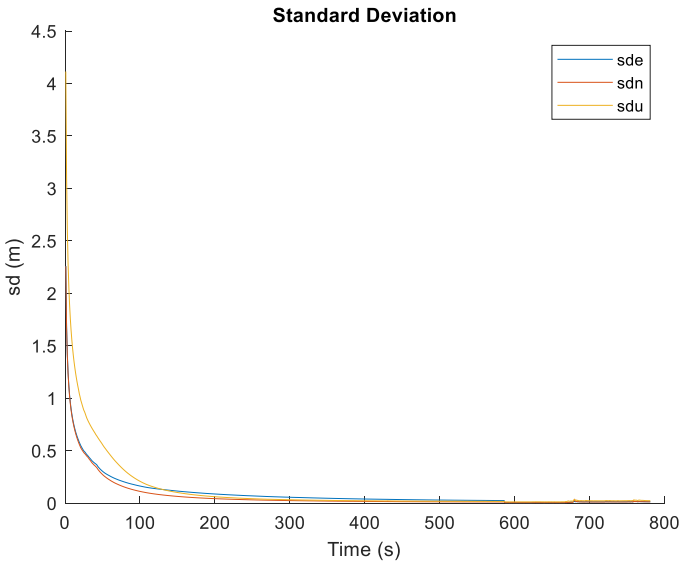

Figure 67. The standard deviation measurements of Dataset 5 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 68.

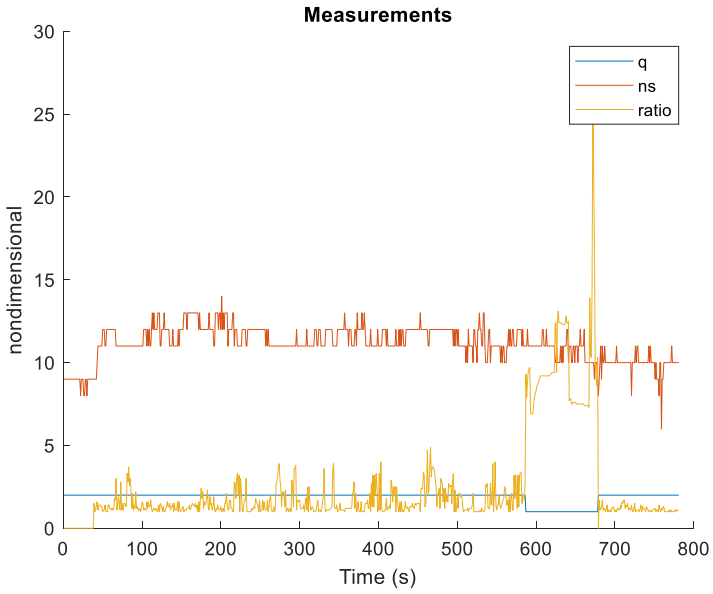

Figure 68. The quality measurements of Dataset 5 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 69.

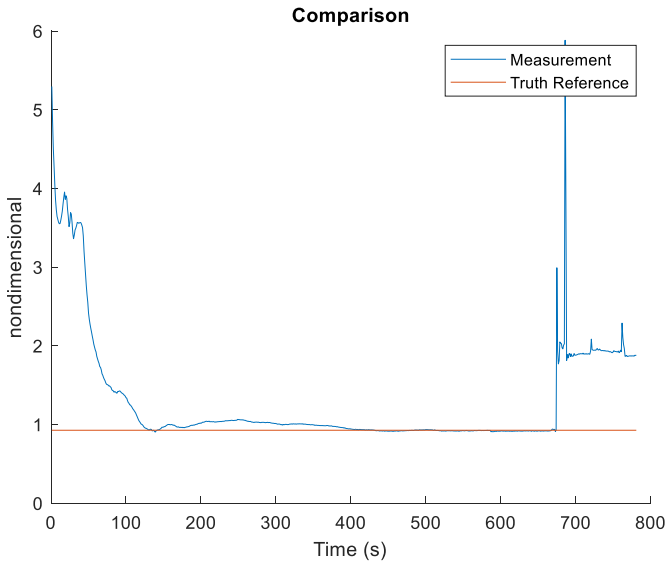

Figure 69. The distance measurements between receivers with a truth reference of Dataset 5 with RANSAC.

F.4 Dataset 6

The baseline measurements without RANSAC are shown below in Figure 70.

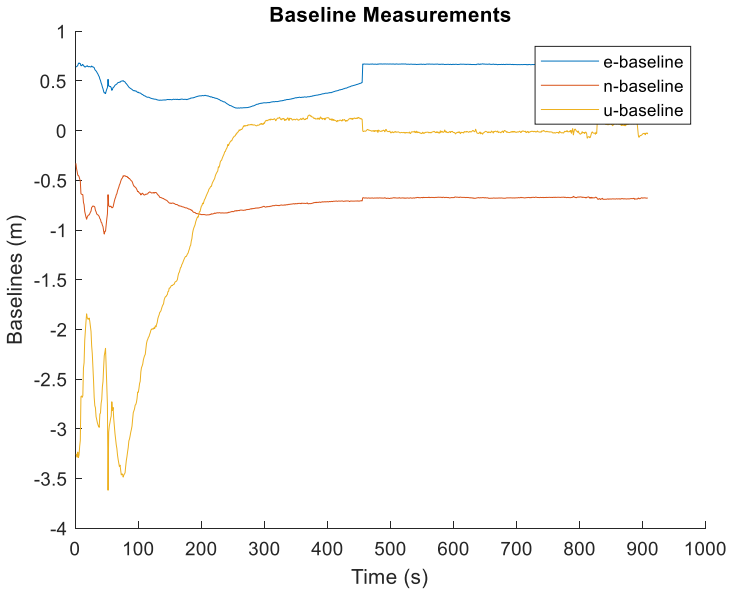

Figure 70. The baseline measurements of Dataset 6 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 71.

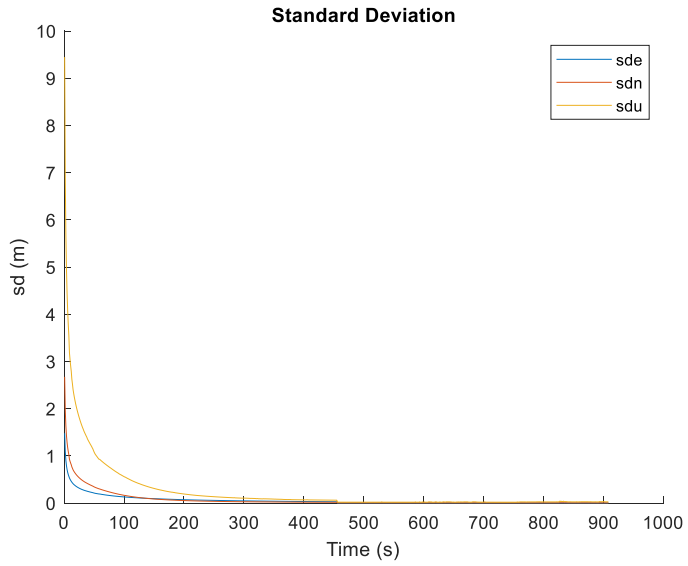

Figure 71. The standard deviation measurements of Dataset 6 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 72.

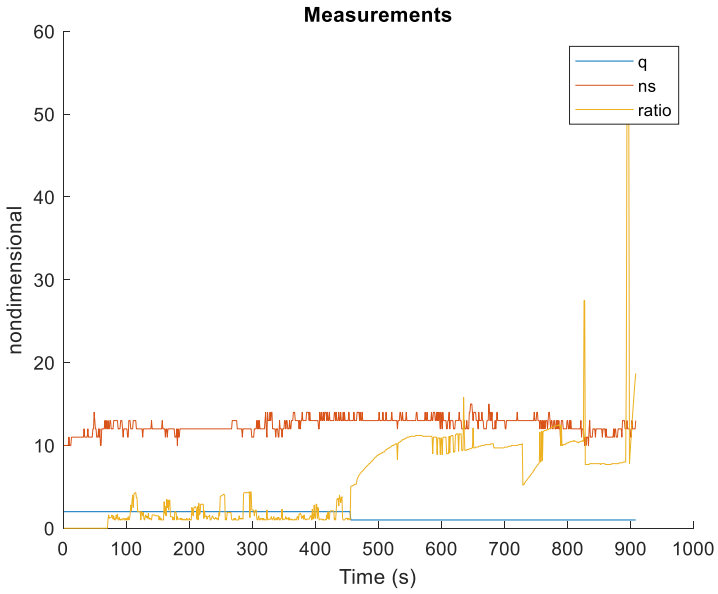

Figure 72. The quality measurements of Dataset 6 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 73.

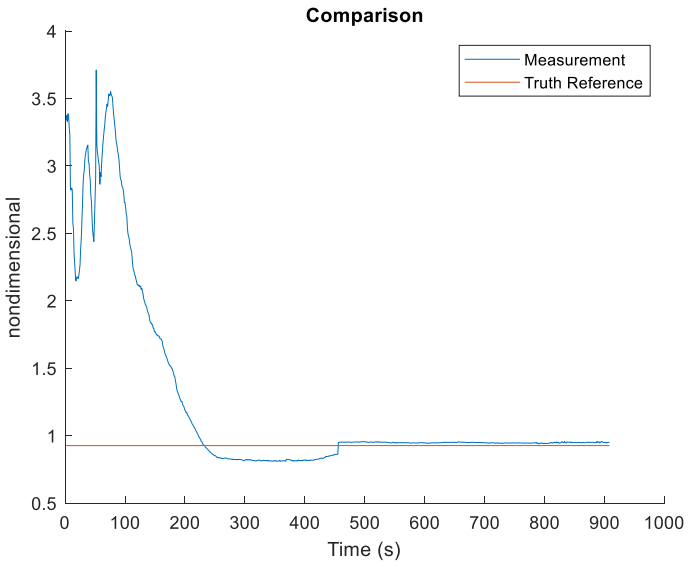

Figure 73. The distance measurements between receivers with a truth reference of Dataset 6 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 74.

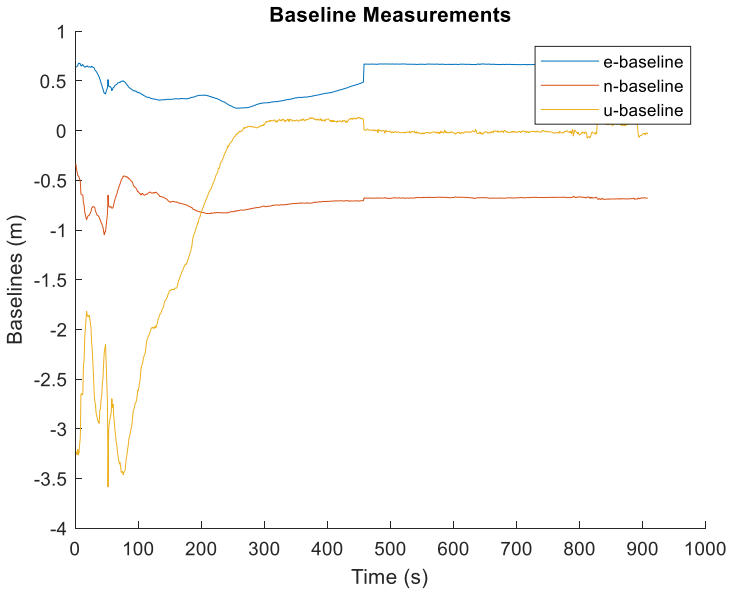

Figure 74. The baseline measurements of Dataset 6 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 75.

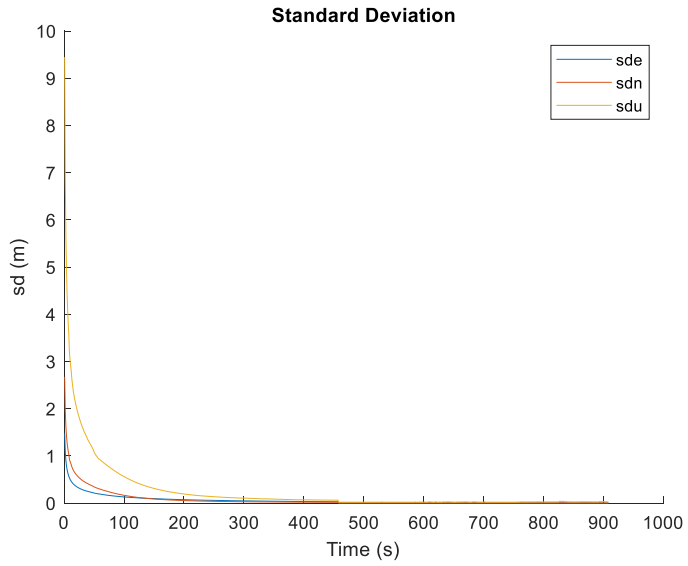

Figure 75. The standard deviation measurements of Dataset 6 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 76.

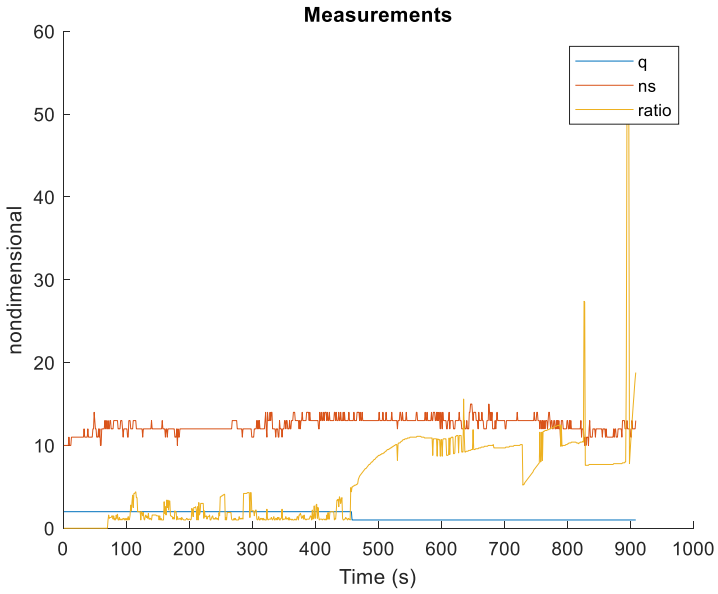

Figure 76. The quality measurements of Dataset 6 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 77.

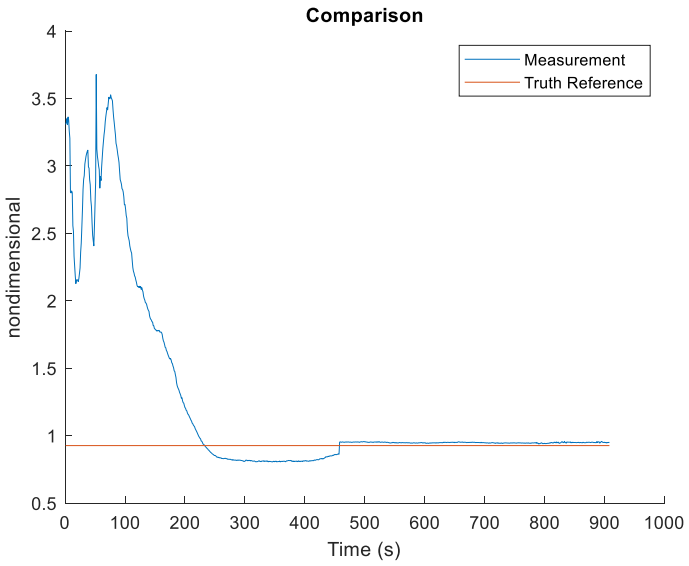

Figure 77. The distance measurements between receivers with a truth reference of Dataset 6 with RANSAC.

F.5 Dataset 7

The baseline measurements without RANSAC are shown below in Figure 78.

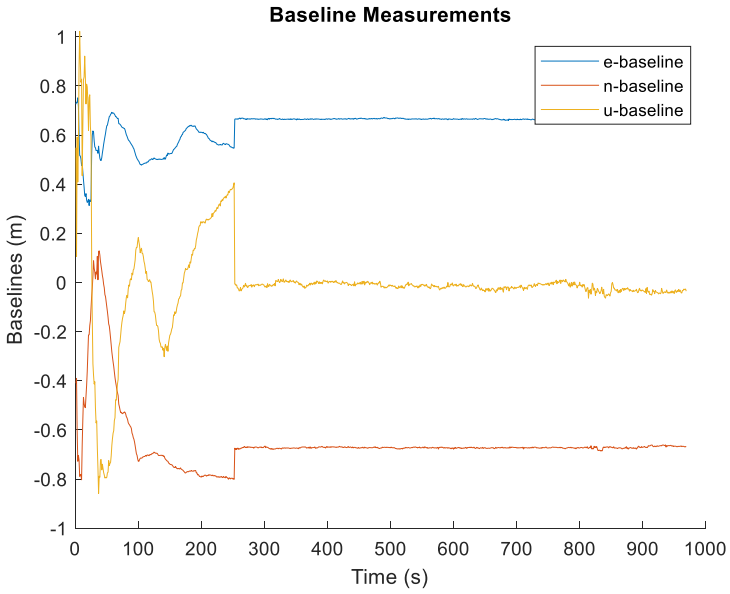

Figure 78. The baseline measurements of Dataset 7 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 79.

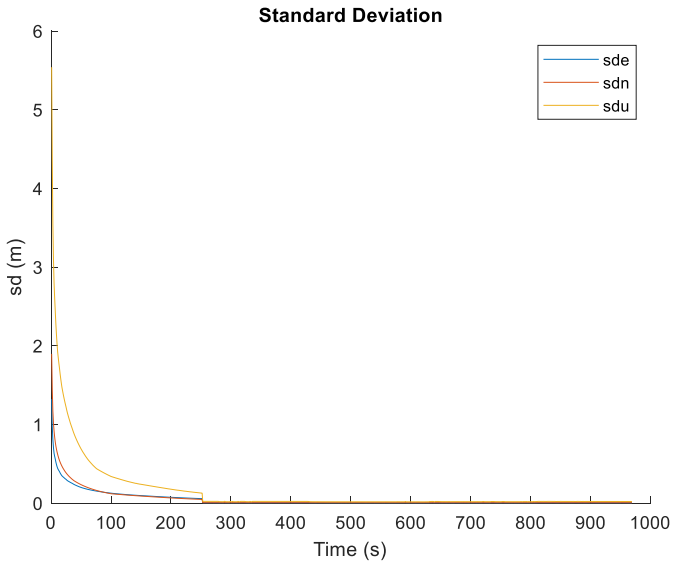

Figure 79. The standard deviation measurements of Dataset 7 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 80.

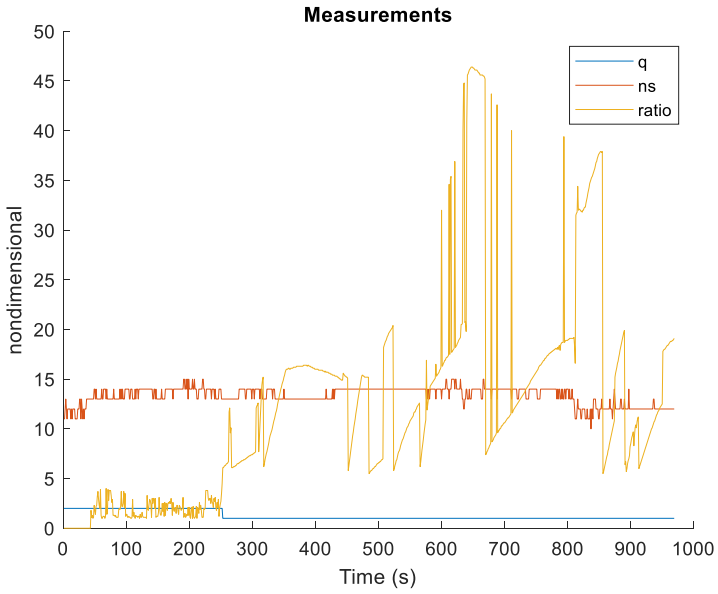

Figure 80. The quality measurements of Dataset 7 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 81.

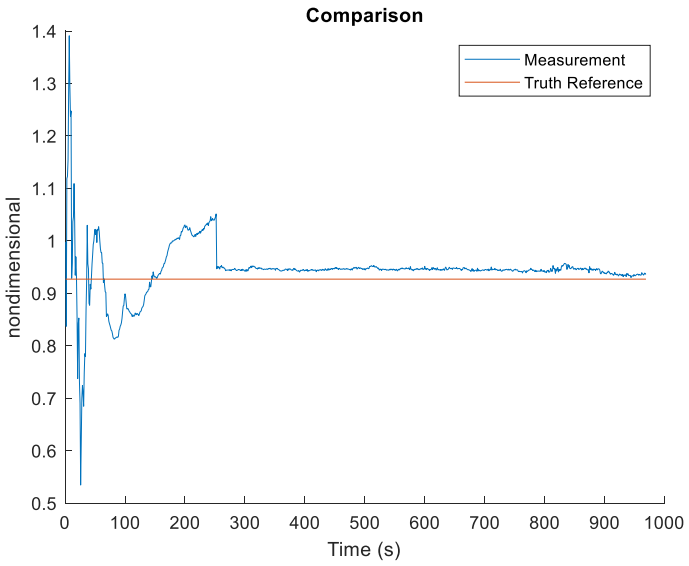

Figure 81. The distance measurements between receivers with a truth reference of Dataset 7 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 82.

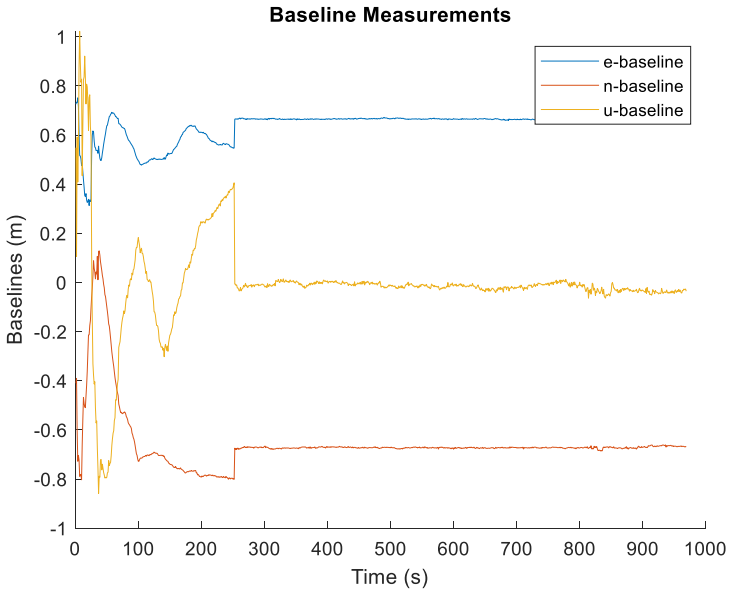

Figure 82. The baseline measurements of Dataset 7 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 83.

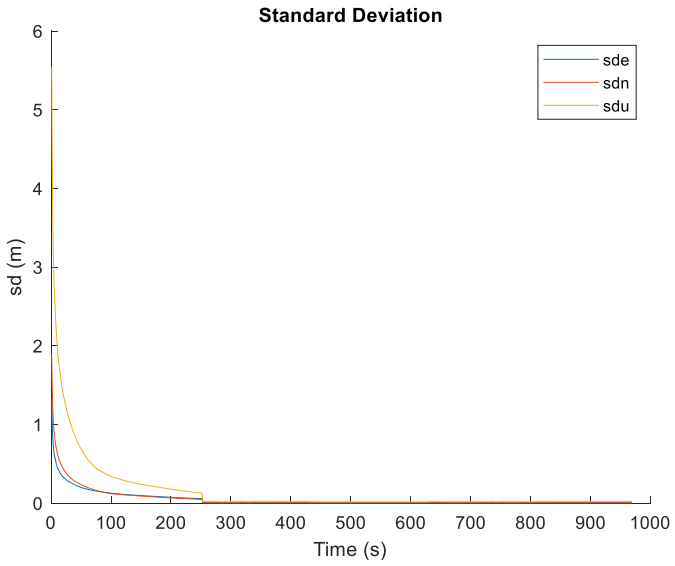

Figure 83. The standard deviation measurements of Dataset 7 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 84.

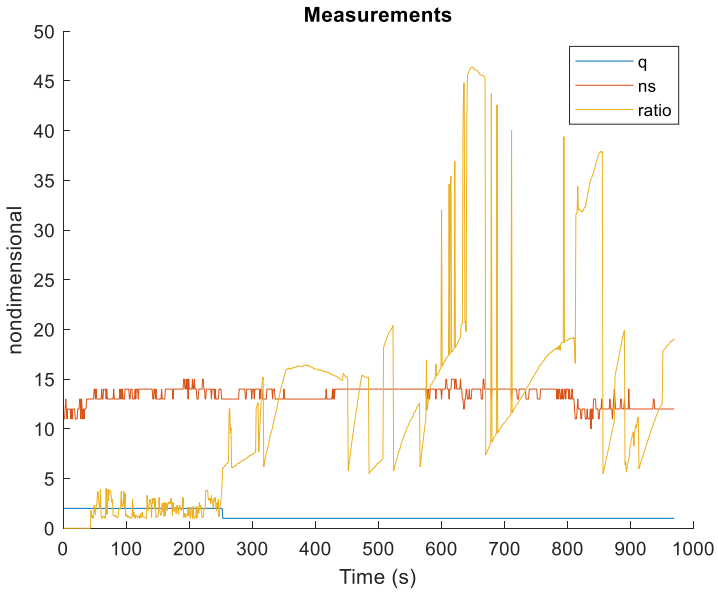

Figure 84. The quality measurements of Dataset 7 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 85.

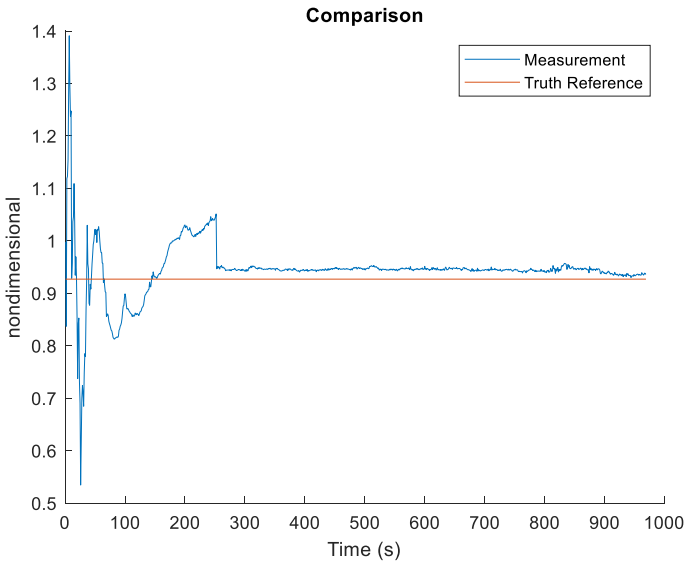

Figure 85. The distance measurements between receivers with a truth reference of Dataset 7 with RANSAC.

F.6 Dataset 8

The baseline measurements without RANSAC are shown below in Figure 86.

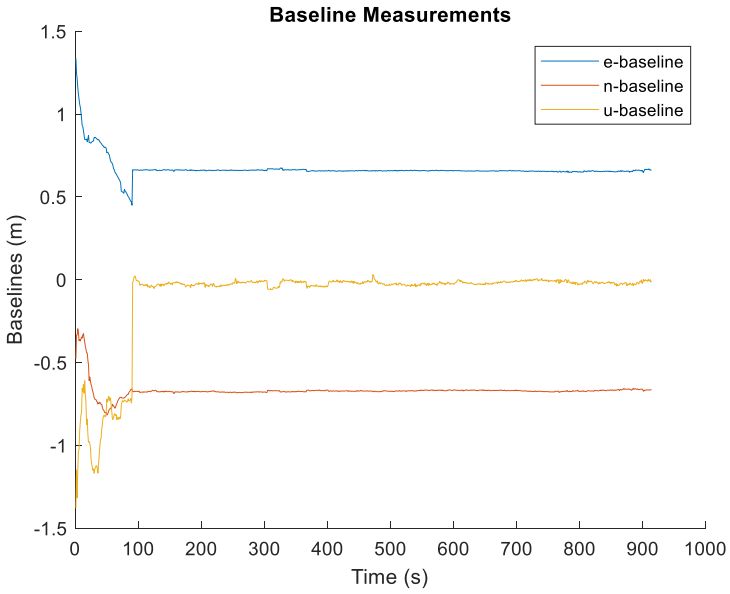

Figure 86. The baseline measurements of Dataset 8 without RANSAC.

The standard deviation measurements without RANSAC are shown below in Figure 87.

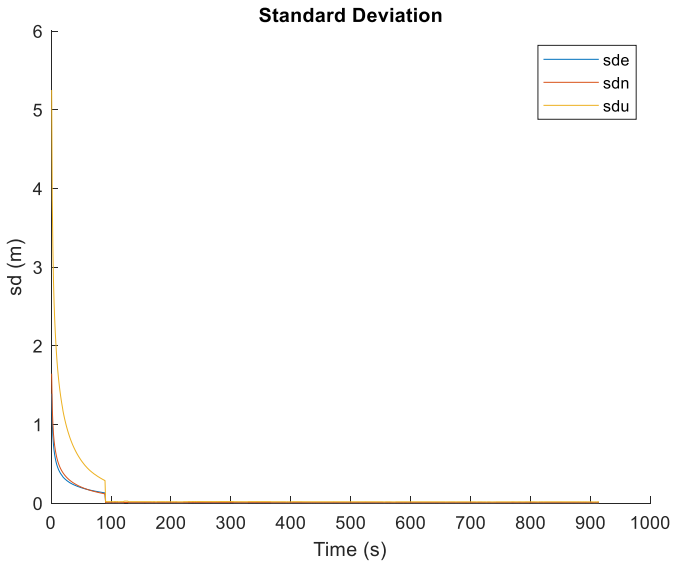

Figure 87. The standard deviation measurements of Dataset 8 without RANSAC.

The quality measurements without RANSAC are shown below in Figure 88.

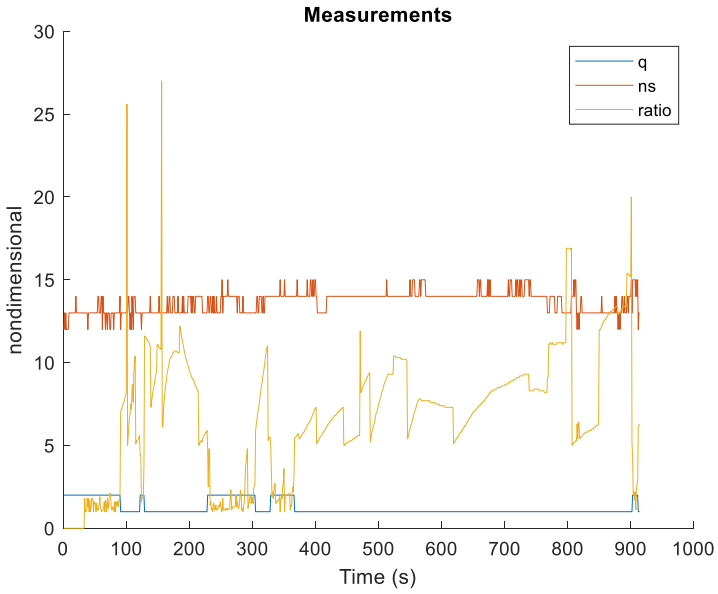

Figure 88. The quality measurements of Dataset 8 without RANSAC.

The comparison measurements of the distance between receivers with a truth reference without RANSAC are shown below in Figure 89.

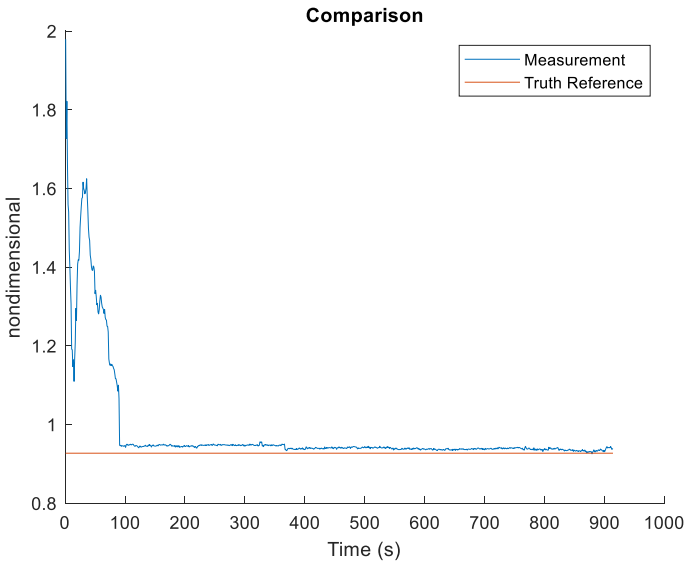

Figure 89. The distance measurements between receivers with a truth reference of Dataset 8 without RANSAC.

The baseline measurements with RANSAC are shown below in Figure 90.

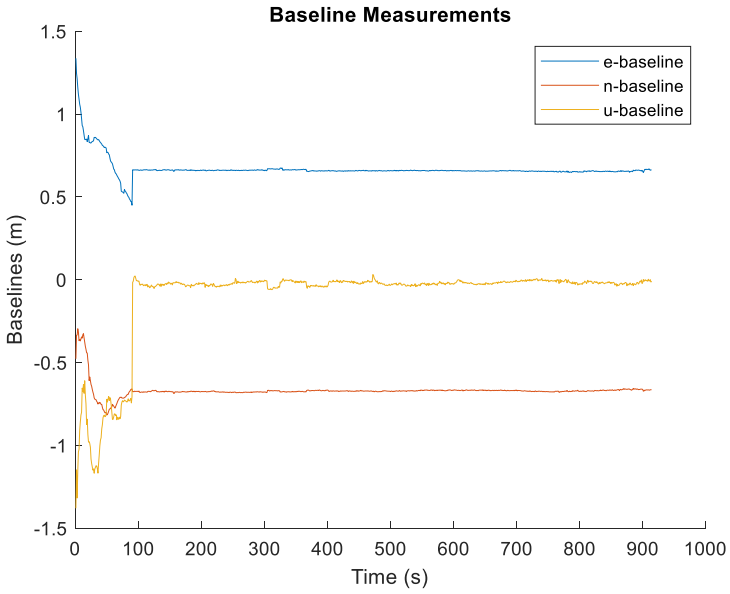

Figure 90. The baseline measurements of Dataset 8 with RANSAC.

The standard deviation measurements with RANSAC are shown below in Figure 91.

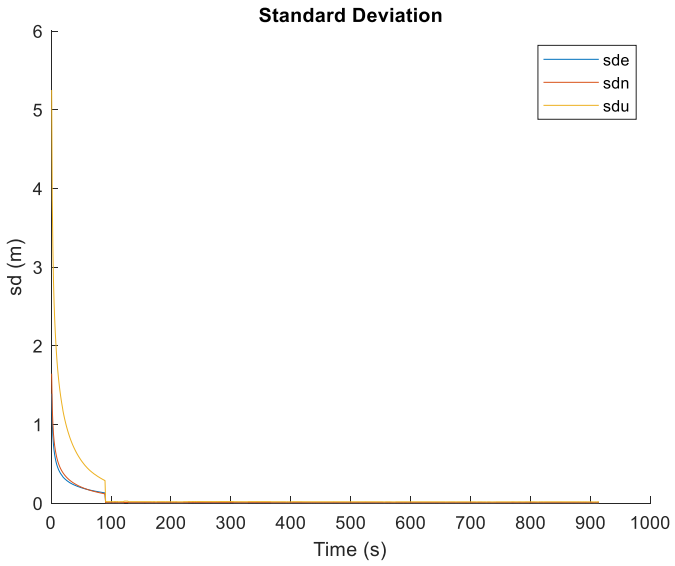

Figure 91. The standard deviation measurements of Dataset 8 with RANSAC.

The quality measurements with RANSAC are shown below in Figure 92.

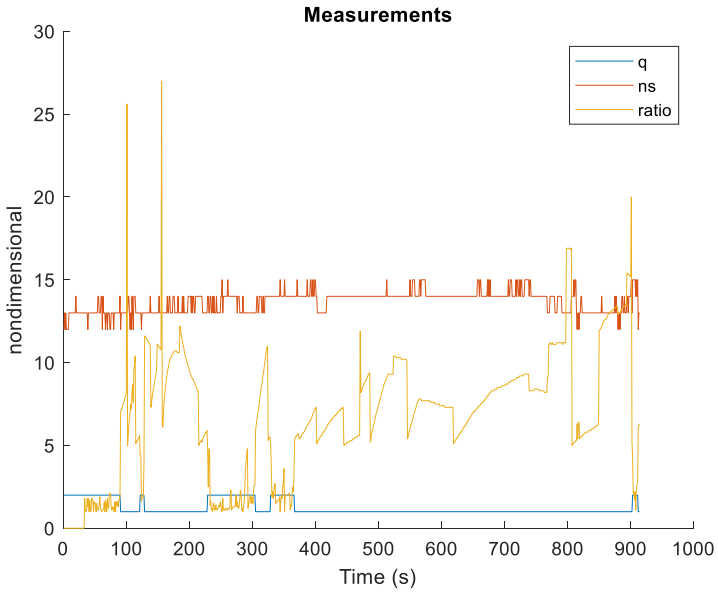

Figure 92. The quality measurements of Dataset 8 with RANSAC.

The comparison measurements of the distance between receivers with a truth reference with RANSAC are shown below in Figure 93.

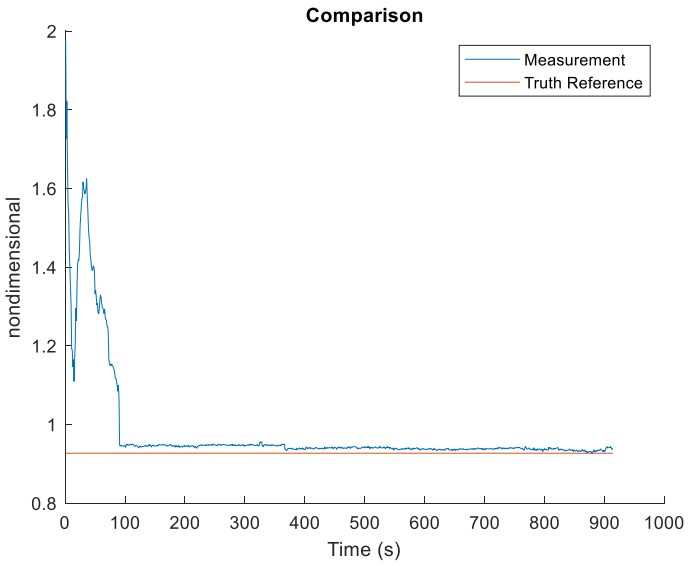

Figure 93. The distance measurements between receivers with a truth reference of Dataset 8 with RANSAC.