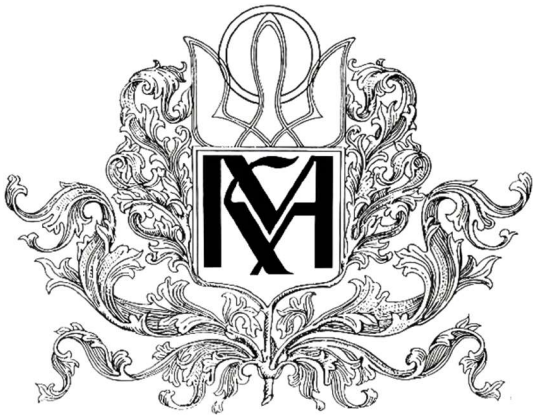
Ministry of Education and Science of Ukraine
National University “Kyiv-Mohyla Academy”

Informatics Department of Informatics Faculty



Emitter Position Estimation Using Time Difference of Arrival

Master Thesis
“Software Engineering” specialty (121)

Master’s thesis supervisor
Professor, doctor of technical science
Hlybovets A. M.

_____

(signature)

“____”_____ 2022

By student Musiiaka O. A.

“____”_____ 2022

Kyiv 2022

Table of Contents

Glossary

AoA	-	Angle of Arrival
CEP50	-	Circular Error Probable 50%
CPU	-	Central Processor Unit
CRLB	-	Cremer-Rao Lower Bound
CUDA	-	Compute-Unified Device Architecture
EM	-	Electro-Magnetic
FDoA	-	Frequency Difference of Arrival
GDOP	-	Geometric Dilution Of Precision
GNSS	-	Global Navigation Satellite System
GPS	-	Global Positioning System
GPU	-	Graphics Processing Unit
HDOP	-	Horizontal Dilution Of Precision
MLE	-	Maximum Likelihood Estimator
PDF	-	Probability Density Function
SIGINT	-	Signal Intelligence
SIMD	-	Single Instruction Multiple Data
SNR	-	Signal to Noise Ratio
TDoA	-	Time Difference of Arrival
ToA	-	Time of Arrival
UAV	-	Uncrewed Aerial Vehicle
WLS	-	Weighted Least Squares

Introduction

Most of the most critical technologies that became an indispensable part of our everyday lives originate from the military or related domains. Global navigation satellite systems are an example closely related to this work's topic; it shares most of the mathematical models and some of the computational algorithms with TDoA emitter localization methods due to the problem's duality. In GNSS systems single receiver determines the temporal offset between signals of cooperative satellite-based transmitters; a typical scenario of a dual problem usually involves multiple receivers in known locations and a non-cooperative transmitter whose position is to be determined [1].

Mathematical methods of TDoA emitter position estimation have been the subject of research for the last 70 years within military and civilian institutions. As a result, several methods with optimal solutions were developed [2], [3], and [4]. Nevertheless, practical aspects of determining non-cooperative emitter location lie far beyond solving a system of quadratic equations.

The key performance metric of emitter localization is accuracy, but even this metric is ambiguous when applied to real-world applications. Estimating just a most likely location is not enough; an estimator should also evaluate its accuracy through dispersion ellipse, CEP50, or provide computed likelihood distribution data based on input data quality. Hidden correlations of measurement errors in input data and unknown error model parameters make an estimation of result accuracy especially challenging, given that sensor's locations are usually determined by GNSS with its statistical properties of measurement error [5].

The most common application of TDoA location is military signal intelligence (SIGINT). That implies that non-cooperative emitters might be both stationary and moving. Therefore, data processing should allow efficient measurement accumulation for non-stationary emitters as well as non-stationary TDoA sensors (receivers) [6].

Doppler frequency offset created by the relative motion of transmitter and receivers might be considered a nuisance for the time-based approach. However, the amount of frequency shift carries additional information that could be used as an input parameter for the

estimator. This approach is known as frequency difference of arrival (FDoA) and can be combined with TDoA to improve accuracy and motion tracking [7].

Another often overlooked aspect of developing new algorithms for TDoA location systems is a paradigm shift in computational methods. The algorithms that were designed at the early stages were targeted for systems with low computational power and often had to use rough approximations and simplified models to obtain results within a reasonable time interval. These methods are often re-used in new systems without considering vastly increased computational power and the availability of massively parallel computing systems suitable for field deployment. The utilization of available computational power uncovers new mathematical approaches that were not practically feasible several decades ago [8].

Radiofrequency emitter localization used in the military domain to determine the geographical coordinates of the source is called geolocation. There are many other techniques for the geolocation of various targets. However, a proliferation of crewless aerial vehicles as a common surveillance platform, on the one hand, dictates stricter weight and size limitations compared to conventional aircraft; on the other hand, the lower cost of the platform allows using a larger number of vehicles in the area of interest. The traditional approach of triangulation using direction-finding results requires larger antenna systems and is usually oriented on manual data acquisition rather than automatic processing. Therefore, TDoA and FDoA methods of geolocation are attaining more interest as they are more suitable for automated surveillance data collection by UAV platforms [9].

Another shift in the paradigm of SIGINT operations is the time interval spent for result processing. Besides traditional offline monitoring, numerous scenarios for tactical geolocation have been discovered. The time between the signal source is localized, and actions taken can be as small as several seconds. This type of usage imposes stricter limitations on both data processing time and mathematical algorithm efficiency. The reason for the latter is that there is no time window for collecting a large number of noisy signal samples; the requirement for a monitoring solution is to achieve the best possible resulting accuracy while having a limited number of signal measurements [10].

The key difference between cooperative and non-cooperative radio frequency emitters is known vs. unknown signal structure. Cooperative emitters like GNSS satellites have specially crafted signals with good autocorrelation properties. At the same time, signals of non-cooperative emitters might range from narrowband voice communication to ultrawideband radar pulses [11].

The key challenge of applying the TDoA geolocation method to the variety of signals of interest is that it is ambiguous and difficult to measure the relative time offset of signals between different receivers. The most common technique of measuring the time difference of arrival is timestamping. It implies that receiver clocks are synchronized with each other with accuracy within several nanoseconds and require either auxiliary radio signals like GNSS or precision cesium clocks that were synchronized prior to operation [12]. The last but not least challenge is determining the location of receivers with the required accuracy, which might be trivial using GNSS and almost impossible for non-stationary receivers in GNSS-denied areas.

Overview of Geolocation Techniques

The most common techniques of locating radio frequency emitters currently in use fall into three categories: wavefront direction, time, and frequency difference between different spatial locations [13]. Historically, the first and the most straightforward way to locate the source of radio waves is to use the directional properties of receiving antenna to determine the direction of wavefront propagation; this method is called the angle of arrival or AoA. The measurements should be taken from at least two distinct spatial locations and estimate the source location using the intersection of resulting lines. If there are more than two lines, some aggregation technique is required to consolidate multiple intersection points; the least-squares method is usually used to obtain a more accurate result.

This method is simple and does not require sophisticated mathematical techniques to obtain results; however, it suffers several fundamental disadvantages. The most significant drawback is the need to use a specially designed directional antenna that should be mechanically rotated to take measurements [14]. Such antenna design is usually bulky and challenging to implement for a broad frequency range. The need for mechanical rotation worsens the situation, often resulting in manual measurement by an operator that introduces the possibility of human error and limits accuracy and the achievable number of measurements per time interval.

A more modern approach to measuring the angle of arrival is using coherent reception from multiple antennas and determining wavefront direction by the phase difference between channels. These methods include interferometry and synthetic aperture techniques and do not require operator involvement for taking measurements, therefore eliminating performance bottleneck.

However, wide-band multi-channel coherent receivers are much more complex and expensive than ordinary radio receivers; these devices require calibration to deliver the necessary performance over the operating frequency range. Antenna system complexity is even higher than in traditional AoA; antenna and cables must have precisely identical parameters and must be calibrated for phase offset [15]. Conventional directional antenna and interferometry AoA methods are sensitive to reflected signals commonly found in ground-based setups.

Time and frequency-based geolocation techniques involve several spaced-out surveillance platforms with single-channel receivers and precisely synchronized clocks. Depending on the particular usage scenario, the system could be ground-based or airborne. Frequency difference measurement applies only if either receiver or transmitter is airborne with sufficient velocity to produce a usable Doppler frequency shift. Because of this limitation, systems based solely on FDoA are rarely used in practice; frequency difference is usually used as supplementary information for TDoA-based measurements [16].

Among described geolocation methods time-based approach is the most challenging in terms of mathematical algorithms and computational power requirements. Nevertheless, this technique results in highly accurate geolocation systems with compact and inexpensive radio receiving hardware suitable for lightweight uncrewed aerial vehicles. TDoA systems rely on GNSS signal availability; therefore, they cannot operate in GNSS-denied areas. However, this limitation does not hinder common radio surveillance scenarios due to the significant distance from the potential jammer location. An inherent drawback of the

TDoA approach is the inability to operate with a continuous wave or narrow-band radio signal due to the lack of rapid signal variation that can be used to determine signal propagation delay. Another prerequisite for differential time measurement is the possibility of unambiguous matching time-spaced signal features. This problem is almost trivial for the coherent multi-channel receivers, but for TDoA, there is usually no communication bandwidth for sending the complete quadrature of the signal. Therefore, techniques like pulse descriptor encoding or signal feature extraction are used to match signal features between receivers [17].

In order to obtain an exact emitter location, the number of receivers needs to be $N + 1$ where N is the number of dimensions [18]. The problem can be stated either as a two-dimensional location on geoid if the emitter is presumably ground-based or three-dimensional for ground-based and airborne radio emitters. Even with a minimum required number of receivers, the system of equations might have non-unique solutions due to poor receiver constellation geometry. For example, putting three receivers on a single line or four receivers in the same plane will result in an ill-conditioned system of equations. For illustrative purposes, the solution to the two-dimensional geolocation problem can be represented as the point of intersection of two hyperbolas.

In the case of a two-dimensional problem with three sensors, three differential time measurements are required. However, only two of them can be linearly independent given a suitable constellation configuration; as a result, intersecting point of any two hyperbolas will be the same. The reasoning above assumes that all the receivers are synchronized to a common time source like a GPS clock; this is the most preferred measurement setup. However, only partial clock synchronization might be available in some systems. In the case of pairwise receiver synchronization, for example, when using the same hyperbolic system of equations for simultaneous multi-point AoA geolocation, the number of differential measurements in non-collinear directions must be larger or equal to the number of problem dimensions [19].

Although some entirely determined systems of hyperbolic equations have analytic solutions, they are rarely used in practice for several reasons. The solution for a two-dimensional problem can be found by finding roots of a fourth-degree polynomial, but in actual application approximating geoid surface with a 2-D plane might result in degraded accuracy. Solving three-dimensional problem variants with analytical methods seems to be infeasible up to date, and computational methods should be employed; the same applies to over-determined systems of hyperbolic equations when there are more sensors than minimally required [20]. The only way to efficiently combine information from extra sensors or multiple repetitive measurements is to employ computational methods that result in a statistically optimal solution.

The discrete likelihood method computes the likelihood score for the selected spatial points. It uses the likelihood ratio to compute the probabilistic distribution of the emitter location given a set of measurements that can be obtained from different sensors or the same set of sensors during the monitoring period. Sensors might be either moving or stationary while handling moving radio emitter geolocation is not straightforward and might require limiting observation time. The most notable advantage of the discrete likelihood method is the ability to spatially separate emitters with similar characteristics when the pre-processing algorithm fails to cluster input measurements correctly using only signal properties. The ability to combine different types of measurement within the single log-likelihood mathematical framework is another significant advantage of the discrete likelihood method [21].

Given the variety of mathematical methods with different applicability, advantages, and drawbacks, the choice of the specific approach depends both on the requirements imposed on the intended system and constraints like computational power budget, power consumption, development cost, and real-time vs. statistical data processing. There is no universal method for all possible applications; furthermore, the scope of the problem is also different depending on the availability of synchronized clocks at the receiving site, the presence of well-detectable structure in the signal of interest, and many other factors. Nevertheless, this work primarily focuses on the computational aspect of measurement data processing rather than on digital signal processing or data pre-processing. Signal timestamping, classification and clustering are challenging problems that are not in the scope of this work. Nevertheless, they impose implications on the geolocation techniques [22].

Source Location Estimation Problem Statement

Consider the scenario of M spatially separated sensors at locations $s_i, i \in 1,2,\dots,M$ that receive the signal from the source with an unknown position. Spatial location variables are Cartesian coordinates in the form of column vectors whose length corresponds to the number of problem dimensions; they can be two or three-dimensional, depending on the problem formulation. The distance between the source and each receiver is defined as a Cartesian coordinates difference vector. In the case of a non-cooperative emitter, the actual time of the signal emission at the transmitter antenna is unknown; therefore, we operate with time difference values as the input data. In order to simplify notation, it is convenient to consider one of the receivers as a reference point by setting its absolute time value 0. Although this simple approach should be taken with care, using a single reference sensor results in an unfavorable error correlation matrix that reduces estimate accuracy.

The propagation velocity of electromagnetic waves is usually considered constant and independent of the propagation direction. This assumption is a simplification based on the line-of-sight electromagnetic wave propagation model. In the natural environment, various phenomena result in deviations from that simple model but introduced inconsistencies are usually small enough to be omitted from the error model. Isotropic EM wave propagation model assumption allows further simplification by replacing time difference with a pseudo-range difference, where pseudo-range is the distance that electromagnetic wavefront travels within the time difference duration. Because the time difference is relative to the reference sensor, the pseudo-range difference is defined as:

$$r_{i1} = r_{i1}^0 + n_{i1}, i = 2,3,\dots,M$$

where r_{i1}^0 is the true pseudo-range difference, and n_{i1} is the additive measurement noise. The true value of TDoA pseudo-range difference is defined as the difference of vector norms of distances between the signal source and each of the receivers, where the second receiver is the reference one:

$$r_{i1}^0 = \|u^0 - s_i\| - \|u^0 - s_1\|$$

Given the variety of TDoA geolocation techniques, it is reasonable to start with the most straightforward approach to establish mathematical terms and notions for the problem statement. As the first approximation, we ignore the statistical nature of the estimation problem, and all the errors present in measurements and focus on obtaining the geolocation problem solution. However, estimator performance evaluation requires accounting for all the measurement errors and their correlation to verify a given method against Cramer-Rao Lower Bound. For the scope of this paper, timing measurement error and sensor location uncertainty are considered as the primary limiting factors of geolocation algorithm accuracy. The sensor location is typically measured using GNSS and is prone to significant temporal autocorrelation errors.

The illustration below demonstrates the difference between ToA and TDoA problem statements in two-dimensional space. In the case of ToA formulation, the emitter location is determined as the intersection point of circles around each sensor; the circle radius is a measured signal propagation delay between the emitter and a corresponding sensor; it is a known input data. In TDoA formulation, signal propagation time is not known from the input measurements; the only known information is the difference between propagation delays between pairs of sensors. In this case, the equal difference curves are not circles but hyperbolas, and the emitter is located at their intersection.

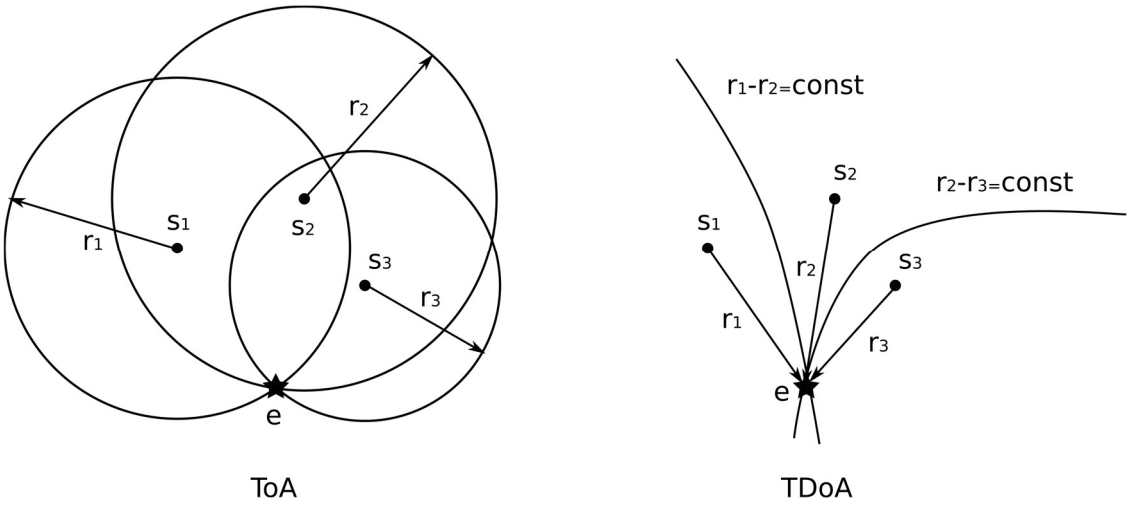


Fig. 1. Two-dimensional ToA versus TDoA geolocation problem illustration

The simplest problem statement shown above has an analytical solution where emitter location is calculated as roots of 4-th degree polynomial. In practice, however, measurements are inaccurate and might even be unreliable (i.e., pulse cluster misclassification). Required location accuracy is achieved by statistical data processing and using multiple redundant sensors.

To sum up, TDoA geolocation problem solution is the coordinates of the most probable emitter location given time difference measurements and sensor position estimates. Multiple measurements and information from redundant sensors should be used in a statistically optimal way to improve location accuracy.

Measurement Error and Estimator Performance

In order to use statistical methods, several assumptions should be made about measurement errors and their statistical distribution. The two primary error sources are timing measurement error and sensor location inaccuracy. The latter might be less significant for stationary sensors; however, a typical TDoA application scenario includes airborne sensors where location error cannot be ignored.

First, we consider timing measurement error; it consists of two components: clock synchronization error and pulse feature detection error. Clock synchronization error is highly correlated between consecutive measurements due to the integrative properties of clocks themselves; random oscillator frequency deviations result in clock random walk. A typical reference clock source includes a GNSS-disciplined oscillator as a part of the Kalman filter; the autocorrelation time constant is usually much larger than the time interval between consecutive measurements but might be smaller than the observation time. That is, accumulating consecutive measurements do little to no effect on reducing the effect of clock random walk inaccuracy.

Pulse feature detection timing error is caused by a non-ideal signal-to-noise ratio at the sensor receiver and possibly the implementation of the feature detector itself and finite precision of the timestamp information coding. Nevertheless, this type of timing error does not exhibit a correlation between consecutive measurements, and its impact on geolocation accuracy can be effectively reduced by using multiple measurements.

Sensor location errors exhibit similar autocorrelation properties to clock errors since they are both based on the GNSS system and usually the same Kalman filter. In order to simplify the analysis, sensor location error is considered independent from clock error. The actual measurement error value does not directly impact the estimator performance; only the dispersion value proportion between sensors and measurement samples is important since data should be weighted appropriately to approach CLRB-predicted performance.

Nevertheless, in the most practical applications, the single-point output is not sufficient; adequate measure is needed to estimate the result's uncertainty. Fisher information matrix and corresponding uncertainty ellipsoid are good indicators of the algorithm's result quality. Given a simplifying assumption regarding error normal distribution and projection linearization Fisher information matrix can be computed using the approximated second derivative of the log-likelihood function.

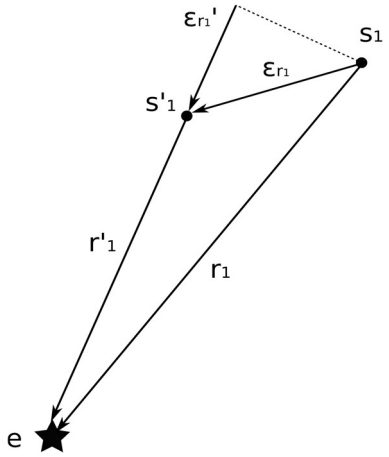


Fig.2. Sensor location error impact on pseudo range measurement

As a first approximation, let us consider the error contribution to a single measurement and keep the error correlation between measurements for further analysis. Fig. 2 illustrates the impact of sensor location error on pseudo-range measurement, where r_1 is the true range between the first sensor and the emitter, ${r_1}'$ is measured range in the presence of sensor location error, ε_{r_1} is location error offset vector, and ${\varepsilon_{r_1}}'$ is location offset error vector projection on the error-affected r_1 estimate. When $r_1 \gg \varepsilon_{r_1}$ the projection of the offset vector is close to the original offset ${\varepsilon_{r_1}}' \approx \varepsilon_{r_1}$, thus we can substitute ${\varepsilon_{r_1}}'$ for ε_{r_1}without accuracy loss in the most practical applications where location error is much smaller than the distance to the emitter.

Another important aspect regarding measurement errors is the necessity to consider sensors pair-wise due to the differential nature of the TDoA method. Using a single reference sensor will introduce a significant error correlation between differential measurements. This problem could be addressed by introducing a weighting matrix into a system of linear equations. This approach is convenient for the weighted least squares method.

However, it is much less suitable for methods like log-likelihood function gradient descent that do not use matrix inversion on each iteration step. In this paper alternative approach for taking differential measurements is used; instead of using a single reference sensor, a circular round-robin difference is used. Since this method does not break measurement weighting symmetry, the weighting matrix can be replaced by the scalar multiplier without loss of estimator accuracy.

The Scope of this Work

The goal of this work is to create a working program capable of estimating emitter position location using algorithms appropriate for practical applications. The program design aims to exploit massive parallel computing using the SIMD model to process multiple measurements from multiple sensors with real-time performance. CUDA and OpenCL are highly efficient technologies for accelerating vector and matrix operation on commercially available hardware and thus will be used for parallelizing TDoA geolocation algorithms in this paper.

The first step is the choice of the algorithms for implementation. Existing methods like linearization [23], log-likelihood gradient descent [24], and iterative least-squares [25] are possible candidates. The first part of this paper focuses on the basic implementation of the TDoA algorithms mentioned above in a single-threaded environment and their comparison based on convergence speed and achieving CRLB performance under appropriate conditions.

Besides the algorithm itself, different numerical optimization techniques will be considered for performing the actual computation; matching the appropriate optimizer with the selected geolocation technique can improve overall computation performance, reduce hardware requirements, and improve system capabilities.

The benchmarking and computational process analysis results will be used to choose the best suitable estimator for the parallelized implementation. For the sake of implementation simplicity, offline statistical data processing is considered; however, the selected algorithm design should be compatible with online data processing when the internal state of the estimator accumulates information from the incoming measurement sequence in real-time.

The next step is modeling practical geolocation scenarios for algorithm performance assessment; the result is the algorithm for parallel implementation. After the parallelized version is implemented, it will be compared with the single-threaded version algorithm in terms of data throughput; possible bottlenecks and optimization opportunities will be analyzed.

Taylor Series Linearization and Spherical Interpolation

This geolocation estimator described in [2] is the only non-iterative method considered in this work. Despite some limitations in sensors type and configuration, this method provides a reasonable initial estimate for iterative methods and can overperform methods like spherical interpolation under low error conditions. Unlike iterative methods, this approach provides a closed-form solution for a hyperbolic position fix; it is valid for both close and distant emitter locations. The method solution approximates the maximum likelihood estimator when the measurement error is small. For two-dimensional geolocation problem formulation using four TDoA sensors provides a solution in closed form.

Using four sensors for a two-dimensional problem allows converting the solution into a system of linear equations and solving it using a weighted least-squares method. This result can be extended to three dimensions using four sensors for closed-form solution and five TDoA sensors for a system of linear equations that can be solved by WLS. Unfortunately, despite many advantages, this single-step method cannot be used to solve a two-dimensional geolocation problem when the emitter is located on the geoid surface.

To obtain the solution, let the be source at the unknown position (x, y) and the sensors are located at positions (x_i, y_i) and the squared distance between the emitter and each of the sources is

$$r_i^2 = (x_i - x)^2 + (y_i - y)^2 = K_i - 2x_i x - 2y_i y + x^2 + y^2, i = 1,2, \dots, M$$

$$\text{where } K_i = x_i^2 + y_i^2 \quad (1)$$

Given c as wavefront propagation velocity distance difference between two sensors

$$r_{i,1} = cd_{i,1} = r_i - r_1 \quad (2)$$

Solving the resulting system of non-linear equations is challenging. One of the possible ways is linearizing the equation using Taylor-series expansion and solving it iteratively [26].

With a set of pseudo-range measurements $d_{i,1}$the method starts with an initial emitter location guess (x_0, y_0) and computes position deviations on each iteration

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = (G_t^T Q^{-1} G_t)^{-1} G_t^T Q^{-1} h_t$$

Where matrix $Q$is the measurement covariance matrix, and definitions of h_t and G_t are

$$h_t = \begin{bmatrix} r_{2,1} - (r2 - r1) \\ r_{3,1} - (r3 - r1) \\ \cdots \\ r_{M,1} - (r_M - r1) \end{bmatrix} G_t = \begin{bmatrix} (x_1 - x)/r_1 - (x_2 - x)/r_2 & (y_1 - y)/r_1 - (y_2 - y)/r_2 \\ (x_1 - x)/r_1 - (x_3 - x)/r_2 & (y_1 - y)/r_3 - (y_3 - y)/r_3 \\ \cdots & \cdots \\ (x_1 - x)/r_1 - (x_M - x)/r_M & (y_1 - y)/r_1 - (y_2 - y)/r_M \end{bmatrix}$$

On each consecutive iteration values x_0 and y_0 are updated as follows: $x_0 = x_0 + \Delta x y_0 = y_0 + \Delta y$. The iterative process continues until update values are sufficiently small. This method requires starting point to be close to the solution; Otherwise, the estimator can diverge, and the estimator convergence is not guaranteed.

Alternatively, [15] and [16] propose transforming equation (2) to a different system of equations. Equation (1) can be rewritten as follows:

$$r_{i,1}^2 + 2r_{i,1}r_1 + r_1^2 = K_i - 2x_i x - 2y_i y + x^2 + y^2 \quad (3)$$

Subtracting (1) at $i = 1$ from (3) result in

$$r_{i,1}^2 + 2r_{i,1} = -2x_{i,1}x - 2y_{i,1}y + K_i - K_1 \quad (4)$$

The resulting system of equations is linear regarding x, y and r_1. The unknown emitter location (x, y) can be computed by eliminating r_1 from (4) [5]; this step results in $M - 2$ linear equations that can be solved using a least-squares method. An alternative solution is proposed by [27] where x and y are solved in terms of r_1. The intermediate result is substituted back into (4) and produces a system of equations where the only unknown variable is r_1. Substituting the computed value of r_1 that minimizes the least-squares equation error to the intermediate result gives the final emitter location estimate. This method is known as spherical interpolation [28], and it is mathematically equivalent to the method described in [15].

The main disadvantages of the methods described above are non-optimal performance relative to CRLB and the necessity of using a weighting matrix that is challenging to determine. An improved version of the estimator for three and four sensors' constellations in two-dimensional problem formulation is proposed by [26]; These two cases are addressed separately because the advantage of extra sensors allows avoiding solving polynomial matrix equation.

Chan-Ho Hyperbolic Location Estimator

The solution for three sensors in two-dimensional space requires finding polynomial roots. When sensors are located in an unfavorable manner relative to the emitter or measurement error is large enough to cause the polynomial to have two positive roots, the result becomes ambiguous. It should be noted that this undesired behavior is not a property of this particular method but applies to the system of hyperbolic equations themselves.

The proposed method for three sensors solves an unknown sensor location (x, y) in terms of r_1 using equation (4)

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\begin{bmatrix} x_{2,1} & y_{2,1} \\ x_{3,1} & y_{3,1} \end{bmatrix}^{-1} \times \left\{ \begin{bmatrix} r_{2,1} \\ r_{3,1} \end{bmatrix} r_1 + \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \end{bmatrix} \right\} \quad (5)$$

Placing the intermediate result into the equation (1) at $i = 1$ results in the quadratic equation in terms of r_1. The positive root of the equation should be substituted back into (5) to obtain the final result. This method is mathematically equivalent to [16].

When the quantity of sensors is four or more $(M \geq 4)$, the system of equations for the two-dimensional problem becomes overdetermined. As a result of measurement errors, the system of equations (4) will not meet at a single point, and the location estimate is the best fit that minimizes error. Let $m_a = \left[m_p, r_1\right]^T$ be the unknown vector, and $m_p = [x, y]^T$; in the presence of the measurement noise, the error vector can be derived from (4)

$$\psi = h - G_a m_a^0 \text{ where}$$

$$h = \frac{1}{2} \begin{bmatrix} r_{2,1}^2 - K_2 + K_1 \\ r_{3,1}^2 - K_3 + K_1 \\ \cdots \\ r_{M,1}^2 - K_M + K_1 \end{bmatrix} \quad G_a = -\begin{bmatrix} x_{2,1} & y_{2,1} & r_{2,1} \\ x_{3,1} & y_{3,1} & z_{3,1} \\ \cdots & \cdots & \cdots \\ x_{M,1} & y_{M,1} & r_{M,1} \end{bmatrix}$$

With simplifying assumptions regarding error vector distribution error covariance matrix can be computed using the formula:

$$\Psi = E[\psi\psi^T] = c^2 BQB$$

where $B = diag\{r_2^0, r_3^0, r, \ldots, r_M^0\}$ and Q is the covariance matrix of pseudo-range measurements. The resulting system of equations is still non-linear in terms of x and y.

One of the approaches to solve the system of non-linear equations is to ignore the existing relationship between variables$x$, y, and r_1. This assumption allows solving the system using the least-squares method in two steps. The two-step procedure approximates the maximum likelihood estimator, and it is mathematically equivalent to the following equation:

$$m_a \approx (G_a^T Q^{-1} G_a)^{-1} G_a^T Q^{-1} h$$

The solution to least squares assumes that variables x, y, and r_1 are independent, but in the actual equation, they are dependent when $i = 1$. Let us consider incorporating this relationship into the estimator to improve the resulting accuracy. The elements of m_a can be expressed as follows:

$$m_{a,1} = x^0 + e_1, m_{a,2} = y^0 + e_2, m_{a,3} = r_1^0 + e_3$$

Where e_1, e_2, and e_3 are estimation errors of m_a. Subtracting the first two components of m_a by x_1 and y_1 results in another system of equations

$$\psi' = h' - G'_a m'^0_a \text{ where}$$

$$h' = \begin{bmatrix} (m_{a,1} - x_1)^2 \\ (m_{a,2} - y_1)^2 \\ m_{a,3}^2 \end{bmatrix} G'_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} m'_a = \begin{bmatrix} (x - x_1)^2 \\ (y - y_1)^2 \end{bmatrix}$$

This approximation is valid for small values of error components e_i; under these conditions, the solution will be reasonably close to the maximum likelihood estimator. The estimate of m_a can be obtained by the formula:

$$m'_a = (G'^T_a \Psi'^{-1} G'_a)^{-1} G'^T_a \Psi'^{-1} h' \text{ where}$$

$$\Psi' = [\psi' \psi'^T] = 4B' cov(m_a) B' \quad B' = diag\{x^0 - x_1, y_0 - y1, r_1^0\}$$

The matrix Ψ' is not known but B' can be approximated using values of m_a and G_a^0. The final position estimate is obtained from computed m'_a in two ways:

$$m_p = \sqrt{m'_p} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \text{ or } m_p = -\sqrt{m'_p} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

The actual location is determined by the region of interest, which makes estimation of the unknown signal source ambiguous, and it constitutes a disadvantage of this method.

Maximum Likelihood Estimation as an Optimization Problem

The most versatile approach that does not impose additional constraints on the input data is stating the geolocation problem as an optimization problem where unknown variables are emitter coordinates. In this case, the optimization criterion maximizes the likelihood of obtaining given measurements under a specific hypothesis, where the hypothesis is the emitter location. The most challenging part of this approach is handling measurement interdependencies that can be expressed in a covariance matrix. Even under the assumption that the measurement error of each sensor is independent of other sensors, the differential nature of the TDoA method requires using pairs of sensors for a pseudo-range estimate. In this case, the error covariance matrix will not be a trivial diagonal matrix.

Nevertheless, choosing a particular way of taking differential measurements between sensors can mitigate the overhead of using a non-trivial error covariance matrix. Under specific conditions, the error covariance matrix can be replaced by a scalar coefficient. Since the scaling coefficient applied to the likelihood function does not impact the position of the maximum, it can be simply omitted during the optimization stage. It will be shown that estimating positioning accuracy can be made after finding the maxima. From the mathematical perspective, the point of the maximum likelihood is also the maximum of the emitter location probability density function since probability density is proportional to the hypothesis likelihood. This can be derived from the duality of likelihood and parameterized random value distribution function:

$$L(\theta|x) = f(x|\theta) \quad (6)$$

where x is the observed outcome of the experiment; in this particular case is a set of measurements and θ is a hypothesis regarding the parameterized model, where the model consists of the wavefront propagation model and the hypothesis regarding the emitter location. L is a likelihood function and f is a parameterized probability density function. Since the definition of the maximum likelihood estimator is

$$\theta_{ML} = argmax_{\theta} L(\theta|x) \quad (7)$$

where θ_{ML} is a parameter set estimate by a maximum likelihood estimator based on the measurements x.

From (6) and (7) it can be shown that the MLE estimate is also the maximum for probability density function for the true model parameters

$$\theta_{ML} = argmax_{\theta} f(x|\theta) \quad (8)$$

since for the particular experiment $x = const$ and wavefront propagation model is fixed the parameterized probability density function depends only on the hypothesis θ about the true emitter location. Under these considerations (8) implies that θ_{ML} is also a probability density maximum for the emitter location.

Re-normalizing parameter space likelihood to cartesian coordinates of the geolocation is challenging and not required for the practical applications; however, such a normalization coefficient always exists and implies that the ratio of the likelihood function and probability density function for two hypotheses θ and θ' is the same

$$\frac{L(\theta|x)}{L(\theta'|x)} = \frac{f(x|\theta)}{f(x|\theta')} \quad (9)$$

The ratio (9) simplifies normalization for the log-likelihood function; furthermore, it might be treated as a log-probability ratio where the logarithm of hypothesis likelihood is the logarithm of probability density function plus the normalization constant.

Using a maximum likelihood estimator requires stating assumptions regarding the statistical distribution of measurement errors, cross-correlation, and autocorrelation. For the scope of this work, two sources of measurement errors are considered: timing measurement errors and the error of sensor location estimate. Autocorrelation of measurement errors reduces the amount of new information the estimator receives with each new measurement and reduces the resulting estimation accuracy.

The actual autocorrelation function is unknown for the most practical applications because parameters of Kalman filter in GNSS receiver are not known and not even guaranteed to be constant; these parameters can change under different receiving conditions like SNR, HDOP, and GDOP. The simplest way to compensate for autocorrelation properties is to add a “dilution of precision” coefficient when estimating the estimate accuracy. It will be shown that these simplifications do not impact maximum likelihood estimator performance.

Unlike autocorrelation measurement errors, cross-correlation has an impact on MLE performance, and it might be easily overlooked when comparing performance with Cramer-Rao lower bound. An unfavorable differential measurement scheme can reduce the available amount of information by using the same reference sensor in all pairs of measurements. Methods like WLS use a weight matrix to compensate for the impact of differential measurements; this approach is inconvenient with MLE. For the sensor location error, the three-dimensional normal distribution model is considered as

$$D_s = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}$$

it corresponds to the typical GNSS location error model where an equal PDF surface is an ellipsoid elongated along the vertical axis. It is beneficial to account for the actual values of GDOP and HDOP reported by GNSS system since different sensors might have different receiving conditions; accounting for the actual variance estimate instead of using a predefined constant will increase system robustness and the resulting estimate accuracy.

The statistical distribution of the timing measurement error is not always a Gaussian normal distribution. When using a sequence of measurements to improve the estimation accuracy, the timing measurement error component that corresponds to the contribution of finite SNR of the signal received at sensors usually does not exhibit any autocorrelation. Therefore, when accumulating multiple measurements, the clock synchronization error will quickly become dominant because of its strong autocorrelation properties.

For the analysis, the distribution of clock random walk is considered a normal Gaussian distribution. Therefore, assuming that timing measurement error has normal distribution can impact single-shot measurements to some extent while not degrading MLE performance for the accumulated series of measurements due to shifting the dominant error source from signal feature detecting to clock synchronization errors.

The model for the maximum likelihood estimator consists of known sensor locations and errors introduced by timing measurement inaccuracy and sensor's location uncertainty. Since MLE deals only with the hypothesis that corresponds to the emitter location, it is convenient to consider that sensors are located in their estimated locations, and inaccuracy is accounted for in the variance of pseudo-range measurement:

$$d_{i,j} = d_i - d_j = c\left(t_i - t_j + \tau_i - \tau_j\right) \quad (10)$$

where $d_{i,j}$ is measured pseudo-range difference between i-th and j-th sensors, c is a wavefront propagation velocity, t_i and t_j are signal propagation delays to the corresponding sensors, τ_i and τ_j are timing measurement errors. This is the first part of the measurement model; it reflects the physical phenomenon itself and accounts for the timing measurement inaccuracy. The second part of the model reflects the geometry of the sensor's constellation, and their location estimate inaccuracy:

$$\left\|\bar{r}'_{i,j}\right\| = \|\bar{r}_i\| - \left\|\bar{r}_j\right\| + \bar{n}_i \frac{\partial\|\bar{r}_i\|}{\partial\bar{s}_i} - \bar{n}_j \frac{\partial\|\bar{r}_j\|}{\partial\bar{s}_j} \quad (11)$$

where $\left\|\bar{r}'_{i,j}\right\|$ is an estimate of the geometric range-to-emitter difference between i-th and j-th sensors, $\bar{r}_i,\ \bar{r}_j$ true range vectors to the emitter from the corresponding sensors, $\bar{n}$ is the location error of the sensor, and $\frac{\partial\|\bar{r}\|}{\partial\bar{s}}$ is the derivative of range-to-emitter as a function of the sensor position that can be approximated using projection of unit vectors aligned with the coordinate system onto the corresponding vector between the emitter and a particular sensor; this vector is not known a priori, and it is a part of the measurement model under a particular hypothesis regarding the emitter location:

$$\frac{\partial\|\bar{r}_i\|}{\partial\bar{s}_i} \approx \frac{\bar{r}_i}{\|\bar{r}_i\|} \quad (12)$$

Combining (10), (11), and (12) results in a basic MLE model for a single differential pseudo-range measurement where the measured timing values are on the right, and the sensors' location and the hypothesis regarding the emitter location are implicitly represented by the corresponding $\bar{r}_i$ vectors on the right:

$$c\left(t_i - t_j\right) = c\left(\tau_i - \tau_j\right) + \|\bar{r}_i\| - \left\|\bar{r}_j\right\| + \bar{n}_i \frac{\partial\bar{r}_i}{\partial\|\bar{r}_i\|} - \bar{n}_j \frac{\partial\bar{r}_j}{\partial\|\bar{r}_j\|} \quad (13)$$

Considering normal distribution and the independent nature of the random variables, let us approximate their sum as a normal distribution with dispersion equal to the sum of dispersions of all random variables. These dispersions can be computed on a per sensor basis to simplify computations, and then dispersion contributions from the corresponding sensors can be added when performing differential signal propagation timing measurements.

The maximum likelihood estimator usually uses a log-likelihood function to reduce the range of numeric values and replace a product with an adduct of the contributing components. Since the product of the likelihood function can be used only for independent measurements, the covariance matrix for the differential measurements should be computed. The measurements themselves are considered independent; their covariance matrix is an identity matrix E, while the resulting covariance matrix C depends on the particular way of combining measurements from the sensors. The necessity of this operation stems from the differential nature of TDoA method and cannot be avoided due to unknown signal emission time at the emitter.

Most papers, including [2], [13], and [14] use the first sensor as a reference for all differential measurements. The differential measurements are defined in the following way

$$d_i = c(t'_{i+1} - t'_1), i \in 1..n-1$$

In this case, the covariance matrix is

$$C = \begin{bmatrix} 1 & 0.5 & 0.5 & ... & 0.5 \\ 0.5 & 1 & 0.5 & ... & 0.5 \\ 0.5 & 0.5 & 1 & ... & 0.5 \\ ... & ... & ... & ... & ... \\ 0.5 & 0.5 & 0.5 & ... & 1 \end{bmatrix} \quad (14)$$

For simplicity, the dispersion values are considered the same for all the sensors, and the actual scalar multiplier is not included in the definition of the covariance matrix. As the matrix suggests, all the differential measurements have covariance because of the shared reference sensor. As an alternative solution, instead of using a single reference sensor, the difference with circular index shift can be performed:

$$d_i = c(t'_i - t'_{i-1}) \quad i \in 2..n; \quad d_1 = c(t'_1 - t'_n) \quad i = 1$$

It results in a tri-diagonal square matrix whose dimensions are larger than (14) by 1 due to the larger number of the output values:

$$C = \begin{bmatrix} 1 & -0.5 & 0 & \dots & 0 \\ -0.5 & 1 & -0.5 & \dots & 0 \\ 0 & -0.5 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \quad (15)$$

The matrix (15) has the advantage of treating all the sensors in the same manner, ensuring that they have the same contribution to the result. Handling of sensors with different estimated dispersion is also simpler because each sensor measurement is used exactly in two differential measurements.

The log-likelihood function is a convenient transform for the exponential family of statistical distribution laws, including normal distribution. A logarithm eliminates the exponent leaving a second-degree polynomial that can be computed more efficiently than the original exponent; no exponentiation or taking a logarithm is needed for the actual computational step of the MLE. The log-likelihood function for normal distribution:

$$l(\bar{x}, \bar{\mu}, \bar{\sigma}) = -\frac{nln(2\pi)}{2} - \sum_{i=1}^{n} \left(\frac{nln(\sigma_i^2)}{2} + \frac{(x_i - \mu_i)^2}{2\sigma_i^2} \right) \quad (16)$$

The goal of the maximum likelihood estimator is to find such a hypothesis about the emitter location that maximizes (16); for that reason, all the parts of the formula that do not depend on the hypothesis can be omitted since we are interested only in finding the stationary point. In order not to confuse the simplified version with the true log-likelihood function, it will be designated as l_s:

$$l_s = -\sum_{i=1}^{n} \frac{\left((d_i - d_j) - (\|\bar{r}_i\| - \|\bar{r}_j\|) \right)^2}{2\left(\sigma_i^2 + \sigma_j^2\right)} \quad (17)$$

In the parameter space of the emitter-sensor distance, the shape of the log-likelihood function is a parabola with the branches facing down. In order to perform numerical optimization on the hypothesis, the simplified log-likelihood function must be differentiated in Cartesian space as a function of the emitter coordinates.

This requires substituting $\bar{r}$ in (17) with the actual expression for i-th sensor and the emitter location:

$$l_s = -\sum_{i=1}^{n} \frac{\left((d_i - d_j) - \left(\sqrt{(x_i - x_e)^2 + (y_i - y_e)^2 + (z_i - z_e)^2} - \sqrt{(x_j - x_e)^2 + (y_j - y_e)^2 + (z_j - z_e)^2} \right) \right)^2}{2\left(\sigma_i^2 + \sigma_j^2\right)} \quad (18)$$

where x_e, y_e, z_e are Cartesian coordinates of the emitter according to the current hypothesis, x_i, y_i, z_i are the coordinates of the i-th sensor, x_j, y_j, z_j are coordinates of j-th sensor. Each value of the index i has the corresponding value of j according to the differential measurement scheme; for this reason, index j does not appear under the adduct sign. The first derivative in Cartesian coordinates by the emitter position is the same as the derivative by the sensor position with a minus sign due to the relative properties of the distance vector. It is useful for computation optimization since the latter is needed to compute the contribution of sensor position uncertainty to pseudo-range dispersion. The derivative by x dimension, and other dimensions derivatives are analogous:

$$\frac{\partial l_s}{\partial x_e} = \sum_{i=1}^{n} \left(\frac{x_i - x_e}{\|\bar{r}_i\|} - \frac{x_j - x_e}{\|\bar{r}_j\|} \right) \frac{(d_i - d_j) - (\|\bar{r}_i\| - \|\bar{r}_j\|)}{\sigma_i^2 + \sigma_j^2}$$

where the left part of the product is the difference of the projection of $\bar{r}_i$ and $\bar{r}_j$ on the corresponding dimension and the right part is the pseudo-range difference between i-th and j-th sensors over their compound dispersion. The derivative can be written using vector notation

$$\nabla l_s = \sum_{i=1}^{n} \left(\frac{\bar{r}_i}{\|\bar{r}_i\|} - \frac{\bar{r}_j}{\|\bar{r}_j\|} \right) \frac{(d_i - d_j) - (\|\bar{r}_i\| - \|\bar{r}_j\|)}{\sigma_i^2 + \sigma_j^2} \quad (19)$$

This formula (19) is the core of numerical MLE computation; it will be used with one of the optimization algorithms to produce the maximum likelihood estimate of the emitter location.

The second derivative is more complex but will be needed for computing the Fisher information matrix. The second derivative of the log-likelihood function is the measure of estimate certainty, and the sum over all measurements is the inverse of the resulting estimate dispersion matrix. The simplified version of the second derivative will be used in the proposed optimization algorithm to improve convergence speed. The full second derivative formula for Cartesian coordinates:

$$\nabla^2 l_s = \sum_{i=1}^{n} \frac{1}{\sigma_i^2+\sigma_j^2}\left(\left(\frac{\bar{r}_i}{\|\bar{r}_i\|}-\frac{\bar{r}_j}{\|\bar{r}_j\|}\right)^T\left(\frac{\bar{r}_i}{\|\bar{r}_i\|}-\frac{\bar{r}_j}{\|\bar{r}_j\|}\right)+\left(\left(d_i-d_j\right)-\left(\|\bar{r}_i\|-\|\bar{r}_j\|\right)\right)(K-P)\right) \quad (20)$$

where $K=\left(\frac{\bar{r}_i}{\|\bar{r}_i\|}\right)^T\left(\frac{\bar{r}_i}{\|\bar{r}_i\|^2}\right)-\left(\frac{\bar{r}_j}{\|\bar{r}_j\|}\right)^T\left(\frac{\bar{r}_j}{\|\bar{r}_j\|^2}\right)$ and $P=diagonal\left(\frac{1}{\|\bar{r}_i\|}-\frac{1}{\|\bar{r}_j\|}\right)$

The complexity of the second derivative as it is would hinder the performance of the Newton-Raphson method negating the benefit of fewer iterations by the increased amount of computation. The optimization method proposed in this work considers the Hessian matrix as a local linear transform that is supposed to convert the elongated “valley” shape of the log-likelihood function to a shape more suitable for the optimization procedure. From this perspective, the computation of the local linear transform can be simplified by setting the measured pseudo-range difference equal to the theoretical pseudo-range that would have been measured if the emitter was located at the current state of the hypothesis vector; that is, under the consideration that the optimizer has already converged onto the stationary point. This assumption sets

$$\left(\left(d_i-d_j\right)-\left(\|\bar{r}_i\|-\|\bar{r}_j\|\right)\right)=0 \quad (21)$$

and eliminates the right part of the (20), resulting in a much simpler formula

$$\nabla^2 l_s = \sum_{i=1}^{n} \frac{1}{\sigma_i^2+\sigma_j^2}\left(\frac{\bar{r}_i}{\|\bar{r}_i\|}-\frac{\bar{r}_j}{\|\bar{r}_j\|}\right)^T\left(\frac{\bar{r}_i}{\|\bar{r}_i\|}-\frac{\bar{r}_j}{\|\bar{r}_j\|}\right) \quad (22)$$

The simplified formula of the second derivative can be used for Fisher matrix computation when the number of sensors is $N+1$ where N is the number of problem dimensions; in this case, under appropriate sensor constellation configuration (not degenerate case), the condition (21) is always true in the stationary point. With the minimum number of sensors, the system of hyperbolic equations is exactly defined, and all residual differences are zeros.

Numerical Optimization Algorithms

After the mathematical terms for the maximum likelihood estimator were established in the previous section, this section will focus on the proper selection of the optimization techniques. Unfortunately, many of the methods proposed in publications, particularly the weighted least squares algorithm [3], do not guarantee convergence if the initial point is chosen too far from the actual emitter location.

The situation where the optimization algorithm diverges despite the fact that an estimate exists should be discerned from the situation where the measurement errors are too large, and the system of hyperbolic equations does not have any solutions. The latter situation might always happen in the practical application but does not pose a significant problem because accumulating a larger number of measurements will usually resolve the issue. On the other hand, numerical stability and guaranteed convergence are strongly desirable qualities for any algorithm that is meant for practical applications. In this work, modifications of the line backtracking search algorithm will be considered [18]; this algorithm is guaranteed to converge on the convex target function.

The guaranteed convergence of the line backtracking algorithm for gradient descent is theoretically proven using the Armijo-Goldstein condition; the gradient descent optimization algorithm in this work will use this condition to provide a guaranteed convergence. In the case of a log-likelihood estimator, it is more intuitive to construct an optimizer for the maximization problem since we are looking for the log-likelihood function maximum; however, in order to be consistent with the terminology of numerical optimization methods in this part, the negated log-likelihood function is considered.

The line backtracking algorithm itself does not dictate the way of choosing the direction of the step; it only requires that direction points towards the minimum. For the scope of this paper, three methods of determining the direction of the step will be considered. The first one is the most basic; the direction of the step is computed as a normal vector towards the maximum negative gradient direction.

The second approach is inspired by the Adam optimizer [19] and uses “momentum” that mimics the inertia of mechanical movement; that helps the optimizer to avoid zigzag bouncing inside elongated minima “valley” before reaching the true minimum. Despite having a better performance than choosing the step direction based solely on the maximum local gradient steepness, it comes with the disadvantage of adding another tunable parameter for the momentum “recall” function. The third proposed method is a modification of the Newton-Raphson method combined with the line backtracking search.

The key difference between the proposed method and the basic Newton-Raphson method implementation is the way the Hessian matrix is computed. As it was shown in the previous chapter, the expression for the mixed derivatives matrix of the log-likelihood function is quite complex (20); for the TDoA problem maximum likelihood estimator, a simpler approximation was proposed (22). The advantage of this simplified version is that it does not depend on the actual measurements except for the dispersion estimate; this dramatically reduces the amount of additional computation since all parts of (22) are already computed during log-likelihood function gradient evaluation.

Since the step direction vector is normalized to the unit vector before further usage, scaling the Hessian matrix by a constant factor will not affect the result of the computation. For this reason, further simplification is possible by removing the reciprocal compound dispersion from (22). On the other hand, this value is already available, and removing it from the formula will result in applying the same weight to the inputs from sensors with different dispersion estimates. Although such an optimization might be acceptable for some algorithm applications, for the general-purpose algorithm, it is undesirable. Ignoring the dispersion-based weight of the sensors input in the situation when those dispersion estimates might differ significantly might result in a degraded optimization algorithm convergence rate.

Having considered the available options for choosing the step direction, let us consider the operation of the line backtracking search algorithm in more detail. The aforementioned Armijo-Goldstein condition is the critical element of the algorithm; it is used for choosing the step size by trial-and-error method. The generic version of the line backtracking search algorithm has tunable parameters σ, τ, and state variable α that should be initialized to some initial value before the algorithm operation.

The τ parameter is a multiplicative rate of the step size change that is applied to the state variable α depending on the outcome of testing Armijo-Goldstein condition for the current value of α. The parameter σ is the part of the Armijo-Goldstein condition; it determines the maximum allowed ratio between the target function increment predicted by the value of the gradient projected onto the step direction and the actual increment determined by computing the target function at the new point. If the ratio between actual and predicted increment values is smaller than σ, then backtracking is performed; the next value of the step size is chosen $\alpha_{n+1} = \tau\alpha_n$. The process is repeated until the step is small enough to satisfy the Armijo-Goldstein condition. When the condition is satisfied, the step is applied to the current estimate, and the process is repeated until the algorithm convergence is detected.

Comparison of Optimization Algorithm Implementations

Three variants of the gradient descent optimization algorithm were implemented using Python and NumPy library. The test environment consists of the implementation of the sensor model, sensors constellation model, simulation test fixture, and simulation parameters configuration. The design under test consists of two parts: the point evaluator and the gradient descent algorithm that has three implementations.

The sensor model is represented by the `Sensor` class that has attributes that define sensor spatial location and standard deviations for the time and location estimates. The `measure` method of the `Sensor` class accepts the true emitter location and stateful random number generator instance that is used to compute random deviations for simulating noisy measurements. The result of this method is a named tuple containing measured pseudo-range, the estimate of the sensor location, and the copy of its own standard deviation attributes to be used as estimates of sensor accuracy.

The group of time-synchronized sensors that operate together and receive the same signal in different spatial locations is represented by the `Constellation` class. This class has a predefined number of sensors with different spatial locations as its attributes and has a `measure` method that aggregates the result of calling a similar method on each of the sensors.

There are several implementations of the test fixture for different types of simulations. The main class of the test fixture is called `WorldSetup`, which contains the constellation, the state of the random number generator, the “dummy” random number generator that produces zero deviation, and the `Configuration` class instance that contains common simulation parameters that are stored in JSON format.

The `PointEvaluator` class contains the code that performs TDoA computations for a set of hypothesis points; it contains method that compute the first, the second mixed derivatives, implements a differential measurement scheme for signal time-of-arrival, and provides aggregation between multiple measurements from the multiple sensors.

The base class of optimizer implementation is `BTOptimizer`; it implements the line backtracking search using the Armijo-Goldstein condition. Two other optimizer implementations are derived classes `BTMOptimizer` that implements “momentum” method and `BTTOptimizer` that implements pseudo-Newton local linear transform method.

In the prototype implementation, the stopping criterion for the optimization algorithm is the step size. It might be somewhat unreliable for the simplest implementation variant that uses the maximum gradient direction, but for the other two improved variants, it demonstrates sufficient reliability.

The scripts for the convergence simulation are located in *program.py* and *benchmark.py* files; they use the Matplotlib library to visualize the results. The examples of the convergence graph are shown for three variants of the optimizer. The initial point $x0$ is taken with the predefined offset from the true emitter location. The measurements used for this simulation are synthesized using an injection of a random error with a specified standard deviation into pseudo-range measurements and sensor location estimates. This example setup is used as a reference testbench for the estimator convergence speed comparison; the figures below illustrate the convergence path for each type of optimizer for the elongated “valley” shaped minima.

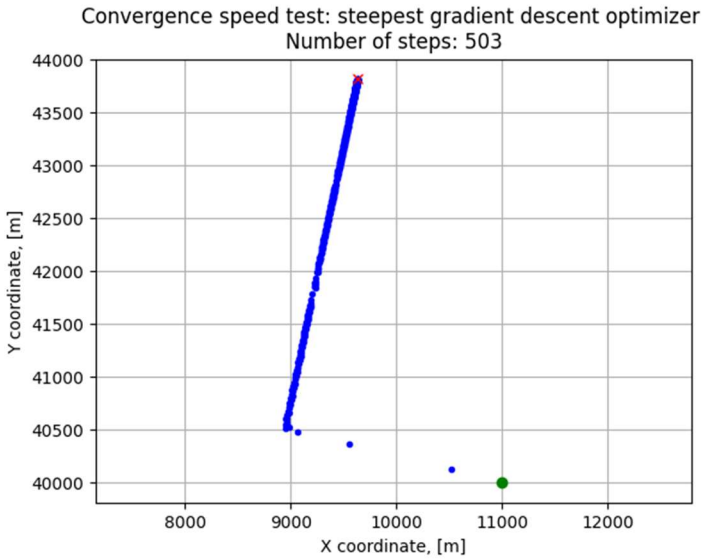


Fig.3 Steepest gradient descent method convergence

The basic gradient descent algorithm exhibits fast convergence towards the direction of the maximum slope followed by bouncing inside the “valley” region, as shown in Fig.3. The total number of steps before reaching the cut-off criterion is 503; most steps are comprised of bouncing along the direction of the vanishing gradient.

This problem can be partially mitigated by using a “momentum” optimizer which uses information from the previous steps to improve the directional stability of the algorithm. The disadvantages of a “momentum” optimizer are the tendency to overshoot and the extra ad hoc “recall” parameter that should be tuned for the specific application to achieve required performance improvement. In this test, the recall parameter is set to $r = 0.9$, which is determined experimentally based on the simulation results.

Using the “momentum” optimizer resulted in a decrease of the number in iterations by 73% in comparison to the steepest gradient descent optimizer. The overshot phenomenon is observed on the transition from the slope of the log-likelihood function to the ridge with a small slope towards the stationary point; it appears in Fig.4 as a minor transient oscillation of the step direction.

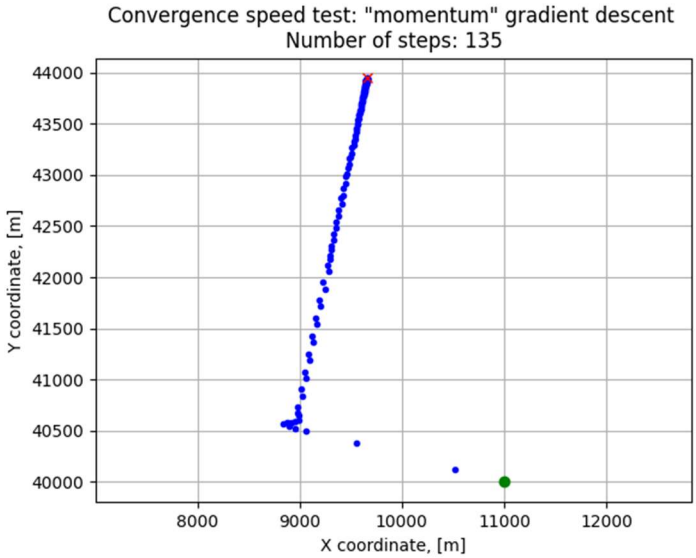


Fig.4 “Momentum” method convergence

The next method to consider is a Newton-Raphson gradient descent method with a simplified Hessian matrix. The main issue with the original Newton-Raphson method in application to the TDoA problem is that the Hessian matrix is costly in terms of computations and might become ill-conditioned and not invertible due to measurement errors at one of the optimization steps.

This problem was addressed by the Hessian approximation proposed in this work. This is not the only available option; the Hessian matrix can be approximated by the finite difference computed on the first derivative. The disadvantage of the finite difference method is a potential problem with numerical stability, especially when a 32-bit floating-point number representation is used. As this work targets GPU as a primary platform for the parallelized algorithm implementation, limiting the algorithm to using 64-bit floats is not desirable from an efficiency standpoint.

The proposed optimization of the Hessian matrix computation is based on the observation that the Hessian matrix in the Newton-Raphson gradient descent method is used as a local linear transform that is applied to the optimization step. Since the length of the step is determined by the line backtracking search algorithm, the only purpose of the Hessian matrix is to provide the direction that maximizes the convergence speed.

On the other hand, the Hessian matrix is the inverse of the Fisher information matrix for the given point in the case of a log-likelihood function based on the normal distribution. The actual Hessian matrix depends on the concrete measurements sample; it determines the Fisher information matrix and the estimate of the dispersion ellipse.

From the statistical perspective, this concrete estimate is just a sample from the general distribution, which also has a corresponding Fisher information matrix and dispersion ellipse. The idea behind the optimization is to replace the Hessian matrix that corresponds to the concrete measurement samples with the Hessian matrix of the general distribution that does not depend on the concrete measurement samples.

However, the general distribution Hessian depends on the sensors' location, which is not known precisely. With the stationary sensors, the averaged location estimate can be used; with the airborne system, each sample should be used to compute Hessian, and a dispersion-based average should be computed.

The third version of the optimization algorithm dramatically reduced the total number of iterations; the optimizer converges in 11 steps on the same test. It makes steps toward the stationary point and reaches the threshold with a slight overshoot, as indicated in Fig.5. No bouncing behavior is observed; as a result, the proposed method converges 45 times faster than the steepest gradient descent method and 12 times faster than the “momentum” method in terms of the number of iterations. It is worth noting that the real run-time advantage is somewhat smaller due to the extra computations involved.

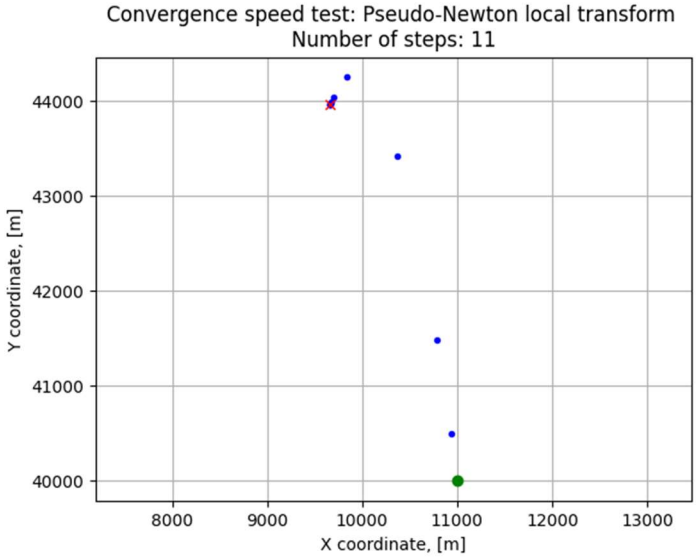


Fig.5 Modification of Newton-Raphson method convergence

The next step for the optimization algorithm performance evaluation is comparing the convergence of the proposed method against the original Newton-Raphson method in terms of convergence speed and the amount of computation required to converge. A modified version of the `BTTOptimizer` is used to evaluate the performance; it is implemented in the class `BTNOptimizer`. It should be noted that the exact Hessian is not guaranteed to be positive definite due to measurement errors. This issue might manifest with an unfavorable sensors’ constellation configuration.

When the Hessian matrix is not positive definite or close to singular numerical error can undermine the algorithm stability and result in the divergence or convergence to the wrong location; therefore, the Hessian matrix condition number should be monitored for validity.

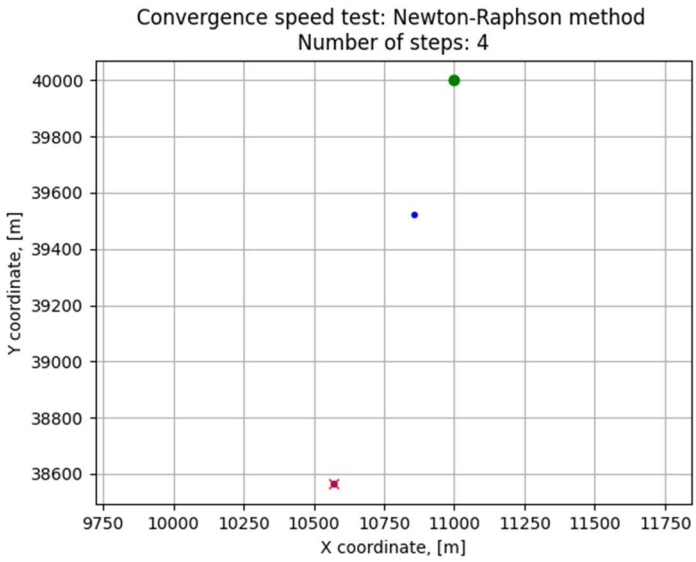


Fig.6 Exact Hessian Newton-Raphson method convergence

Fig.6 indicates that the line backtracking optimizer converged to the wrong value in just 4 steps. Further investigation of this failure has shown that the reason for the wrong behavior is the exact Hessian matrix; it had a condition number $cond(H) > 2500$ at the beginning and was close to singular at the last iteration. The Hessian matrix was not effectively invertible in the presence of numerical errors; this result confirms potential issues with using the exact Hessian value for computing the direction for the gradient descent method.

Despite the order-of-magnitude difference in the number of steps needed for the algorithm to converge between the “momentum” steepest gradient descent method and the modified Newton-Raphson method, the difference in terms of actual CPU time for a single-threaded implementation is expected to be smaller. In order to determine which method converges faster in terms of the amount of computation time, profiling code was added to the corresponding optimizer implementations.

Table 1. Measured optimizer execution time on CPU

Optimizer	SGD	Momentum	Proposed
Execution time, ms	671.8	185.6	23.8
Speedup, times	-	3.61	28.19

The proposed optimization method has improved overall convergence time ~28 times compared to the steepest gradient method; it indicates that computational complexity per step increased by ~80% due to transform computation and using it to determine the direction for the line backtracking algorithm. These numbers correspond to the implementation that performs addition after the inverse transform matrix computation and, therefore, does not impose any limitations on the sensor constellation motion.

The sum of positive definite matrices is a positive definite matrix, implying that sensors' motion does not cause any issues with the summed matrix inversion. Measurement error autocorrelation does not need to be compensated since the transform is used to compute step direction only, and scaling it by scalar value does not impact the result. Variance-based component weighting can be used when summing the contribution from each measurement; however, in most applications, it is redundant. Usually, a consecutive batch of measurements is used for the optimization run; in this situation, the dispersion estimate will not vary significantly over the measurement series.

The previous convergence tests did not take into account different scenarios of the relative sensors and emitter positions. The convergence speed might be different depending on the relative distance to the emitter compared to the radius of the sensors' constellation. This dependency is important for practical applications where the situation with a small constellation radius and a distant emitter is a well-known adverse scenario that impacts the performance of TDoA algorithms. This behavior is caused by the elongation of the minima "valley" when the distance to the emitter increases; at some point, the range to the emitter cannot be determined reliably, and the TDoA algorithm degrades into AoA estimator. Some algorithms degrade gracefully, providing an accurate estimate for the angle of arrival, while other approaches might fail completely.

Benchmarking a dense mesh of all possible constellation configurations along with emitter locations is a computationally intense task that is not feasible in the scope of this work. The simplified version with fixed sensor positions and a sparse mesh of the sensor locations is used to assess the proposed algorithm performance depending on the relative distance to the emitter.

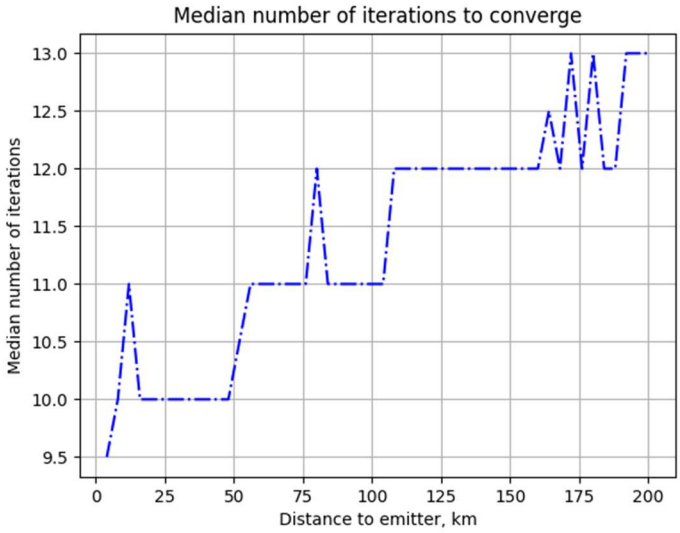


Fig.7 Number of iterations to converge

Fig.7 illustrates the number of iterations required for the proposed optimizer to converge. The tendency to increase the number of iterations for larger distances is observed. However, the trend is not significant to have an impact on the practical applications due to TDoA estimate accuracy degradation with the distance to the emitter makes the estimate not worthwhile before any concerns with the algorithm convergence speed could arise.

The Hessian matrix of a log-likelihood function can be used as an approximation of the Fisher information matrix. When the log-likelihood function formula was established, simplification assumptions were made. The first one is the linearity of projection transforms, which is used to compute pseudo-range dispersion contributions; for relatively small measurement errors, it does not introduce a significant discrepancy. The second one is that the sum of normal distributions is a normal distribution; this term does not diminish when measurement errors are small, and it is considered a significant source of error in the Fisher matrix estimate.

The dispersion predicted by the Fisher information matrix can be represented by an error ellipse or an error ellipsoid in the case of 3-dimensional distribution. The ellipsoid itself is an approximation; the actual shape of the points cloud is always slightly different; nevertheless, this is the most common way of presenting the location accuracy data.

The proposed estimator is a maximum-likelihood estimator by design and should attain the Cramer-Rao lower bound performance. The comparison between the actual location estimate distribution and the predicted Fisher information matrix is made by sampling the estimates for noisy measurements with the predefined standard deviations and computing the actual dispersion matrix from the location estimate deviations. The Fisher information matrix is computed as a negative Hessian matrix for the noise-free measurement at the given emitter location and then inverted using `numpy.linalg` library. The decomposition into error ellipsoid half-axes is accomplished by obtaining the dispersion matrix eigenvalues using the same library.

The error ellipsoid half-axes represent the standard deviation of the location estimate points cloud along the corresponding eigenvector direction. This representation reduces the dispersion representation from a 3-by-3 matrix to 3 scalar values under the assumption that error ellipsoids have similar spatial configurations. The half-axes length is computed as square roots of the corresponding eigenvalues of the dispersion matrix. The problem with this approach is that eigenvalues must be matched between the two matrices. The simplest way to accomplish that is to order them by their magnitude. For the estimator Cramer-Rao lower bound performance evaluation, the distance of the emitter from the sensors' constellation centroid is a variable parameter of the simulation; the outputs are measured and predicted error ellipse half-axes lengths.

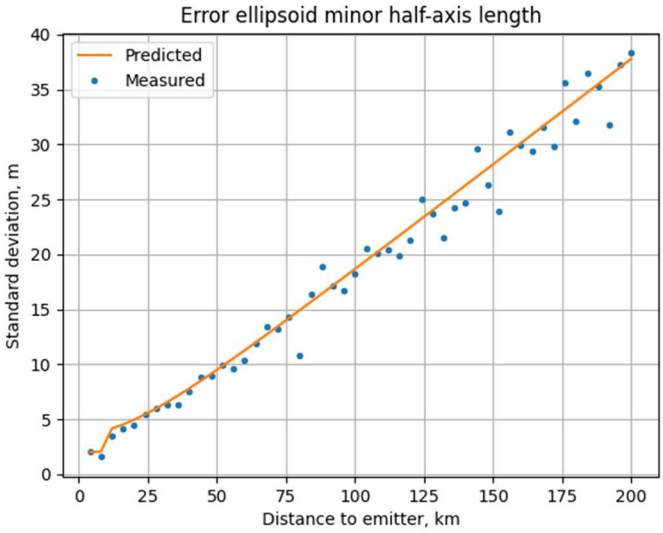


Fig.8 The minor error ellipsoid half-axis

The results for the minor error ellipse half-axis that correspond to TDoA azimuth error are shown in Fig.8; the experimental results are well-matched with the predicted values.

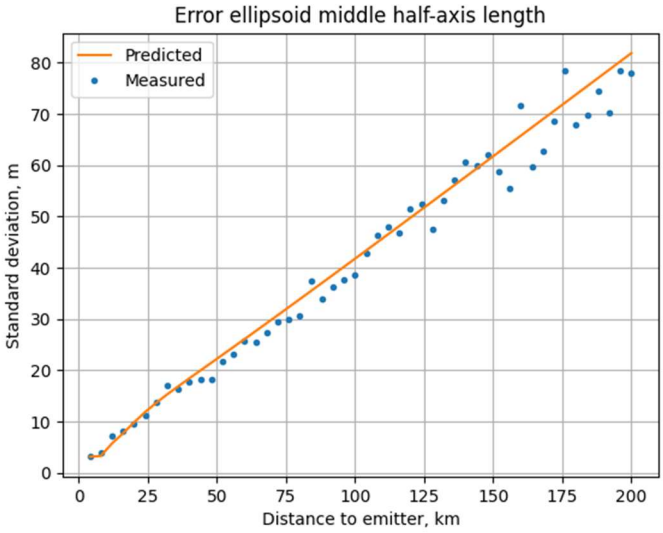


Fig.9 The middle error ellipsoid half-axis

The middle and the major half-axes correspond to the elevation and the range to the emitter; they have a reasonable match between the predicted and the measured values. These results indicate that the optimizer attains Cramer-Rao lower bound performance. The important detail that should be noted from Fig.10 is that the range estimate standard deviation has a quadratic dependency on the range. At the same time, the azimuth and the elevation are linear.

The performance of the geolocation algorithm in regard to the Cramer-Rao lower bound does not depend on the particular optimizer used as long as the optimizer converges to the same point; the underlying mathematical model of the log-likelihood function is the same for all optimizers, and therefore, the stationary points of the log-likelihood function are also identical. Nevertheless, the comparison of the results to the Cramer-Rao lower bound performance is a sanity check for the geolocation algorithm that confirms the correctness of the implementation.

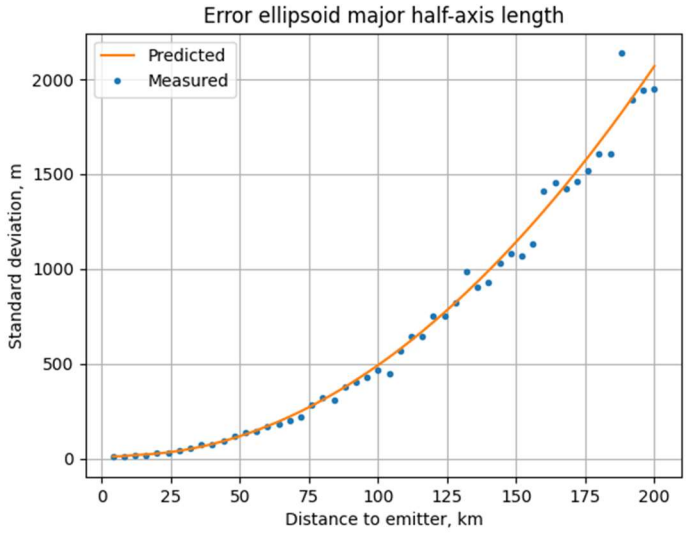


Fig.10 The major error ellipsoid half-axis

Several methods of determining the starting point exist; the Chang-Ho method allows for obtaining a reasonable estimate when an accumulation of a large number of measurements is not needed to provide the required accuracy for the initial location estimate. Since the Chang-Ho method implementation is not in the scope of this work and computations are intended for the parallelized implementation, the initial point can be computed as the maximum point of the log-likelihood function computed on the fixed grid.

It is reasonable to use the irregular grid in polar coordinates that change the points' density according to the dispersion ellipse estimate. The estimator usually converges quite well even from the location estimate that corresponds to the centroid of the sensor constellation; nevertheless, there is a risk of log-likelihood function non-convexity under conditions of unfavorable sensors positions or significant measurement errors. Using the sensor formation centroid as a starting point for the optimizer can be considered a degenerate case of the mesh-based approach where the mesh is reduced to a single fixed initial point.

The proposed optimization algorithm provides reasonable performance improvement by using a local linear transform instead of an actual Hessian matrix to implement the modified Newton-Raphson method in combination with the line backtracking search. This implementation is chosen for the parallelization using massive parallel data processing utilizing GPU and CUDA technology in conjunction with appropriate linear algebra libraries.

Implementing a Parallelized Version of the Estimator

The possible alternatives for utilizing GPU are CUDA [29], OpenCL [30], and linear algebra libraries that are wrappers over the CUDA interface [31]. From the runtime performance perspective, CUDA and OpenCL are the most potent options, provided the code is well-optimized. On the other hand, the ready-made linear algebra and vectorized computation library is a more attractive option for prototyping and experimentation; the implementation effort could be dramatically reduced by the expense of the potential computation speedup.

The CUDA toolkit by Nvidia already contains libraries for linear algebra and other applications, including cuBLAS, cuRAND, cuSOLVER. The python wrapper CuPy provides an application programming interface similar to NumPy, and therefore it is the library of choice for exploring the potential speedup of the algorithm functions implemented using NumPy. In fact, the geolocation algorithm is already designed to use vectorized operations, and the computationally heavy functions are placed into a separate class that can be easily replaced with an implementation that uses CUDA libraries instead of CPU.

From the performance perspective, the greatest concern when using GPU is a potential input-output bottleneck. It is important to avoid data type conversions and make intermediate results stay in GPU memory whenever possible to avoid communication overhead. The implementation of *CuPy* addresses this issue by using a special handle object that is a wrapper over the actual data located in the GPU memory. From the user perspective, the operations and library calls are performed on the handle objects themselves; the implementation conceals the details of GPU memory management.

Since the program prototype is written in a scripted language that uses garbage collection, performance benchmarking between libraries that use CPU and GPU should use an appropriately large size of the input data. The larger the data size, the less impact the prototype implementation will have on the results; on the other hand, the data size, including all the intermediate results, should not exceed the amount of memory of the GPU. For the available hardware using 100000 measurement samples produced consistent results with the smaller batches but with lower variance; therefore, it might be concluded that the memory bottleneck was not reached.

Performance testing was done by comparing the running time of the optimizer on pre-generated data; data transition between CPU and GPU memories was not accounted into the running time, and the implementation of the optimizer was “pre-heated” by running once with the same data set. The details about the test hardware can be found in table 2.

Table 2. Hardware used

Parameter	CPU	GPU
Model	Intel Core i7-1185G7	Nvidia GeForce GTX1650
Memory	16GB	4GB
Number of physical cores	4@3.0GHz	896@1.485GHz

Three versions of optimizers were tested in three different versions: CPU-based, GPU-based, and GPU-based, using 32-bit floating-point numbers representation. The test was conducted several times without running any demanding background tasks to ensure results consistency and repeatability. The results are shown in Table 3.

Table 3. Performance testing results

Result	SGD	Momentum	Proposed
CPU time, s	80.03	22.49	2.02
GPU time, s	6.72	1.84	0.193
GPU time 32bit, s	3.80	1.02	0.108
GPU/CPU speedup	11.9	12.2	10.5
GPU 32bit/CPU speedup	21.1	22.1	18.7

The performance improvement from parallelization is relatively consistent for all three implementations ranging from 10.5 to 12.2 times when using only offloading from CPU to GPU and 18.7 to 22.1 times when the offloading is combined with reduced precision to 32 bits floating-point numbers. The proposed version of the optimization algorithm benefits from using the GPU slightly less than simpler implementations; nevertheless, overall performance improvement achieved is 741 times compared to the SGD on the CPU to the most optimized version or 208 times for the “momentum” optimizer on the CPU.

The performance difference between 64 and 32-bit floating-point numbers on the CPU was not measured. It is expected to be the same order of magnitude as for the GPU, ~2 times judging from the processor floating-point computation benchmarks. The final question with the 64 to 32-bit transition is the impact on the algorithm accuracy. In order to assess the accuracy loss, the resulting dispersion matrices of 64 and 32-bit implementations were compared. Increased error due to round-off errors would result in an increase in the dispersion along the axes of the dispersion ellipsoid. Since the difference is expected to be small, it is expressed in percentages for each of the ellipsoid axes.

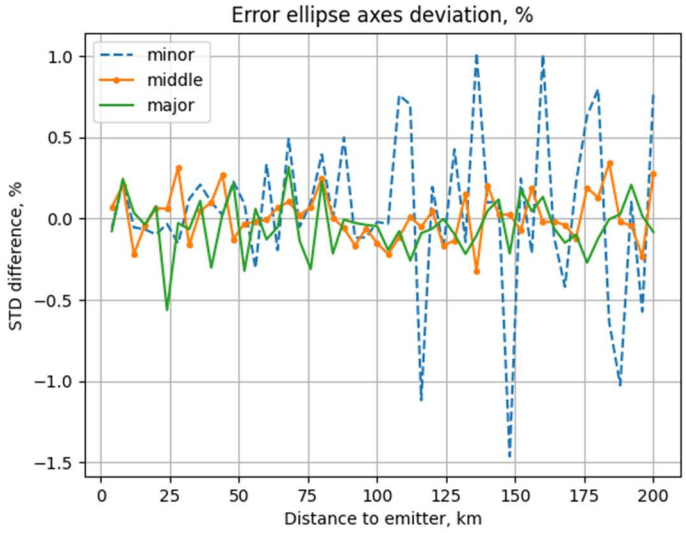


Fig.11 Dispersion difference between 64 and 32-bit implementations

Fig.11 suggests that lower representation accuracy for floating-point numbers has an impact on the optimizer, but there is no steady trend that might indicate estimation accuracy degradation; the random difference can be attributed to the noise introduced by a limited number of simulations per emitter location. Single precision floating point numbers in the matrix inversion operation introduce a risk of insufficient numerical stability, and this specific operation could be done using double-precision; however, the result of the simulation is totally acceptable even if all the computations are made using a single-precision.

Conclusion

The maximum likelihood estimator for solving TDoA and multi-sensor geolocation problems for arbitrary sensors array configuration was considered in the first part of this work, as well as closed-form solutions for the hyperbolic equations for the geolocation problem. Three different gradient descent methods based on the line backtracking algorithm were proposed and implemented as optimizers for the maximum-likelihood estimator.

As a result of this work, the estimator convergence speed was improved by using a local linear transform. This new approach can be considered an adaptation of the Newton-Raphson algorithm with the Hessian matrix replaced with a statistically approximated value that guarantees the algorithm convergence and reduces the amount of computation. Performance testing of the optimization algorithms has shown that the proposed algorithm outperforms the steepest gradient descent 28.2 times and “momentum” modification 7.8 times on the CPU implementation.

In the next part, these three algorithms were parallelized using CUDA linear algebra library wrapped into CuPy library using both 64 and 32-bit floating-point numbers representation, resulting in a further performance increase of 10.5 and 18.7 times. The simulation has shown that the difference between 64 and 32-bit implementations is present but does not impact the overall algorithm accuracy in a significant way; the dispersion of the estimate in both cases is sufficiently close to the Cramer-Rao lower bound.

TDoA algorithm implementation, optimization, and parallelization have shown that both algorithmic approach and appropriate hardware for performing the computations are essential for achieving performance improvement. Some of the fastest optimization algorithms, like Newton-Raphson, might not work with the problem statement in the original version. Using the appropriate precision to represent the floating-point number is an effective optimization for numerically stable algorithms that comes without extra cost.

References

[1] ITU, 2018, *Comparison of Time-Difference-of-Arrival and Angle-of-Arrival Methods of Signal Geolocation*.

[2] Chan, Y.T. & Ho, K. (1994). *A Simple and Efficient Estimator for Hyperbolic Location.* Signal Processing, IEEE Transactions on. 42. 1905 - 1915. 10.1109/78.301830.

[3] Jin, Bonan & Xu, Xiaosu & Zhang, Tao. (2018). *Robust Time-Difference-of-Arrival (TDOA) Localization Using Weighted Least Squares with Cone Tangent Plane Constraint.* Sensors (Basel, Switzerland). 18. 10.3390/s18030778.

[4] S. Safavi and U. A. Khan, “Localization in mobile networks via virtual convex hulls,” IEEE Trans. Signal Inf. Process. Netw., vol. 4, no. 1, pp. 188–201, Mar. 2018.

[5] G. Wang, A. M. C. So, and Y. Li, “Robust convex approximation methods for TDOA-based localization under NLOS conditions,” IEEE Trans. Signal Process., vol. 64, no. 13, pp. 3281–3296, Jul. 2016.

[6] Y. Sun, K. C. Ho, and Q. Wan, “Solution and analysis of TDOA localization of a near or distant source in closed-form,” IEEE Trans. Signal Process., vol. 67, no. 2, pp. 320–335, Jan. 2019.

[7] Li, Sasa, and Ho Dominic K C. *“An Efficient and Computationally Attractive Localization Algorithm under Large Equal Radius Scenario.”* 2015.

[8] Y. T. Chan, H. Y. C. Hang, and P. C. Ching, “Exact and approximate maximum likelihood localization algorithms,” IEEE Trans. Veh. Technol., vol. 55, no. 1, pp. 10–16, Jan. 2006.

[9] L. Lin, H. C. So, Frankie K.W. Chan, Y. T. Chan, and K. C. Ho, “A new constrained weighted least squares algorithm for TDOA-based localization,” Signal Process., vol. 93, no. 11, pp. 2872–2878, Nov. 2013.

[10] S. Aditya, A. F. Molisch, N. Rabeah, and H. M. Behairy, “Localization of multiple targets with identical radar signatures in multipath environments with correlated blocking,” IEEE Trans. Wireless Commun., vol. 17, no. 1, pp. 606–618, Jan. 2018.

[11] Senturk, Hakan. *“Performance Evaluation of Hyperbolic Position Location Technique in Cellular Wireless Networks.”* Thesis / Dissertation ETD, Air Force Institute of Technology, 2002.

[12] X. Qu, L. Xie, and W. Tan, “Iterative constrained weighted least squares source localization using TDOA and FDOA measurements,” IEEE Trans. Signal Process., vol. 65, no. 15, pp. 3990–4003, Aug. 2017.

[13] Jiang M, Niu R, Blum R S. Bayesian target location and velocity estimation for multiple-input multiple-output radar. IET Radar, Sonar and Navigation 2011; 5(6): 666-670.

[14] Li T, Ekpenyong A, Huang Y F. Source localization and tracking using distributed asynchronous sensors. IEEE Transactions on Signal Processing 2006; 54(10):3991-4003.

[15] Li, Wenhua & Liu, Peiguo. (2005). *3D AOA/TDOA emitter location by integrated passive radar/GPS/INS systems.* 121 - 124. 10.1109/IWVDVT.2005.1504566.

[16] Jahshan, Bhatti A, et al. University of Texas, 2012, *Development and Demonstration of a TDOA-Based GNSS Interference Signal Localization System.*

[17] Hugo Seuté, Cyrille Enderli, Jean-François Grandin, Ali Khenchaf, Jean-Christophe Cexus. *Experimental Measurement of Time Difference Of Arrival.* International Radar Symposium 2016 - IRS2016, May 2016, Krakow, Poland.

[18] Du, Huai-Jing & Lee, Jim. (2004). *Simulation of Multi-Platform Geolocation Using a Hybrid TDOA/AOA Method.*

[19] YU, Huagang & HUANG, Gaoming & GAO, Jun & WU, Xinhui. (2012). *Approximate Maximum Likelihood Algorithm for Moving Source Localization Using TDOA and FDOA Measurements.* Chinese Journal of Aeronautics. 25. 593–597. 10.1016/S1000-9361(11)60423-8.

[20] N. Vankayalapati, S. Kay and Q. Ding, *"TDOA based direct positioning maximum likelihood estimator and the cramer-rao bound,"* in IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 3, pp. 1616-1635, July 2014, doi: 10.1109/TAES.2013.110499.

[21] Díez-González J, Álvarez R, Sánchez-González L, Fernández-Robles L, Pérez H, Castejón-Limas M. *3D Tdoa Problem Solution with Four Receiving Nodes.* Sensors (Basel). 2019 Jun 29;19(13):2892. doi: 10.3390/s19132892. PMID: 31261946; PMCID: PMC6651820

[22] H. Liu, Y. -M. Pun and A. M. -C. So, *"Local strong convexity of maximum-likelihood TDOA-Based source localization and its algorithmic implications," 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1-5, doi: 10.1109/CAMSAP.2017.8313119.

[23] K. W. Cheung, H. C. So, W. -. Ma and Y. T. Chan, *"Least squares algorithms for time-of-arrival-based mobile location,"* in *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 1121-1130, April 2004, doi: 10.1109/TSP.2004.823465.

[24] B. Friedlander, *"A passive localization algorithm and its accuracy analysis,"* in IEEE Journal of Oceanic Engineering, vol. 12, no. 1, pp. 234-245, January 1987, doi: 10.1109/JOE.1987.1145216.

[25] H. Schau and A. Robinson, *"Passive source localization employing intersecting spherical surfaces from time-of-arrival differences,"* in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35, no. 8, pp. 1223-1225, August 1987, doi: 10.1109/TASSP.1987.1165266.

[26] J. Smith and J. Abel, *"Closed-form least-squares source location estimation from range-difference measurements,"* in IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 35, no. 12, pp. 1661-1669, December 1987, doi: 10.1109/TASSP.1987.1165089.

[27] Absil, P.-A., Mahony, R., Andrews, B.: Convergence of the iterates of descent methods for analytic cost functions. SIAM J. Optim. 16(2), 531–547 (2005)

[28] Dubey, Shiv Ram & Basha, Sh Shabbeer & Singh, Satish & Chaudhuri, Bidyut. (2021). Curvature Injected Adaptive Momentum Optimizer for Convolutional Neural Networks.

[29] https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

[30] https://www.khronos.org/opencl

[31] https://cupy.dev