Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for GPS Receiver Position Estimation in Coastal Region of India

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

B. Sai Sampath (319126512070)

M. Satya Sudha (319126512098)

G. Sai Jashwanth (319126512113)

P. Akhila Princy (319126512106)

Under the guidance of

Dr. Ashok Kumar N

Associate Professor

PRAGNANAM BRAHMA

ANITS

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

(*Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)*

Sangivalasa, Bheemili Mandal, Visakhapatnam Dist. (A.P)

2022-2023

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES
(UGC AUTONOMOUS)
(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC)
Sangivalasa, Bheemili Mandal, Visakhapatnam Dist. (A.P)

PRAGNANAM BRAHMA

ANITS

CERTIFICATE

This is to certify that the project report entitled **"Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for GPS Receiver Position Estimation In Coastal Region of India"** submitted by **B. Sai Sampath (319126512070), M. Satya Sudha (319126512098), G. Sai Jashwanth Kumar (319126512113), P. Akhila Princy (319126512106)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Anil Neerukonda Institute of Technology and Sciences (A), Visakhapatnam is a record of bona fide work carried out under my guidance and supervision.

Project Guide

Dr. Ashok Kumar N
Associate Professor
Department of E.C.E
ANITS (A)

Associate Professor
Department of E.C.E.
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

Head of Department

Dr. B. Jagadeesh
Professor & HOD
Department of E.C.E
ANITS (A)

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr. Ashok Kumar N,** Associate Professor, Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement.

We are grateful to **Dr. B. Jagadeesh**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation in carrying out this work.

We express our thanks to all **Teaching Faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all Non- Teaching Staff** of the Department of ECE, ANITS, for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

B. Sai Sampath (319126512070)

M. Satya Sudha (319126512098)

G. Sai Jashwanth (319126512113)

P. Akhila Princy (319126512106)

ABSTRACT

A significant problem in many fields is how to navigate, track, and position an object. Global Positioning System (GPS) is the best resolution to this problem. Since GPS is a form of wireless communication in space, the ephemeris data that have been received are erroneous. Therefore, the main challenge is to separate the unique ephemeris data from this inaccurate information. For this problem as well as the nonlinear system processing, adaptive algorithms deliver better results. Better outcomes are also produced by metaheuristic algorithms, but the processing time is much longer.

In this project, **Covariance Matrix Adaptation Evolution Strategy (CMA-ES)** a stochastic process and derivative free algorithm for better numerical optimization of nonlinear problems. It works on two main principles for the parameters adaptation of the search distribution. The data input device is a dual-frequency GPS receiver situated at the IISc Bangalore. To ensure accuracy, the estimated GPS receiver position is compared to the original position coordinates.

Keywords: Adaptation, Covariance, Evolution, Optimization

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

GLOBAL POSITIONING SYSTEM (GPS)

Chapter 1
Global Positioning System (GPS)

1. 1 INTRODUCTION TO GPS

The Global Positioning System (GPS), originally NAVSTAR GPS, is a government-owned, space-based radio navigation system that the United States Air Force manages. A GPS receiver can receive geolocation and time data from the global navigation satellite system from any point on or near the Earth where there is an unhindered line of sight to four or more GPS satellites. The GPS Project was launched by the US Department of Defense in 1973 for use by the United States military and became fully operational in 1995. In the 1980s, civilian use was permitted.

One of the earliest functioning satellite positioning systems was GPS, and its advancements have fueled the development and widespread use of positioning technologies ever since. GPS will continue to be a vital part of daily life as GPS-based autonomous applications proliferate. A network of satellites called the Global Positioning System (GPS) enables extremely precise Timing, Navigation, and Positioning (TNP) measurements across the world. GPS being the first satellite positioning systems, it played a crucial role in a variety of fields, such as defence, autonomous cars, farming, and aerial or marine surveying.

A GPS device does not transmit data to satellites. To further improve position accuracy, GPS-enabled devices, such as smartphones, can additionally make use of telephonic networks and towers as well as internet connections. GPS device may provide data to these systems using these latter two systems. Because the US government owns the GPS satellite system and can selectively reject or limit network access, GPS satellite networks are developed by other countries on their own. They are:

1. China's Bei Dou Navigation Satellite System
2. Russia's Global Navigation Satellite System (GLONASS)
3. The European Union's Galileo positioning system
4. India's Indian Regional Navigation Satellite System (IRNSS), NAVIC.

1.2 HOW GPS WORKS

The space segment, control segment, and user segment are the three main sections that make up the GPS constellation, like many other GNSS constellations. US Space Force manages and operate over 30 satellites which makes the GPS space segment. These satellites transmit radio signals to control and monitoring stations on Earth, as well as directly to handlers who require highly precise satellite positioning.

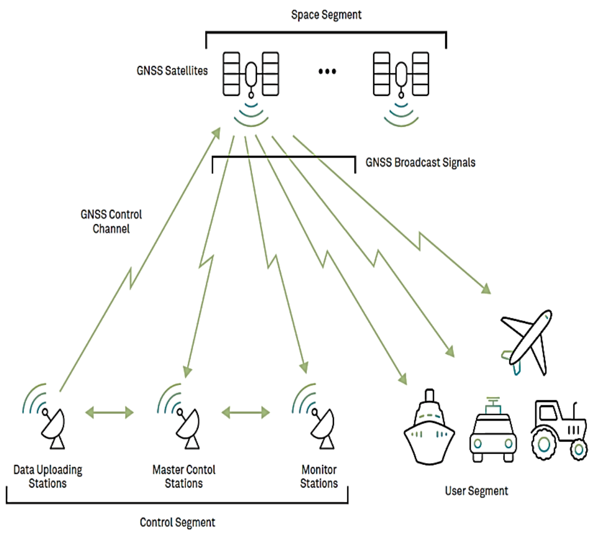

Figure 1:1 How GPS Works

The GPS control segment is managed by the US Space Force too. It has dedicated master control, ground antennas and backup control stations, and numerous monitor stations dotted around the world. These stations keep an eye on the GPS satellites to make sure they are functioning properly, are circling in the right places, and have precise atomic clocks. The general health and accuracy of the GPS constellation depend on these stations.

The user group includes anyone who utilises GPS satellites to determine PNT. Numerous applications employ GPS for very accurate location across the world, from mobile phones that provide instructions to autonomous vehicles that require lane-level positioning perfection; from farmers who track crops and harvesting routes year after year to a UAV that maps a rainforest.

1.3 MAJOR SEGMENTS OF GPS

There are three main segments in GPS.:

1. Space segment
2. Control segment
3. User segment

1.3.1 SPACE SEGMENT

24 satellites make up the space segment, which orbits the Earth at a distance of 12,000 miles. The signals can reach a wider area because of their high height. A GPS receiver on Earth is capable of receiving a signal from no less than four satellites at any given point in time because of the way the orbits of the satellites are constructed. The GPS receiver can distinguish between the signals because each satellite sends out low radio signals with a distinctive code on various frequencies. The ability to calculate the range between the satellite and the device that receives GPS is the main purpose of these coded messages. By dividing the trip period by the rate of light, the distance between the satellite and the GPS receiver is determined. Since they are faint signals that won't flow through solid objects, an unobstructed view of the sky is essential.

1.3.2 CONTROL SEGMENT

The satellites are tracked by the control segment, which then sends them updated orbital and time data. Four self-regulating stations and one master control station make up the control section. The master control station corrects the data before it is sent back to the GPS satellites. The four unmanned stations receive the information from the satellites and deliver it to it.

1.3.3 USER SEGMENT

The user segment is made up of users and their GPS receivers. The number of concurrent users is infinite.

1.4 GPS SATELLITE SIGNALS

Satellites constantly broadcast their orbital position and accurate time on radio frequencies at that location. This signal, together with at least three other satellite signals, is received by antennas and analysed in a GPS receiver to determine a user's location.

GPS broadcasts on L1 (1575.42 MHz), L2 (1227.60 MHz) and L5 (1176.45 MHz) civilian frequencies; GPS also broadcasts on L3 (1381.05 MHz) and L4 (1379.913 MHz) for governmental and regional satellite-based augmentation systems (SBAS). M-code, a military code transmitted on frequencies L1 and L2 designated for use by the United States military, is also broadcast by numerous satellites.

1.5 GPS ACCURACY

The processor in a positioning system determines its performance. A very accurate GPS receiver, for example, than a cell phone, it will be much more accurate. To improve accuracy, potential sources of error are identified and modelled at monitoring and control stations.

The majority of faults are caused by clock problems, orbital drift, atmospheric and multipath delays, and radio frequency interference. By contributing to geometric dilution of precision, positioning, navigation, and timing accuracy are constantly threatened by these sources.

Some technologies, such as satellite-based augmentation systems (SBAS), correction services for GPS, and the integration of extra sensors like radar or inertial navigation systems, help reduce the loss of precision and these errors. By calculating a position through pseudo range or carrier wave calculations, accurate GPS receivers assist in reducing errors through various algorithms.

1.6 APPLICATIONS OF GPS

GPS supports applications that rely on satellite technology for accurate timing, navigation, and positioning measurements all over the world. Despite the fact that these uses for GPS vary by industry, they are all rooted on the need for precise positioning, safe and reliable navigation, tracking and monitoring an object's movement, surveying and mapping an area, or timing to the nearest billionth of a second.

Applications like mining, for instance, to survey a location before beginning work. Companies monitor potential mineral deposits, identify which areas to avoid reducing their environmental impact, and enable autonomous machinery to transport minerals across the site.

GPS is used in conjunction with other constellations in applications that require high precision positioning. The US military, on the other hand, relies on GPS in an unusual way due to its encrypted M-code signal. M-code allows the military to maintain continuous access to positioning while also increasing resilience to potential jamming and interference sources.

1.7 TRILATERATION

A GPS gadget uses the mathematical method of trilateration to ascertain the user's position, speed, and elevation. A GPS device can calculate the precise range and the separation between each satellite being tracked constantly receiving and analysing radio transmissions of multiple using GPS satellites with triangle, circle, and sphere geometry.

1.7.1 HOW TRILATERATION WORKS

An improved form of triangulation known as trilateration does not include measurements of angles in its calculations. A point's approximate location on the surface of the Earth within a sizable circular area can be determined using data from a single satellite. The GPS can pinpoint the exact position of that point to an area where the two satellite data areas overlap by including data from a second satellite. The addition of data from a third satellite provides an accurate position on the Earth's surface.

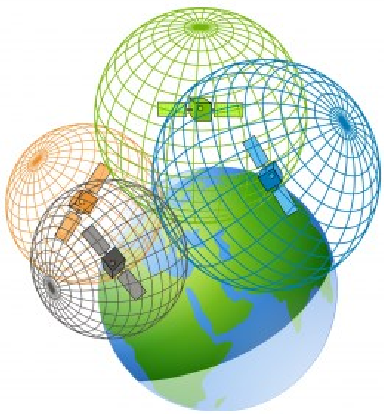

Figure 1:2 Pictorial Representation of Trilateration

All devices need 3 satellites to calculate position accurately. The location of the point is more precisely determined due to information from a fourth or even more satellites, which also makes it possible to calculate variables like altitude or, in the case of airplanes, height. GPS receivers routinely track four to seven satellites at the same time and analyse the data using trilateration.

GPS receiver uses speed equation to calculate the distance to satellites,

$$DISTANCE = SPEED \times DURATION\,(Time) \qquad (1)$$

You've probably seen advertisements for a new GPS satellite for your car that show a airplane or spaceship coming in for a landing or a helicopter flying in the air. Aren't these awesome? You've undoubtedly aware of GPS receivers, which measure the Earth's location and determine their distance from one another. But how exactly does it work?

They employ a procedure known as Trilateration. Distances are measured during trilateration. Let's dig a little deeper into this.

1.7.2 TRILATERATION MEASURES DISTANCE, NOT ANGLES

Trilateration is a technique used to determine the position of a point in space by measuring its distance from three known points. This is accomplished by measuring the time it takes for a signal to travel from the point being located to each of the known points and using this information to calculate the distance between the points.

It is important to note that trilateration measures distances, not angles. This contrasts with triangulation, which uses the measurement of angles to determine the location of a point.

Trilateration is commonly used in various applications, such as in GPS (Global Positioning System) technology to determine the location of a receiver on Earth's surface, or in surveying to measure distances between points on a terrain.

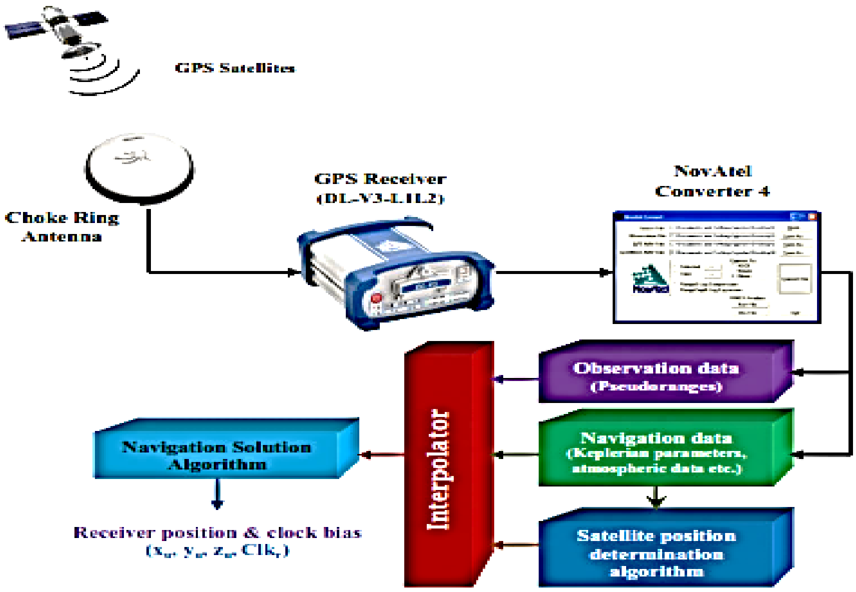

Figure 1:3 Data Collection

1.8.1 GPS SATELLITE

GPS satellite blocks are different generations of the Global Positioning System that are used for satellite navigation. GPS satellites orbit the Earth in Medium Earth Orbit (MEO) at an altitude of about 20,200 km (12,550miles). Each satellite makes two daily orbits around the Earth.

The GPS satellites are organised into six evenly spaced orbital planes that circle the Earth. Baseline satellites occupy four "slots" on each plane. This 24-slot configuration ensures that users can see at least four satellites from virtually any location on the planet.

When the baseline satellites are serviced or decommissioned, the Space Force typically flies more than 24 GPS satellites to maintain coverage. The additional satellites may improve GPS performance, but they are not part of the core constellation.

1.8.2 CHOKE RING ANTENNA

A choke ring antenna is a type of circularly polarized antenna that is used for GPS (Global Positioning System) applications. The antenna is designed to mitigate the effects of multipath interference, which can occur when GPS signals bounce off surfaces such as buildings, trees, or the ground before reaching the antenna.

The choke points act as a filter, preventing unwanted signals from entering the antenna, and reducing the effects of multipath interference. The circular shape of the antenna also helps to maintain a consistent signal quality in all directions, making it well-suited for GPS applications.

Choke ring antennas are commonly used in a variety of GPS applications, such as geodetic surveying, precision agriculture, and mapping. They are also used in aviation and maritime navigation, where accurate GPS signals are critical for safe and efficient operations.

1.8.3 GPS RECEIVER

A GPS (Global Positioning System) receiver is an electronic device that receives and processes signals from GPS satellites to determine the receiver's location, speed, and other related information.

The receiver uses a GPS antenna to receive the signals broadcast by GPS satellites. The receiver then uses specialized algorithms and processing techniques to calculate the receiver's location based on the time it takes for the signals from multiple GPS satellites to reach the receiver.

In addition to determining the receiver's location, GPS receivers can also provide information such as speed, heading, altitude, and distance traveled. They may also include additional features such as maps, route planning, and point-of-interest information.

1.8.4 NovAtel Converter

NovAtel Converter can be used in conjunction with NovAtel's Global Navigation Satellite System (GNSS) receivers and other software tools to collect high-precision GNSS data.

Here are the general steps involved in using NovAtel Converter for data collection in GPS:

- → Set up the GNSS receiver: This involves configuring the receiver settings, such as the type of GNSS signals to receive, the data rate, and the antenna type and location. This can be done using NovAtel's receiver software or a third-party software tool.

  Connect the receiver to the computer: This can be done using a USB or serial cable, depending on the type of receiver and the computer.
- → Start NovAtel Converter: The software tool should be started on the computer, and the appropriate input and output data formats should be selected. The input format should be set to NovAtel's binary format (OEM6/OEM7), and the output format should be set to the desired format for data collection, such as NMEA or RTCM.
- → Collect the data: Once NovAtel Converter is running and the GNSS receiver is connected, the software will begin collecting data in the desired format. The data can be saved to a file or streamed to a third-party software tool for further processing and analysis.

NovAtel Converter can also be used to process and filter the data in real-time, which can enhance the accuracy and dependability of GNSS data. This can be particularly important in applications requiring precise positioning, such as surveying and mapping.

In summary, NovAtel Converter is a software tool that can be used in conjunction with NovAtel's GNSS receivers and other software tools to collect high-precision GNSS data in a variety of formats. It provides a flexible and reliable solution for data collection and processing in GPS applications.

1.8.5 OBSERVATION DATA

Observation data from satellites typically consists of a range of measurements and signals that are transmitted from the satellite to ground-based receivers. The specific types of data and signals that are transmitted can vary depending on the type of satellite and the purpose of the mission.

1.8.6 NAVIGATION DATA

Navigation data from satellites consists of information that is transmitted from the satellite to ground-based receivers, and that is used for precise positioning, navigation, and timing. The specific types of navigation data that are transmitted can vary depending on the type of satellite and the purpose of the mission.

1.8.7 INTERPOLATOR

An interpolator is a tool used to estimate a value between two known values. In the context ofGPS data collection, an interpolator can be used to estimate the location of a GPS receiver at a time for which no direct measurement was made.One common type of interpolator used in GPS data collection is a linear interpolator. A linearinterpolator assumes that the location of the GPS receiver changes at a constant rate betweentwo known measurements. The linear interpolator then estimates the location of the receiverat a given time based on this assumption.Other types of interpolators that can be used in GPS data collection include splineinterpolators and kriging interpolators. These interpolators use more complex algorithms toestimate the location of the GPS receiver at a given time.

1.9 WHAT IS PSEUDORANGE

The distance between a satellite and a GNSS receiver is approximated by the pseudo range. When positional data is transmitted, a GNSS receiver will attempt to measure the ranges of (at least) four satellites as well as their positions.

The basic observed pseudorange equation is given by

$$P_{rng} = \rho_{geo} + c(dt_r - dt_s) + I_{del} + T_{del} + \varepsilon_{mul} \qquad (2)$$

Simplified pseudorange equation is given by,

$$P_{rng} = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2 + (z_s - z_u)^2} + Clk_r \qquad (3)$$

Where, $Clk_r = c \cdot (dt_r)$

The observed pseudorange equations from **i** number of GPS satellites are given by,

$$P^1_{rng} = \sqrt{({x^1}_s - x_u)^2 + \left({y^1}_s - y_u\right)^2 + ({z^1}_s - z_u)^2} + Clk_r$$

$$P^2_{rng} = \sqrt{({x^2}_s - x_u)^2 + \left({y^2}_s - y_u\right)^2 + ({z^2}_s - z_u)^2} + Clk_r$$

$$P^3_{rng} = \sqrt{({x^3}_s - x_u)^2 + \left({y^3}_s - y_u\right)^2 + ({z^3}_s - z_u)^2} + Clk_r$$

$$\vdots \qquad \vdots \qquad \vdots \qquad \vdots \qquad \vdots$$

$$P^i_{rng} = \sqrt{({x^i}_s - x_u)^2 + \left({y^i}_s - y_u\right)^2 + ({z^i}_s - z_u)^2} + Clk_r \qquad (4)$$

Pseudorange is used because the calculated distance is not a true, precise range between the two points, but rather an estimate that is subject to errors and variations caused by a variety of factors, including atmospheric conditions, satellite orbit errors, and clock errors in the receiver and the satellite.

To calculate pseudorange, the GPS receiver measures the time delay between the transmission of a GPS signal by a satellite and its reception by the receiver. The time delay is then multiplied by the speed of light to obtain an estimate of the distance between the satellite and the receiver. This estimated distance is called the pseudorange.

Pseudorange is expressed in units of time (nanoseconds or microseconds) or distance (meters or feet), depending on the convention used by the receiver.

To account for variations in the GPS signal caused by the ionosphere and troposphere, the pseudorange is typically corrected using a process called differential GPS. This involves comparing the pseudorange of a GPS receiver to the pseudorange of a reference station with a known location, and applying a correction factor to the receiver's pseudorange based on the difference between the two.

In addition to pseudorange, GPS receivers also measure other parameters related to the GPS signal, such as carrier phase and signal strength. These measurements can be used to further refine the accuracy of the GPS position calculation and precise point positioning.

CHAPTER 2

ADAPTIVE OPTIMIZATION TECHNIQUES

Chapter 2
Adaptive Optimization Techniques

2.1 LEAST SQUARES ALGORITHM

2.1.1 INTRODUCTION

Carl Fredrich Gauss invented the method of least squares and the normal distribution in 1795, when he was 18 years old, to investigate celestial body positions that are prone to unpredictable measurement errors.

The Least squares method is implemented in the calculation of the receiver coordinates obtained from the pseudo ranges for every epoch. By means of pseudo-range data, we have position with less precision and more distortion. Every satellite sends a signal at a specific time t_{sv}, which the receiver receives later t_u. If the user clock is perfectly in sync with the satellite clock, the distance travelled is $c(t_u - t_{sv})$. Hence range measured is $\rho = c(t_u - t_{sv})$. From practical view, obtaining the correct time from the user or satellite is difficult. The actual satellite clock time t'_{sv} and actual user clock will be t'_u.

Then range measured will be

$$t'_{sv} = t_{sv} + dt \quad (5)$$

$$t'_u = t_u + dT \quad (6)$$

Where dt and dT are offsets of satellite and receiver clocks respectively.

In this part, we'll examine how to infer a constant's value from a number of erratic measurements of it. Consider the case where we are dealing with a resistor but are unsure about its resistance. We use a multimeter to measure its resistance several times, but because we are using a cheap multimeter, the measurements are noisy. We aim to calculate the resistance based on our noisy measurements. We would like to calculate a constant scalar in this situation, but we could also estimate a constant vector. To put it mathematically, consider x to be n-element vector that is constant but unknown, and y as a noisy measurement vector with k elements. How do we determine the "best" estimate h of x? Assume that each measurement vector y element is created by linearly combining the x elements with some measurement noise:

$$
\begin{aligned}
y_1 &= H_{11}x_1 + \ \dots + H_{1n}x_n + v_1 \\
&\vdots \qquad\qquad\quad \vdots \qquad \vdots \\
&\vdots \qquad\qquad\quad \vdots \qquad \vdots \\
y_k &= H_{k1}x_1 + \ \dots + H_{kn}x_n + v_k
\end{aligned} \tag{7}
$$

This group of equations may be transformed into a matrix as

$$y = Hx + v \tag{8}$$

Now define $\in_y$ as the difference between the noisy measurements and the vector $H\hat{x}$:

$$\in_y = y - H\hat{x} \tag{9}$$

$\in_y$, is called the measurement residual. According to Karl Gauss [Gau04], the most probable value of the vector x is the vector $\hat{x}$ that minimizes the sum of squares between the observed values y and the vector $H\hat{x}$. As a result, let us compute the h that reduces the cost function J, J is given as

$$
\begin{aligned}
J &= \epsilon_{y1}^2 + \cdots + \epsilon_{yk}^2 \\
&= \epsilon_y^T \in_y
\end{aligned} \tag{10}
$$

J is frequently referred to as a cost function, objective function, or return function in control and estimation books and papers. We can substitute for $\in_y$ in the above equation to rewrite J as

$$
\begin{aligned}
J &= (y - H\hat{x})^T(y - H\hat{x}) \\
&= y^T y - \hat{x}^T H^T y - y^T H\hat{x} - \hat{x}^T H^T H\hat{x}
\end{aligned} \tag{11}
$$

To minimize J in terms of $\hat{x}$, we calculate the partial derivative and set to zero:

$$
\begin{aligned}
\frac{\partial J}{\partial \hat{x}} &= -y^T H - y^T H + 2\hat{x}^T H^T H \\
&= 0
\end{aligned} \tag{12}
$$

Solving this equation for $\hat{x}$ results in

$$H^T y = H^T H\hat{x} \quad (13)$$

$$\hat{x} = (H^T H)^{-1} H^T y \quad (14)$$

2.1.2 LEAST SQUARES ALGORITHM PSEUDOCODE

Pseudo ranges are given manually

Satellite positions(spos) are taken for an epoch

```
for j = 1: number of iterations
        for i = 1: number of satellites
                H(i,:) = [(spos(i,1)-xu)/pr(i), (spos(i,2)-yu)/pr(i),(spos(i,3)-
zu)/pr(i)]
        end
        x̂ = (H^T H)^-1 H^T y
        x̂_u = x_u + x̂(1)
        ŷ_u = y_u + x̂(2)
        ẑ_u = z_u + x̂(3)
end
```

2.1.3 LEAST SQUARES ALGORITHM FLOWCHART

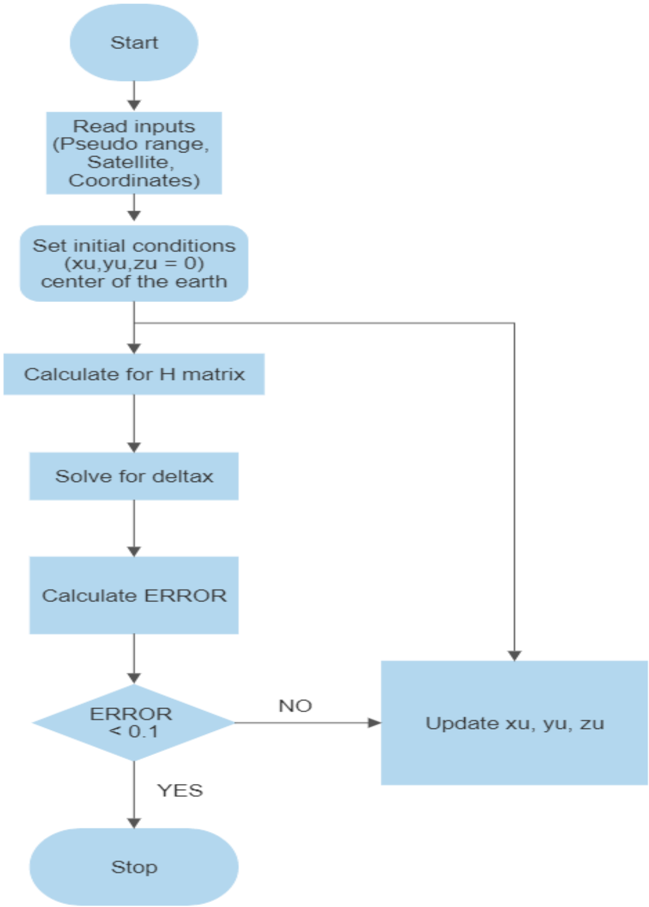

Figure 2:1 Least Square Algorithm Flowchart

2.1.4 LEAST SQUARES ADVANTAGES AND DISADVANTAGES

Advantages:

1. Accuracy: The least squares algorithm is a highly accurate method for determining the position of a GPS receiver. It can achieve centimetre-level accuracy in ideal conditions, and sub-meter accuracy in more challenging environments.
2. Robustness: The least squares algorithm is a robust method that can handle a wide range of measurement errors and signal interference. It can work well in challenging environments where other positioning methods may struggle.
3. Flexibility: The least squares algorithm is a flexible method that can incorporate different types of measurements, such as pseudo range and carrier

phase, to improve accuracy.

4. Availability: The least squares algorithm is widely used in GPS receivers and is supported by many GPS processing software packages, making it a readily available and widely accessible method.

Disadvantages:

1. Computationally intensive: The least squares algorithm requires significant computational resources to process the large amount of data involved in GPS positioning. This can make it difficult to implement in resource-constrained environments.
2. Limited in dynamic applications: The least squares algorithm assumes a stationary receiver and is not well suited for dynamic applications, such as moving vehicles or airborne platforms. Alternative methods, such as Kalman filtering, are often used in these applications.
3. Vulnerability to multipath: The least squares algorithm is vulnerable to errors caused by multipath, which occurs when GPS signals reflect off nearby surfaces and arrive at the receiver at different times. Multipath can cause large errors in pseudo range measurements, which can affect the accuracy of the least squares algorithm.
4. Dependence on satellite availability: The least squares algorithm requires signals from multiple GPS satellites to accurately determine the position of a receiver. In environments where satellite visibility is limited, such as urban canyons or dense foliage, the accuracy of the algorithm can be significantly reduced.

2.1.5 LEAST SQUARES ALGORITHM APPLICATIONS

The least squares algorithm is a statistical method used to fit a linear equation to a set of data points. It is widely used in various fields for different applications, some of which are:

1. GPS Positioning: The least squares algorithm is widely used in GPS positioning to determine the location of a receiver based on measurements of

GPS signals from multiple satellites. It is used to calculate the position of the receiver based on the pseudo range measurements and known positions of the satellites.

2. Image and Video Processing: The least squares algorithm is used in image and video processing to estimate parameters of a model from noisy data. It is used in a variety of applications, such as image compression and restoration, face recognition, and motion tracking.
3. Regression analysis: The least squares algorithm is commonly used in regression analysis to estimate the relationship between an independent variable and a dependent variable. This is commonly used in finance, economics, and social sciences.
4. Time series analysis: The least squares algorithm is used in time series analysis to forecast future values of a variable based on its past values.
5. Machine learning: The least squares algorithm is used in some machine learning algorithms, such as linear regression and support vector regression, to predict continuous variables based on input features.

Overall, the least squares algorithm is a powerful and widely used method that has applications in many fields, including GPS positioning, image and video processing, system identification, machine learning, and finance.

2.2 WEIGHTED LEAST SQUARES ALGORITHM

2.2.1 INTRODUCTION

Weighted least squares is an extension of the traditional least squares method that dates to the early nineteenth century. Adrien-Marie Legendre, a French mathematician and astronomer, invented the method in 1805, and Carl Friedrich Gauss refined it in 1809.

The original method of least squares developed by Legendre and Gauss assumed that all data points had equal weight, or in other words, that the measurement errors were normally distributed with equal variance. However, it was soon recognized that this assumption was not always appropriate, as different data points may have different levels of measurement error or uncertainty.

The concept of using weights in least squares was first introduced by the British statistician Francis Galton in the late 19th century. Galton suggested using weights that were proportional to the inverse of the variance of the measurement error, to give more importance to the more reliable data points. This idea was later formalized by the German mathematician Felix Klein in 1893, who introduced the term "weighted least squares" and provided a general framework for the method.

It is a variation of the least squares method that is commonly used in situations where the data have different levels of uncertainty or noise. In traditional least squares, all data points are treated equally, and the objective is to minimize the sum of the squared errors between the observed data and the model prediction. However, in many cases, some data points are more reliable than others, and the least squares method can lead to biased results.

Weighted least squares address this issue by assigning weights to each data point, which reflect the level of uncertainty or noise in the data. The objective of the weighted least squares method aims to reduce the weighted sum of squared errors between the observed data and the predicted value of the model. The weight of each data point is inversely proportional to the variance of the measurement error, with the more reliable data points given a higher weight.

The use of weights in weighted least squares allows the algorithm to give more importance to the more reliable data points, while reducing the influence of the less reliable data points. This results in more accurate parameter estimates and better model fits, particularly when dealing with noisy data.

Weighted least squares can be used in a variety of applications, including regression analysis, time series analysis, and machine learning. It is particularly useful in situations where the data have significant measurement errors or where the quality of the data varies across the dataset.

The RLS algorithm is an extension of the classical least squares method used to guesstimate the parameters of a linear regression model.

The RLS algorithm can be used to estimate the parameters of a wide range of linear models, including autoregressive moving average (ARMA) models, linear prediction models, and adaptive filters. The algorithm is highly adaptable and can be modified to handle a variety of constraints and objectives.

Weighted least squares (WLS), also known as weighted linear regression, is a generalization of ordinary least squares and linear regression that incorporates knowledge of the variance of observations. WLS is also a subset of generalized least squares.

The ordinary least squares method assumes that the error variance is constant (which is called homoscedasticity). When the ordinary least squares assumption of constant variance in errors is violated, the weighted least squares method can be used (which is called heteroscedasticity). The model being considered is.

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{R} \tag{15}$$

In this algorithm, generalizing the results of least squares algorithm to obtain the weighted least squares estimation.

$$\begin{bmatrix} \delta P_1 \\ \delta P_2 \\ \delta P_3 \\ \delta P_4 \\ \vdots \\ \delta P_n \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12}\,\alpha_{13} & 1 \\ \alpha_{21} & \alpha_{22}\,\alpha_{23} & 1 \\ \alpha_{31} & \alpha_{32}\,\alpha_{33} & 1 \\ \alpha_{41} & \alpha_{42}\,\alpha_{43} & 1 \\ \vdots & \vdots \quad \vdots & \vdots \\ \alpha_{n1} & \alpha_{n2}\,\alpha_{n3} & 1 \end{bmatrix} \begin{bmatrix} \delta x_u \\ \delta y_u \\ \delta z_u \\ \delta b_u \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ \vdots \\ v_n \end{bmatrix}$$

The measurement noise variance may differ for each element of P.

$$R = E(vv^T) = \begin{bmatrix} \sigma_1^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_n^2 \end{bmatrix} \quad (16)$$

We will now minimize the following quantity in relation to $\hat{x}$

$$J = \epsilon_{p1}^2 / \sigma_1^2 + \cdots + \epsilon_{pn}^2 / \sigma_n^2 \quad (17)$$

$$\frac{\partial J}{\partial \hat{x}} = -y^T R^{-1} H + \hat{x}^T H^T R^{-1} H = 0 \quad (18)$$

$$\hat{x} = (H^T R^{-1} H)^{-1} H^T R^{-1} \text{y} \quad (19)$$

2.2.1 WEIGHTED LEAST SQUARES ALGORITHM PSEUDOCODE

Pseudo ranges are given manually

Satellite positions(spos) are taken for an epoch

```
for j=1: number of iterations
   for i=1: number of satellites
      h(i,:) = [(spos(i,1)-xu)/pr(i), (spos(i,2)-yu)/pr(i),(spos(i,3)-zu)/pr(i)];

      g(i,:) = sqrt(((spos(i,1)-xu)^2)+((spos(i,2)-yu)^2)+((spos(i,3)-zu)^2));
      delp(i,:) = [pr(i) - g(i)];
   end
   take r as covariance matrix

   K=[(pinv(h.'*pinv(r)*h))*h.'*pinv(r)*delp];

   xu = xu + K(1);
   yu = yu + K(2);
   zu = zu + K(3);

end
```

2.2.3 WEIGHTED LEAST SQUARES ALGORITHM FLOWCHART

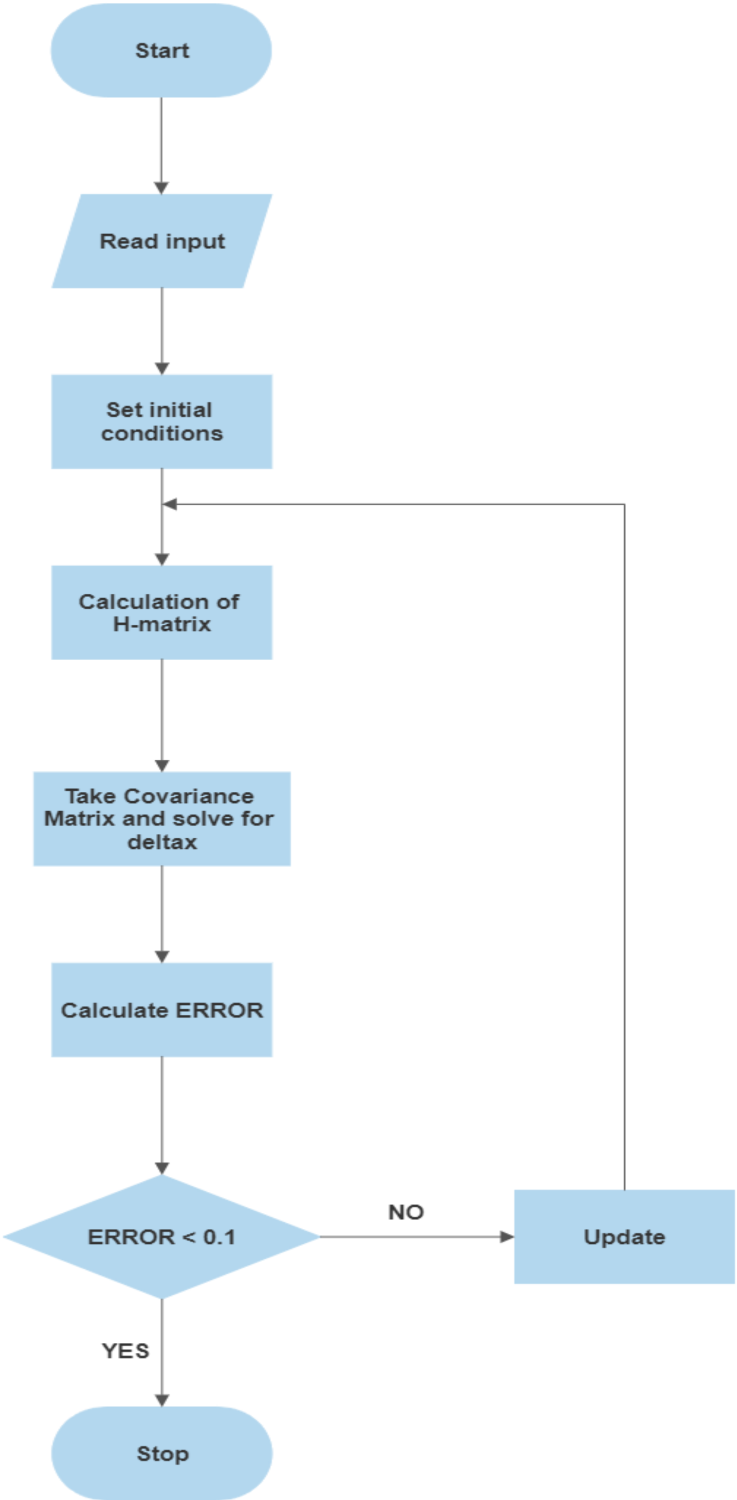

Figure 2:2 Weighted Least Squares Algorithm Flowchart

2.2.4 WEIGHTED LEAST SQUASRES ADVANTAGES AND DISADVANTAGES

Advantages

1. Better accuracy: Weighted least squares provides better accuracy in estimating the receiver position by giving more weight to the more reliable measurements and less weight to the less reliable measurements. This results in more accurate parameter estimates, and a better model fit, especially when dealing with noisy data.
2. Better robustness: Weighted least squares is more robust to outliers and data points with large measurement errors. The weights allow the algorithm to reduce the influence of these data points, which can help to improve the overall robustness of the estimation process.
3. Flexibility: The use of weights in the weighted least squares algorithm makes it flexible and adaptable to different types of data and measurement errors. The weights can be adjusted to account for the level of uncertainty or noise in the data, making the algorithm suitable for a wide range of applications.

Disadvantages

1. More complex: Weighted least squares is more complex than the traditional least squares method, as it involves assigning weights to each data point. This can make the algorithm more difficult to implement and interpret, especially for users who are not familiar with statistical concepts.
2. Data requirements: Weighted least squares requires that the measurement errors of each data point are known or can be estimated. If the measurement errors are unknown or cannot be estimated accurately, then the algorithm may not be suitable.
3. Computationally intensive: Weighted least squares can be computationally intensive, especially when dealing with large datasets. The algorithm requires the inversion of a matrix, which can be time-consuming and computationally expensive for large matrices. However, there are numerical methods that can be used to speed up the computation.

2.2.5 WEIGHTED LEAST SQUARES APPLICATIONS

Econometrics: In econometrics, WLS is used to guesstimate the limitations of a regression model when the variance of the errors is not constant across observations.

This is common in finance where stock prices may have different levels of volatility.

1. Biostatistics: In biostatistics, WLS is used to analyze clinical trials where the response variable has different levels of variability across different groups. This can be due to differences in sample sizes, different subgroups, or differing baseline risk factors.
2. Survey sampling: In survey sampling, WLS is used to adjust for sampling weights that reflect the unequal probabilities of selection in a survey. This helps to produce unbiased estimates of population parameters.
3. Geostatistics: In geostatistics, WLS is used to model spatial data where the variability of the errors depends on the distance between the observations. This is useful in fields such as meteorology and environmental science.
4. Image processing: In image processing, WLS is used to remove noise from images by assigning higher weights to the pixels that are more reliable and have lower noise levels.

2.3 RECURSIVE LEAST SQUARES ALGORITHM

2.3.1 INTRODUCTION

The Recursive Least Squares (RLS) algorithm is a statistical signal processing technique that is used to recursively estimate the parameters of a linear regression model.

1805: Gauss proposed the method of least squares to estimate the parameters of a linear regression model.

1950s: The first recursive least squares algorithms were developed by researchers at Bell Labs, including Widrow and Hoff.

1980s: The RLS algorithm was extended to handle time-varying parameters and nonlinear regression models.

1990s: The RLS algorithm found new applications in machine learning, including artificial neural networks and support vector machines.

2000s: Variations of the RLS algorithm, including regularized and sparse RLS, were developed to handle ill-conditioned or underdetermined systems.

2.3.2 RECURSIVE LEAST SQUARES PSEUDOCODE

Pseudo ranges are given manually

Satellite positions(spos) are taken for an epoch

Initialise covariance matrix

```
for j=1: number of iterations
    for i=1: number of satellites
        h(i,:) = [(spos(i,1)-xu)/pr(i), (spos(i,2)-yu)/pr(i),(spos(i,3)-zu)/pr(i)];
        g(i,:) = sqrt(((spos(i,1)-xu)^2)+((spos(i,2)-yu)^2)+((spos(i,3)-zu)^2));
        delp(i,:) = [pr(i) - g(i)];
    end
    M = [xu;
       yu;
       zu;];
    initialize error covariance
    K = [pk * h.' * pinv(r)];
    X= [M + K * (delp - h * M)];
    pk = [(1 - K * h)* pk];
    xu = xu + X(1);
    yu = yu + X(2);
    zu = zu + X(3);
end
```

2.3.3 RECURSIVE LEAST SQUARES FLOWCHART

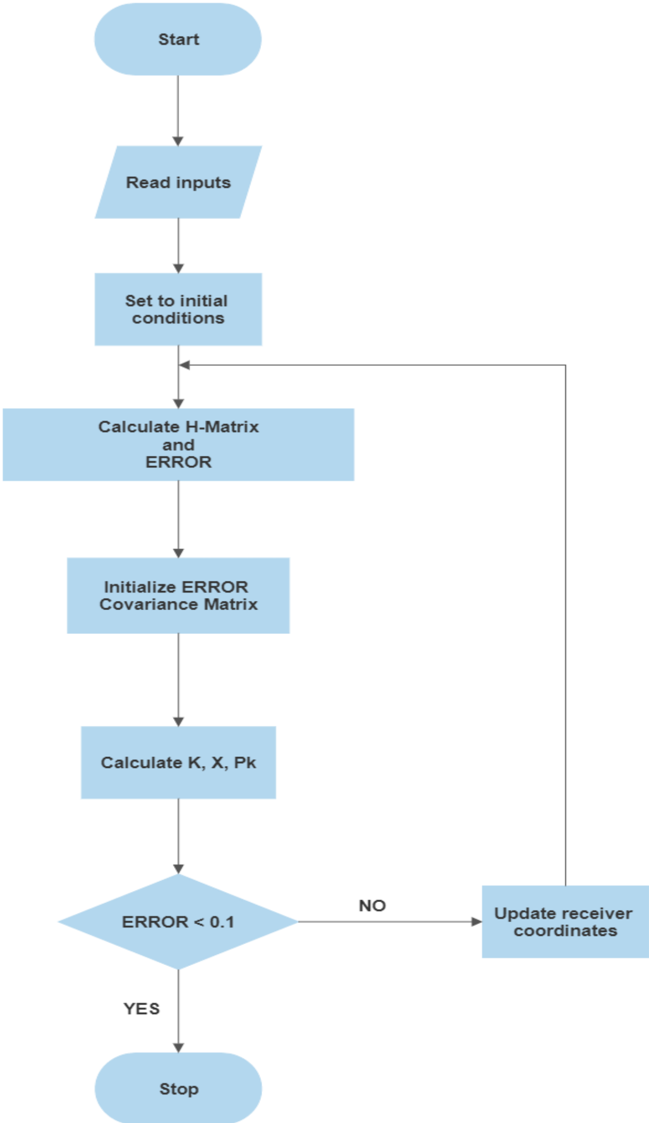

Figure 2:3 Recursive Least Squares Algorithm Flowchart

2.3.4 RECURSIVE LEAST SQUARES ADVANTAGES AND DISADVANTAGES

Advantages:

1. Real-time updates: RLS algorithm can update the parameter estimates in real-time, making it well-suited for streaming data and online applications.
2. Flexibility: RLS algorithm can be used to estimate the parameters of a wide range of linear models, including ARMA models, linear prediction models,

and adaptive filters.

3. Fast convergence: RLS algorithm can converge to the true parameter values faster than batch methods like the ordinary least squares method, making it useful in applications where fast adaptation is important.
4. Robustness: RLS algorithm can handle noisy and correlated data, making it a robust tool for modelling complex systems.

Disadvantages:

1. Memory requirements: RLS algorithm requires storing and updating the covariance matrix, which can be computationally expensive and require a lot of memory.
2. Initialization of sensitivity: RLS algorithm can be sensitive to the initial values of the parameters and covariance matrix, which can affect the performance of the algorithm.
3. Complexity: RLS algorithm can be complex and difficult to implement, particularly for non-experts in signal processing and machine learning.
4. Limited to linear models: RLS algorithm is limited to linear regression models and cannot be used for non-linear models without modifications.

2.3.5 RECURSIVE LEAST SQUARES APPLICATIONS

1. Adaptive filtering: RLS algorithm is often used for adaptive filtering applications, such as noise reduction and signal enhancement. In these applications, the RLS algorithm is used to estimate the filter coefficients in real-time, allowing the filter to adapt to changing conditions and improve the signal-to-noise ratio.
2. Control systems: RLS algorithm is used in adaptive control systems to estimate the parameters of the plant model. The algorithm can adapt the control parameters in real-time based on the measured output, allowing the system to achieve the desired response even in the presence of external disturbances.
3. Speech processing: RLS algorithm is used in speech processing applications,

such as speech recognition and synthesis. In these applications, the RLS algorithm is used to estimate the parameters of the speech model, allowing the system to generate or recognize speech signals in real-time.

4. Time-series analysis: RLS algorithm can be used in time-series analysis applications, such as stock market prediction and weather forecasting. In these applications, the RLS algorithm is used to estimate the parameters of the time-series model, allowing the system to predict future values based on past observations.
5. Machine learning: RLS algorithm is used in machine learning applications, such as linear regression and support vector machines. In these applications, the RLS algorithm is used to estimate the parameters of the model, allowing the system to learn from data and make predictions in real-time.

2.4 KALMAN FILTER

2.4.1 INTRODUCTION

Rudolf E. Kalman, a Hungarian-American electrical engineer and mathematician, invented the Kalman filter in the early 1960s. Kalman was born in Budapest in 1930 and moved to America in 1943. In 1957, he received a Ph.D. in electrical engineering from Columbia University and went on to teach at the Massachusetts Institute of Technology (MIT).

The development of the Kalman filter was motivated by Kalman's work on the Apollo program, where he was involved in the development of the guidance and navigation system for the Apollo spacecraft. The problem of estimating the state of the spacecraft based on noisy measurements was a major challenge, and Kalman realized that the existing methods were inadequate.

The Kalman filter is a mathematical algorithm that uses a series of noisy measurements to estimate the state of a system. It is a recursive algorithm that estimates the state of a system using a series of measurements and mathematical models.

The filter works by predicting the state of the system at each time step using a system model and previous state estimates, and then correcting the prediction using the measured data. The filter combines the predictions and measurements in a way that minimizes the error between the predicted and measured values, while also considering the uncertainty in both the system model and measurement data.

The Kalman filter is widely used in control systems, where it is used to estimate the state of a system and adjust the system's control inputs to achieve a desired output. It is also used in navigation systems, where it is used to estimate the position and velocity of a vehicle based on sensor measurements. In robotics, the Kalman filter is used to estimate the position and orientation of a robot based on sensor measurements, while in finance, it is used to estimate asset prices and returns based on historical data.

The filter is based on statistical theory and makes use of probability distributions to estimate the state of the system. It is designed to handle both linear and nonlinear systems and can be extended to handle non-Gaussian noise and other challenges.

The Kalman filter has become an essential tool in many fields, and its importance and versatility continue to grow as new applications are developed. It has many variations and extensions, including the extended Kalman filter (EKF) and the unscented Kalman filter (UKF), which are used to estimate the state of nonlinear systems.

The measurements of the coordinates are updated with the help of the equations below.

$$K_t = P_t H_t^T (H_t P_t H_t^T + R)^{-1} \tag{20}$$

$$\hat{x} = \hat{x_t} + K_t \hat{z_t} - H_t \hat{x_t} \tag{21}$$

$$P_t = (I - K_t H_t)\, P_t \tag{22}$$

The receiver clock is updated with the help of the below equations:

$$\hat{x} = F_t \hat{x}_{t-1} + W_t \quad (23)$$

$$P_t = F_t P_{t-1} F_t^T + Q_t \quad (24)$$

Where $\hat{x}_{t-1}$ - is a state vector to be estimated.

F_t - Transition matrix to transfer state vectors from one state to other.

W – System noise

P = State Covariance Matrix

Q – Process Noise Covariance

K = Kalman Gain

H_t= Measurement Matrix

R = Measurement of Noise Covariance

$\hat{z}$ = Vector of observed values

2.4.2 KALMAN FILTER PSEUDOCODE

Inputs: X_{est}, P_{est}, z, Q, R

Outputs: $X_{updated}$, $P_{updated}$

Step 1: Initialize F and H matrix

Predicted State vector and Covariance

$X_{prd} = F(X_{est})$

$P_{prd} = F(P_{est})(F_{transpose}) + Q$

Step 2: Estimation:

$S = H\, P_{prd}(H_{transpose}) + R$

Step 3: Compute Kalman Filter Gain factor

$K_{gain} = (P_{prd})(H_{transpose})(S_{inverse})$

Step 4: Correction based on observation

$X_{updated} = X_{prd} + K_{gain}\,(z - H\, X_{prd})$

$P_{updated} = P_{prd} - K_{gain}(H)P_{prd}$

Step 5: Return $X_{updated}$ and $P_{updated}$

Step 6: End

2.4.3 KALMAN FILTER FLOWCHART

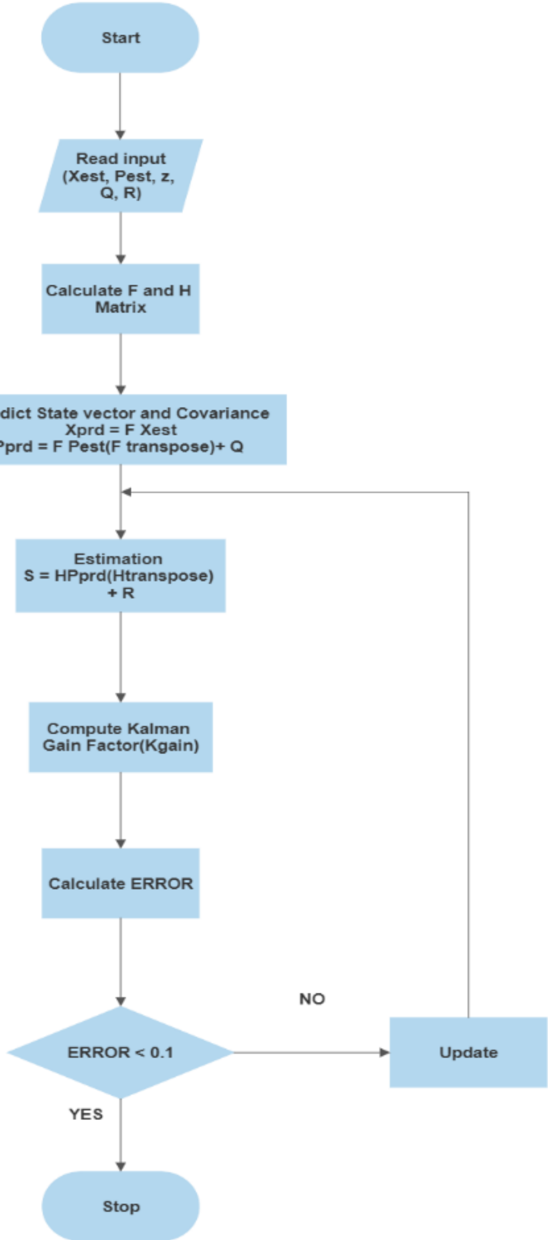

Figure 2:4 Kalman Filter Algorithm Flowchart

2.4.4 KALMAN FILTER ADVANTAGES AND DISADVANTAGES

Advantages:

1. Efficient: The Kalman filter is a computationally efficient algorithm that provides accurate estimates of the state of a system even in the presence of noise and uncertainties.

2. Recursive: The filter is a recursive algorithm, which means that it can be used to continuously update the estimates of the state of the system based on new measurements.
3. Adaptability: The Kalman filter is adaptable to different systems and can handle both linear and nonlinear models.
4. Optimal: The Kalman filter is an optimal estimator, which means that it provides the best estimate of the state of the system based on the available measurements and system model.
5. Versatile: The filter is widely used in various fields such as control systems, navigation, robotics, and finance, among others.

Disadvantages

1. Complexity: Although the Kalman filter is an efficient algorithm, it can be complex to implement and requires a good understanding of statistical theory.
2. Model dependency: The filter relies on a mathematical model of the system, and the accuracy of the estimates depends on the accuracy of the model. If the model is incorrect or incomplete, the estimates may be inaccurate.
3. Limited to Gaussian noise: The filter is designed to handle Gaussian noise and may not be suitable for systems with non-Gaussian noise.
4. Sensitive to initial conditions: The accuracy of the filter depends on the initial conditions and may require a warm-up period to achieve stable and accurate estimates.
5. Memory requirement: The filter requires storage of previous estimates and covariance matrices, which can be a memory burden for some systems.

2.4.5 KALMAN FILTER APPLICATIONS

1. Control systems: The Kalman filter is used in control systems to estimate the state of a system and adjust the control inputs to achieve a desired output. It is commonly used in aerospace, automotive, and industrial control systems.
2. Navigation: The Kalman filter is used in navigation systems, such as GPS, to calculate a vehicle's position and speed using sensor readings. It is also used

in autonomous vehicles, robotics, and aircraft navigation systems.

3. Signal processing: The Kalman filter is used in digital signal processing to remove noise from signals and improve the accuracy of measurements.
4. Finance: The Kalman filter is used in finance to estimate asset prices and returns based on historical data. It is also used in portfolio optimization and risk management.
5. Speech and image processing: The Kalman filter is used in speech and image processing to remove noise and improve the quality of signals and images.
6. Medical diagnosis: The Kalman filter is used in medical diagnosis to estimate the state of a patient based on sensor measurements, such as blood pressure, heart rate, and oxygen saturation.
7. Weather forecasting: The Kalman filter is used in weather forecasting to estimate the state of the atmosphere based on sensor measurements and mathematical models.
8. Time series analysis: The Kalman filter is used in time series analysis to estimate the underlying trend and seasonal patterns in data, and to make predictions based on historical data.
9. Robotics: The Kalman filter is used in robotics to estimate the position and orientation of a robot based on sensor measurements, and to plan and control the motion of the robot.

CHAPTER 3

METAHEURISTIC OPTIMISATION TECHNIQUES

Chapter 3
Metaheuristic Optimization Techniques

3.1 GENETIC ALGORITHM

3.1.1 INTRODUCTION

The origins of genetic algorithms can be traced back to the 1950s and 1960s, when early pioneers like John Holland and his colleagues began to investigate the idea of using computational models inspired by biological evolution to solve optimization problems.

In 1962, Holland published his book "Adaptation in Natural and Artificial Systems," which introduced the concept of the genetic algorithm and laid the foundation for much of the work that followed. Holland's work was focused on developing computational models that could simulate the process of natural selection and evolution to find optimal solutions to complex problems.

In the 1970s and 1980s, genetic algorithms gained popularity among researchers in various fields, including engineering, computer science, and economics. During this time, significant progress was made in developing more sophisticated genetic algorithms, including the use of crossover and mutation operators and adaptive selection strategies.

In the 1990s and 2000s, genetic algorithms continued to be refined and improved, with new variations and extensions being proposed for different types of problems. One major development during this time was the introduction of multi-objective optimization, which allowed genetic algorithms to simultaneously optimize multiple objectives.

Today, genetic algorithms remain a popular optimization technique and are widely used in fields such as machine learning, artificial intelligence, and engineering. The continued development and refinement of genetic algorithms and related optimization techniques promises to offer new insights and solutions to a wide range of complex problems in the years to come.

A genetic algorithm is a type of optimization algorithm inspired by the process of natural selection and evolution. It is commonly used in machine learning, artificial intelligence, and engineering applications to solve optimization problems.

A genetic algorithm works on the principle of starting with a population of randomly generated candidate solutions to a problem and then iteratively improving the quality of the solutions over many generations. The fittest individuals are chosen in each generation based on their fitness scores, which are a measure of how well they perform the desired task. These fitter individuals are then used to produce new offspring via genetic operators like crossover and mutation. The new offspring are evaluated, and the cycle is repeated until either a satisfactory solution or a termination condition is met.

The process of selection, crossover, and mutation in a genetic algorithm mimics the natural process of evolution, where organisms with favorable traits are more likely to survive and reproduce, passing their traits on to the next generation. By using these genetic operators, a genetic algorithm can explore a large search space efficiently and converge on good solutions to complex problems.

Genetic algorithms are generally used for optimization problems where there are many potential solutions and it's difficult to find the best one through brute force search.

The fitness function is a key component of a genetic algorithm, as it evaluates how well everyone in the population solves the problem. The fitness function should be designed so that individuals with better solutions have higher fitness scores.

1. Selection is the process of selecting the most fit individuals from a population to raise the next generation as parents. Tournament selection, roulette wheel selection, and rank selection are all common methods of selection.

2. Crossover is the process of combining genetic material from two parents to create new offspring. Common crossover methods include one-point crossover, two-point crossover, and uniform crossover.

3. Mutation is the process of introducing random changes to an individual's genetic material. This helps to introduce new genetic material into the population and prevent convergence to local optima. The mutation rate determines how frequently mutations occur.

4. Termination conditions for a genetic algorithm can vary. Common termination conditions include reaching a maximum number of generations, achieving a satisfactory fitness level, or exceeding a certain amount of computation time.

Genetic algorithms have been successfully applied to a wide range of optimization problems, including machine learning model parameter optimization, scheduling problems, and engineering design problems.

There are many variations and extensions of the basic genetic algorithm, such as elitist selection, adaptive mutation rates, and multi-objective optimization.

3.1.2 GENETIC ALGORITHM FLOWCHART

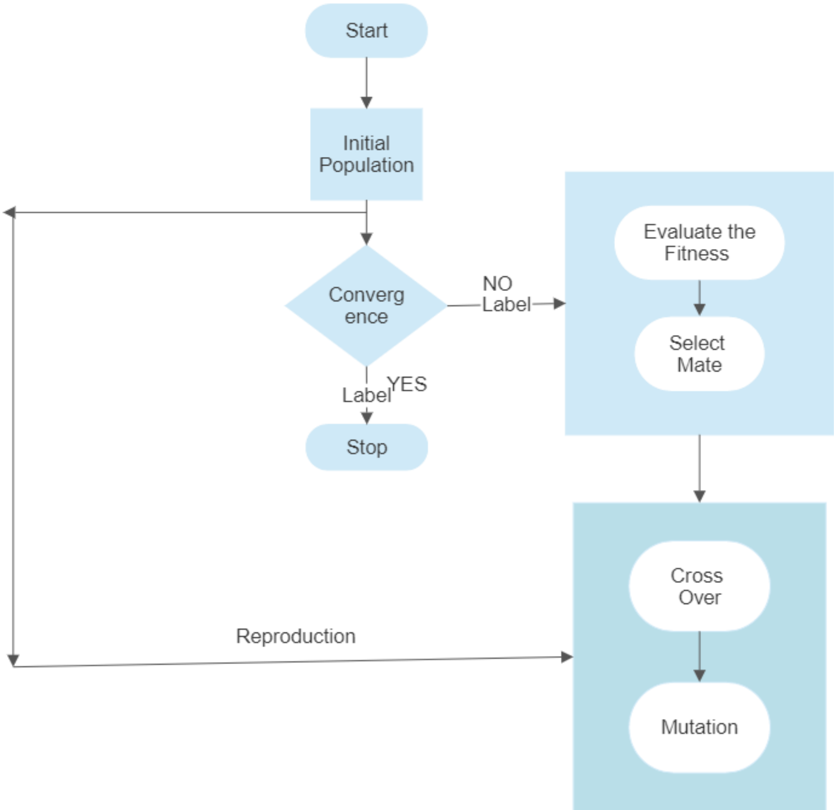

Figure 3:1 Genetic Algorithm Flowchart

3.1.3 GENETIC ALGORITHM PSEUDOCODE

1. Initialize the population with randomly generated individuals
2. Evaluate the fitness of everyone in the population
3. Repeat until the termination condition is met:
   a. Select parents for reproduction from the population based on their fitness.
   b. Create offspring by applying crossover and mutation operators to the selected parents.
   c. Evaluate the fitness of the offspring.
   d. Replace some of the least fit individuals in the population with the offspring.
   e. Return the best individual in the final population as the solution

3.1.4 GENETIC ALGORITHM ADVANTAGES AND DISADVANTAGES

Advantages

1. Exploration: GAs is good at exploring a large search space and finding good solutions in complex and diverse environments, where other search algorithms can get stuck in local optima.
2. Flexibility: Numerous optimisation issues, including those with numerous, competing objectives, can be solved with GAs.
3. Scalability: GAs can handle large-scale problems efficiently and can be easily parallelized to speed up computation.
4. Robustness: GAs is robust to noise and uncertainty in the problem, which means they can still find good solutions even when the problem is imperfectly defined, or the data is noisy.
5. Easy to implement: GAs is comparatively easy to implement and can be adapted to many different programming languages.

Disadvantages

1. Slow convergence: GAs can take a long time to converge to a good solution, especially for problems with many variables or constraints.
2. Representation bias: The effectiveness of GAs can be limited by the way the problem is represented, and it can be difficult to find a representation that works well for all problem instances.
3. Premature convergence: GAs can get stuck in a suboptimal solution if the population converges too quickly, which can happen if the crossover and mutation operators are not properly designed.
4. Parameter sensitivity: GAs is sensitive to the values of the parameters, such as the mutation rate, selection pressure, and population size, and tuning these parameters can be time-consuming and challenging.
5. No guarantee of finding the optimal solution: GAs is a heuristic search technique, which means there is no guarantee of finding the optimal solution. It is possible that the GA will find a good solution, but not the best solution.

3.1.4 GENETIC ALGORITHM APPLICATIONS

Numerous optimisation issues in several fields can be solved using genetic algorithms (GAs), including:

1. Engineering: GAs can be used for engineering design optimization, such as designing efficient mechanical structures, circuit design, or antenna design.
2. Finance: GAs can be used to optimize financial portfolios or to develop trading strategies.
3. Manufacturing: GAs can be used to optimize production schedules, material flow, and equipment selection.
4. Transportation: GAs can be used to optimize routing and scheduling in transportation networks, such as airline or shipping logistics.
5. Healthcare: GAs can be used for medical image analysis and diagnosis, as well as for optimizing treatment plans.
6. Energy: GAs can be used to optimize energy systems, such as renewable energy systems, smart grids, and power system operation.
7. Marketing: GAs can be used to optimize advertising campaigns and customer segmentation.
8. Agriculture: GAs can be used to optimize crop yields, plant breeding, and irrigation management.
9. Robotics: GAs can be used for robot path planning, sensor placement, and control optimization.
10. Game AI: GAs can be used to optimize game AI, such as developing intelligent opponents and game strategies.

3.2 FIREFLY ALGORITHM

3.2.1 INTRODUCTION

The Firefly Algorithm (FA) is a metaheuristic optimization algorithm proposed in 2008 by Xin-She Yang. The algorithm was inspired by the flashing behaviour of fireflies, which communicate using bioluminescence. Fireflies flash to attract mates or to communicate with other members of their species.

The Firefly Algorithm is a swarm-based optimization algorithm that mimics the flashing behavior of fireflies. In the algorithm, each firefly represents a potential solution to an optimization problem, and the intensity of the flashing light represents the quality of the solution. Fireflies move towards each other based on the attractiveness of their flashing light, with brighter fireflies attracting other fireflies towards them. As a result, the algorithm is able to converge to the best solution by repeatedly adjusting the position of the fireflies.

Numerous optimisation issues, including design for engineering, function optimization, processing images, and machine learning, have been successfully tackled using the Firefly Algorithm. The algorithm has proven to be successful and efficient at locating superior answers to challenging optimisation issues.

3.2.3 FIREFLY FLOWCHART

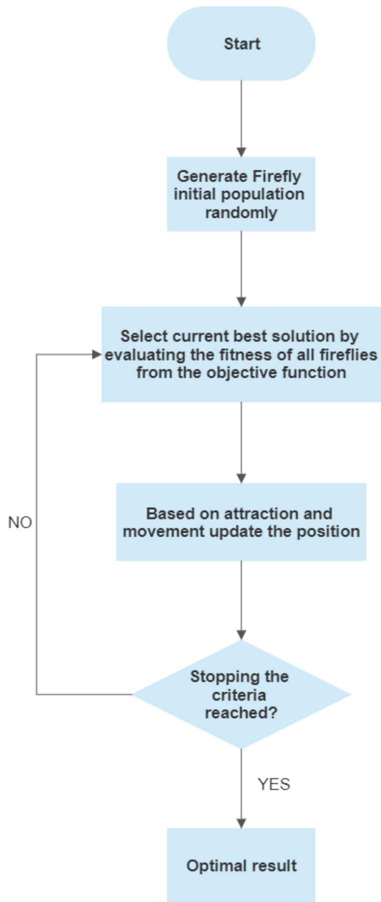

Figure 3:2 Firefly Algorithm Flowchart

3.2.2 FIREFLY ALGORITHM PSEUDOCODE

Initialization of the parameters of FA (Population size, α, β, γ and the number of iterations).

```
The cost function determines the light intensity f(x_i) where x_i(i = 1,...,n).
While (iter < Max Generation).
        for i = 1: all n fireflies
          for j = 1: all n fireflies
            if (f(xi) < f(xj)), move firefly i towards j,
            end if.
            With distance r, update attractiveness β.
            Evaluate new solution and update f(xi) in the same way as (4).
          end for j
        end for i
Sort the solutions and find the best global optimal solution.
end while.
```

3.2.3 FIREFLY EQUATIONS

The Firefly Algorithm, which was based on the flashing behaviour of fireflies, was developed by Xin-She Yang in 2008. As the number of iterations increases, the convergence can be sped up by controlling the randomness.

The following equations are used to update the firefly's position based on motion and attraction:

$$\beta = \beta_0 e^{-\gamma d^2} \tag{25}$$

$$x_i^{t+1} = x_i^t + \beta e^{-\gamma d_0^2}\left(x_j^t - x_i^t\right) + \alpha_t \varepsilon_t \tag{26}$$

Where β Coefficient base value

γ Light absorption coefficient

d is the distance between 2 consecutive fireflies

α_t is the step size controlling parameter.

ε_t is a vector of Gaussian or other distribution.

3.2.4 FIREFLY ADVANTAGES AND DISADVANTAGES

Advantages

1. Efficient: The FA is an efficient algorithm, meaning that it can find with only a small number of rounds, produce high-quality solutions to challenging optimisation issues. This is particularly true for large-scale optimization problems, where traditional optimization algorithms may struggle.
2. Versatile: Engineering design, function optimization, processing images, and machine learning are just a few of the optimisation issues that the FA can be used to solve. This versatility makes it a valuable tool for researchers and practitioners in a variety of fields.
3. Easy to implement: The FA is relatively easy to implement, with simple coding requirements and a small number of parameters to tune. This makes it accessible to researchers and practitioners with limited programming experience.
4. Robustness: The FA is a robust algorithm, meaning that it can handle noisy or incomplete data without compromising its performance. This is particularly useful in real-world applications where data may be incomplete or noisy.

Disadvantages

1. Randomness: Like many metaheuristic algorithms, the FA relies on randomness to research the search space. This can lead to inconsistent performance, particularly when the algorithm is applied to non-convex or multimodal optimization problems.
2. Parameter tuning: Although the FA has relatively few parameters to tune, finding the optimal values for these parameters can be challenging. This can be particularly true for complex optimization problems where the optimal parameter values may not be obvious.
3. Lack of theoretical guarantees: The FA does not have strong theoretical guarantees for convergence, unlike some optimization algorithms such as gradient-based methods. This means that it can be difficult to predict the performance of the algorithm for a given problem.

4. Limited scalability: Although the FA is an efficient algorithm for large-scale optimization problems, it may struggle with extremely large problems. This is particularly true when the problem involves a high-dimensional search space, where the algorithm's performance may deteriorate.

3.2.5 FIREFLY APPLICATIONS

1. Engineering Design: The FA has been used to optimize the design of various engineering systems, including mechanical, electrical, and civil engineering. For example, the FA has been used to optimize the design of aircraft wing structures, vehicle suspensions, and heat exchangers.
2. Function Optimization: The FA has been applied to various function optimization problems, including nonlinear and multimodal problems. It has been used to optimize complex functions in areas such as finance, biology, and physics.
3. Image Processing: The FA has been used to optimize image processing algorithms, such as image segmentation and feature extraction. It has also been applied to image restoration and denoising problems.
4. Machine Learning: The FA has been used to optimize the performance of machine learning algorithms, including artificial neural networks and support vector machines. It has also been used to optimize the hyperparameters of deep learning models.
5. Renewable Energy: The FA has been used to optimize the design and control of renewable energy systems, including wind turbines, solar photovoltaic systems, and fuel cells.
6. Signal Processing: The FA has been used to optimize signal processing algorithms, including speech recognition and radar signal processing.
7. Healthcare: The FA has been applied to various healthcare-related problems, including disease diagnosis and drug discovery.

CHAPTER 4

Covariance Matrix Adaptation –
Evolution Strategy (CMA-ES) Algorithm

Chapter 4
Covariance Matrix Adaptation – Evolution Strategy (CMA-ES) Algorithm

4.1 INTRODUCTION

CMA-ES (Covariance Matrix Adaptation Evolution Strategy) was developed in 2001 by Nikolaus Hansen and Andreas Oster Meier of Germany's Technical University of Berlin. Hansen and Oster Meier developed CMA-ES as an improvement over existing evolutionary algorithms, with the aim of addressing issues such as slow convergence and premature convergence in high-dimensional search spaces. The use of a covariance matrix to model the search space was the key innovation in CMA-ES. By doing so, CMA-ES can capture correlations between different variables in the search space and adapt to the structure of the problem. Since its introduction, CMA-ES has gained popularity and has been widely adopted as a powerful optimization algorithm. CMA-ES has been used in a variety of applications, including machine learning, robotics, finance, and engineering, among others. In the years following its introduction, several variations and extensions of CMA-ES have been proposed, such as the constrained CMA-ES, multi-modal CMA-ES, and the separable CMA-ES. These variants are designed to handle specific types of optimization problems and further improve the performance of CMA-ES.

In order to repeatedly produce new solutions, the CMA-ES algorithm first generates an initial population of potential solutions. This population is then subjected to an assortment of recombination, mutation, and selection procedures.

The main equations used in CMA-ES are:

1. Mean vector: The mean vector represents the current best estimate of the problem's optimal solution. At each iteration, it is updated based on the fitness values of the candidate solutions.
2. Covariance matrix: The covariance matrix represents the most recent estimate of the search distribution's shape and orientation. At each iteration, it is updated based on the fitness values of the candidate solutions.
3. Sample generation: The mean vector and covariance matrix are used as parameters to generate candidate solutions from a multivariate normal distribution.
4. Fitness function evaluation: The fitness function evaluates how well a candidate solution performs on the problem being solved. It is typically a function that takes the candidate solution as input and returns a scalar value that represents the quality of the solution.
5. Ranking and selection: The candidate solutions are ranked based on their fitness values, and the best solutions are selected to be used as the basis for the next iteration.
6. Update of mean vector and covariance matrix: Based on the candidate solutions chosen, the mean vector and covariance matrix are updated. The update rule consists of two steps: first, the mean vector is updated using the selected solutions, and then the covariance matrix is updated using the difference between the selected solutions and the current mean vector.
7. Adaptation of step size: The step size parameter is set to control the size of the search steps taken during each iteration. The step size is adjusted based on the current iteration's success rate, with the goal of maintaining a balance between search space exploration and exploitation.

The details of these equations can vary depending on the specific implementation of CMA-ES.

4.2 CMA - ES PSEUDOCODE

Input:
n: Dimensionality of the search space
f: Objective function to be minimized.
x_0: Initial guess for the search point
$sigma_0$: Initial standard deviation for the search distribution
$max_{evaluations}$: Maximum number of function evaluations allowed.
$stop_{fitness}$: Stop if objective function value is below this value.
$stop_{evaluations}$: Stop if maximum number of function evaluations is reached.
$stop_{stagnation}$: Stop if the best fitness has not improved for a certain no of iterations.

Output:

- → x_{best}: best solution found.
- → f_{best}: Objective function value of the best solution found.
- → Initialize the mean vector m and the covariance matrix C with diagonal entries of ${sigma_0}^2$.
- → Set the number of offspring, lambda, proportional to the dimensionality n.
- → Generate lambda candidate solutions z_1, ..., $z_{lambda}$from the search distribution $N\ (m, C)$.
- → Evaluate the objective function at the candidate solutions: $f(z_1)$,...,$f(z_{lambda})$
- → Sort the candidate solutions by their objective function values in ascending order.
- → Update the mean vector m as the weighted average of the lambda best solutions:

$$m\ =\ (\frac{1}{w})\ *\ sum_i = w_i\ *\ z_i$$

where $w_i\ =\ log(lambda + \frac{1}{2})\ -\ log(i)$ and w is a normalization constant.

- → Update the covariance matrix C using the formula:

$$C\ =\ \left(\frac{1}{c}\right) *\ sum_i\ =\ w_i\ *\ (z_i - m)\ *\ (z_i - m)^T$$

where c is a normalization constant, and the sum is only over the w best solutions.

- → Compute the eigenvalue decomposition of $C\ =\ B\ *\ D\ *\ B^T$.
- → Generate a new sample point by adding a weighted combination of the eigenvectors of C to the mean vector:

$$x_{new} = m\ +\ sigma\ *\ B\ *\ D^{\frac{1}{2}} *\ y$$

where $y \sim N(0, I)$ is a random vector and sigma is the step size.

- → Evaluate the objective function at the new sample point: $f(x_{new})$.
- → If any stopping criterion is met, return x_{best} and f_{best}. Otherwise, go to step 3.

4.3 CMA-ES FLOWCHART

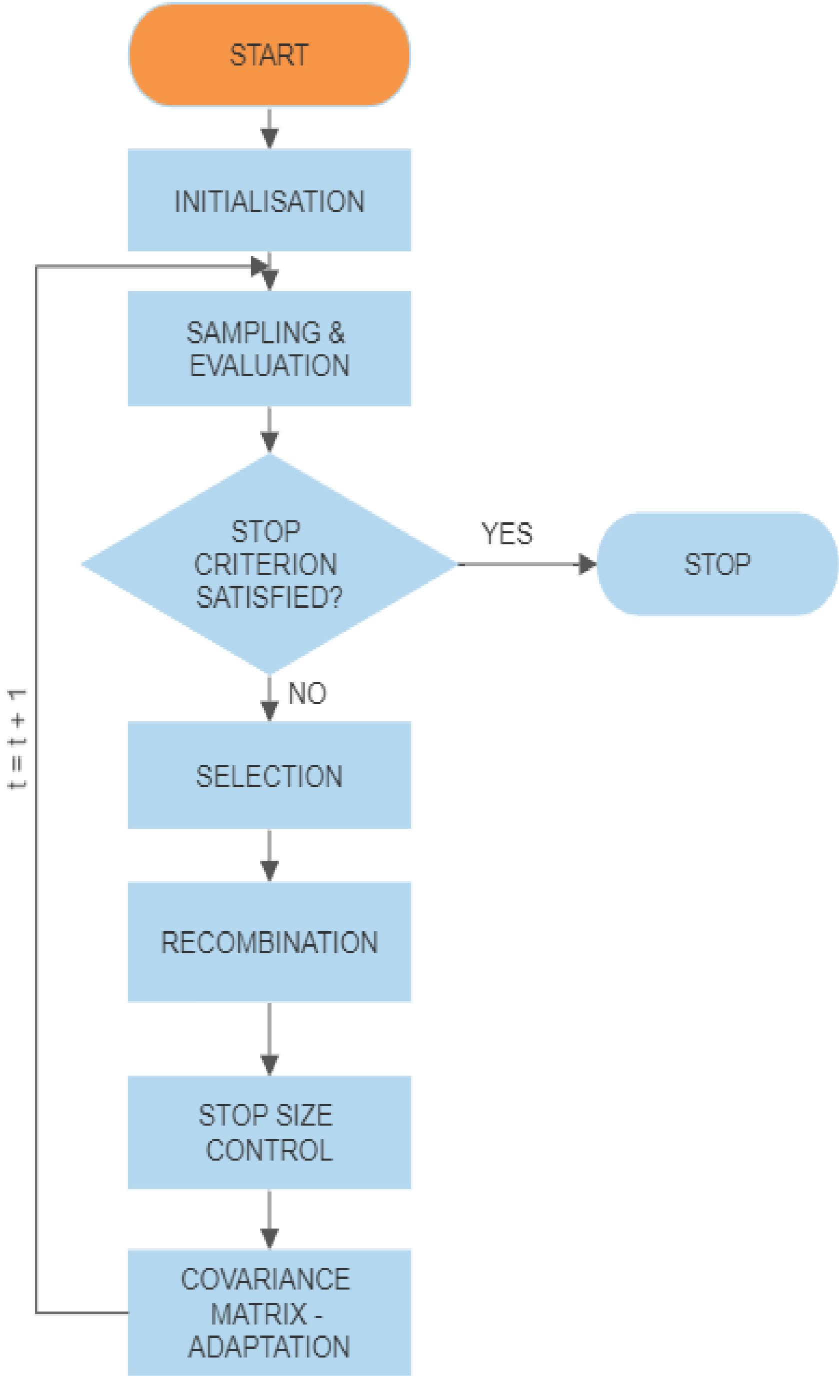

Figure 4:1 CMA-ES Algorithm Flowchart

4.4 CMA-ES ADVANTAGES AND DISADVANTAGES

Advantages

1. Effective in high-dimensional search spaces: CMA-ES is particularly effective at optimizing problems with a large number of variables or dimensions. This is because it can adapt to the structure of the search space and explore it efficiently.
2. Robust to noise: CMA-ES is designed to handle noisy objective functions, meaning it can still find good solutions even when the objective function is noisy or stochastic.
3. Efficient use of evaluations: CMA-ES uses a small number of function evaluations compared to other optimization algorithms, making it an efficient choice for expensive or time-consuming objective functions.
4. Global optimization: CMA-ES can find global optima, making it a powerful tool for solving difficult optimization problems that may have many local optima.

Disadvantages

1. Computationally intensive: CMA-ES can be computationally expensive, especially for high-dimensional search spaces, which may limit its applicability to some problems.
2. Requires tuning of parameters: CMA-ES requires tuning of its parameters, such as the population size and step size, which can be time-consuming and difficult.
3. Can get stuck in local optima: Although CMA-ES is designed to avoid local optima, it is still possible for the algorithm to get stuck in a local optimum if the initial population is not diverse enough or if the search space has multiple local optima.
4. Limited applicability to non-continuous optimization problems: CMA-ES is designed for continuous optimization problems and may not be suitable for problems with discrete variables or constraints.

4.5 CMA-ES APPLICATIONS

1. Machine learning: CMA-ES has been used to optimise machine learning hyperparameters such as neural networks, support vector machines, and decision trees. It can also be used to select features and reduce dimensionality.
2. Robotics: CMA-ES has been used to optimize the parameters of robotic systems, such as control algorithms and sensor placement. It has also been used to optimize the design of robots and to generate robot movements.
3. Engineering: CMA-ES has been used in a variety of engineering applications, including structural optimization, aerodynamic optimization, and control system design.
4. Finance: CMA-ES has been used to optimize investment portfolios and to develop trading strategies. It has also been used in risk management and option pricing.
5. Biology: CMA-ES has been used in biology to optimize the design of experiments, to fit models to data, and to analyze biological data.
6. Game theory: CMA-ES has been used to optimize strategies in games, such as poker and chess.
7. Energy optimization: CMA-ES has been used to optimize the design and control of energy systems, such as wind turbines and power grids.

CHAPTER 5

RESULTS

Chapter 5
Results

5.1 LEAST SQUARES ALGORITHM

Table 1 Estimated Position Errors by using Least Squares Algorithm Optimization

Time(hrs)	**X - Position Error (mts)**	**Y - Position Error (mts)**	**Z - Position Error (mts)**
02:00	27.56153	32.90497	5.41437
04:00	35.17906	15.28419	4.75700
06:00	44.20714	5.67702	4.19799
08:00	44.76592	4.29053	3.04549
10:00	42.12289	9.17686	2.23863
12:00	41.56468	7.82657	3.50141
14:00	36.31786	16.98890	1.72434
16:00	37.15185	12.31815	0.92316
18:00	35.73506	19.28707	1.10586
20:00	31.78404	18.74840	1.62505
22:00	34.58102	9.96644	1.08701
24:00	40.31054	9.96515	4.15698
Mean	*37.6068*	*13.536188*	*2.814774*

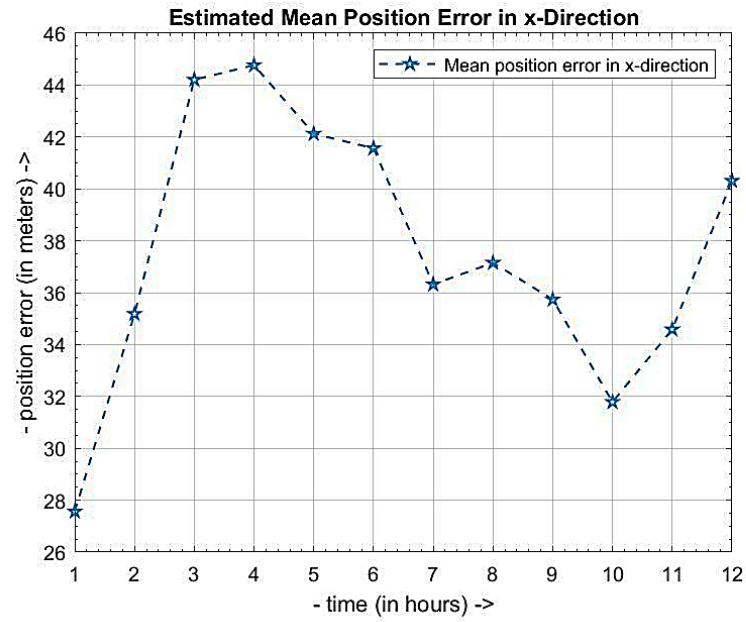

Figure 5:1 X- Position Error due to Least Square Algorithm

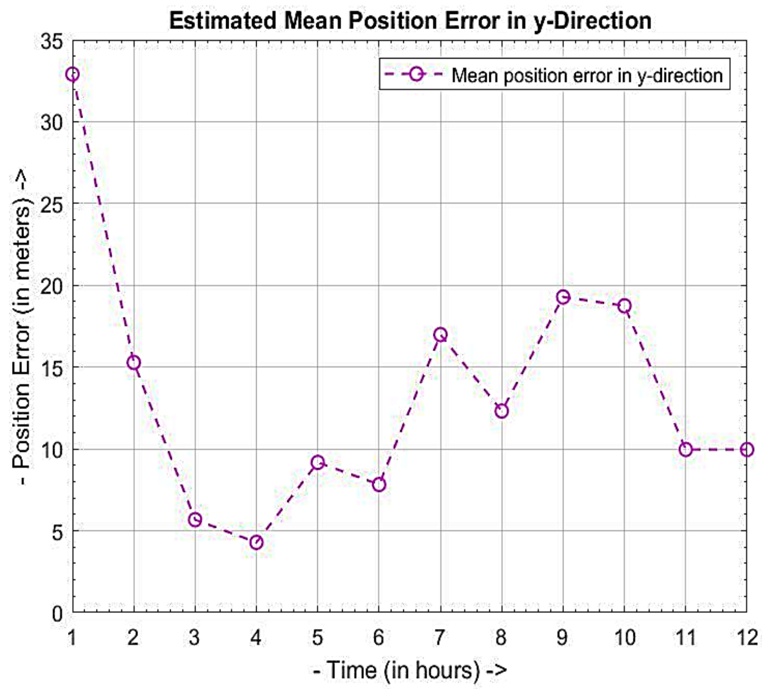

Figure 5:2 Y - Position Error due to Least Square Algorithm

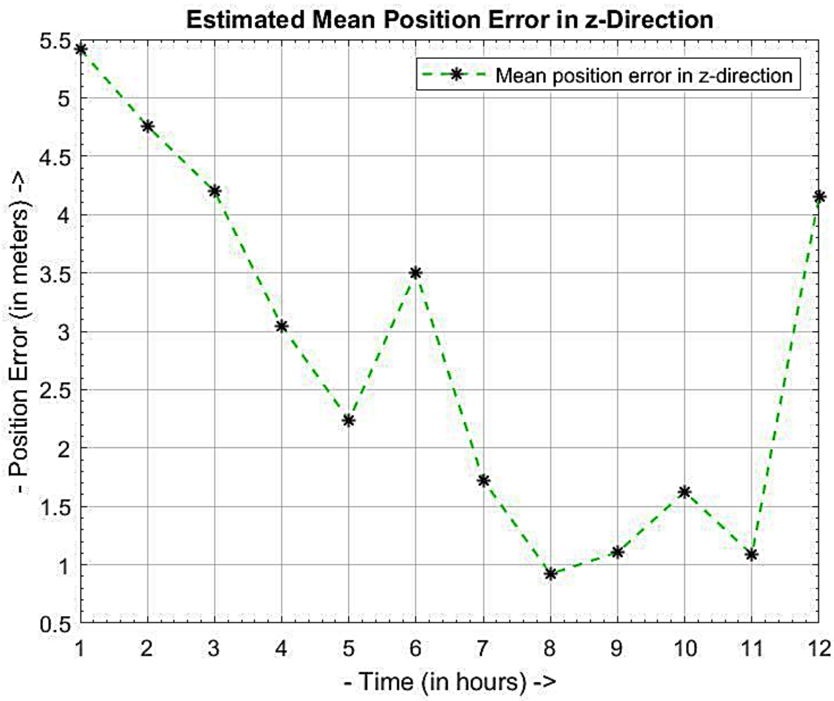

Figure 5:3 Z - Position Error due to Least Square Algorithm

In TABLE 1, the mean position errors due to Least Square Algorithm are presented. The mean position error due to Least Square Algorithm is X = *37.6068* m, Y = *13.536188* m, and Z = *2.814774* m.

5.2 KALMAN FILTER

Table 2 Estimated Position Errors by using Kalman Filter Optimization

Time(hrs)	**X - Position Error (mts)**	**Y - Position Error (mts)**	**Z - Position Error (mts)**
02:00	66.24816	176.46099	76.69777
04:00	60.45443	139.79821	51.4704
06:00	58.57798	151.43579	42.96877
08:00	54.39499	151.30843	54.37816
10:00	63.07131	152.84302	46.53391
12:00	80.51836	154.37682	54.96981
14:00	58.11651	134.1234	50.99792
16:00	64.03504	133.1206	52.9261
18:00	67.90131	128.5886	49.43039
20:00	65.60441	133.4324	50.61133
22:00	63.38104	128.8685	48.76521
24:00	57.30394	130.2463	48.39909
Mean	*63.30062*	*142.88359*	*52.34574*

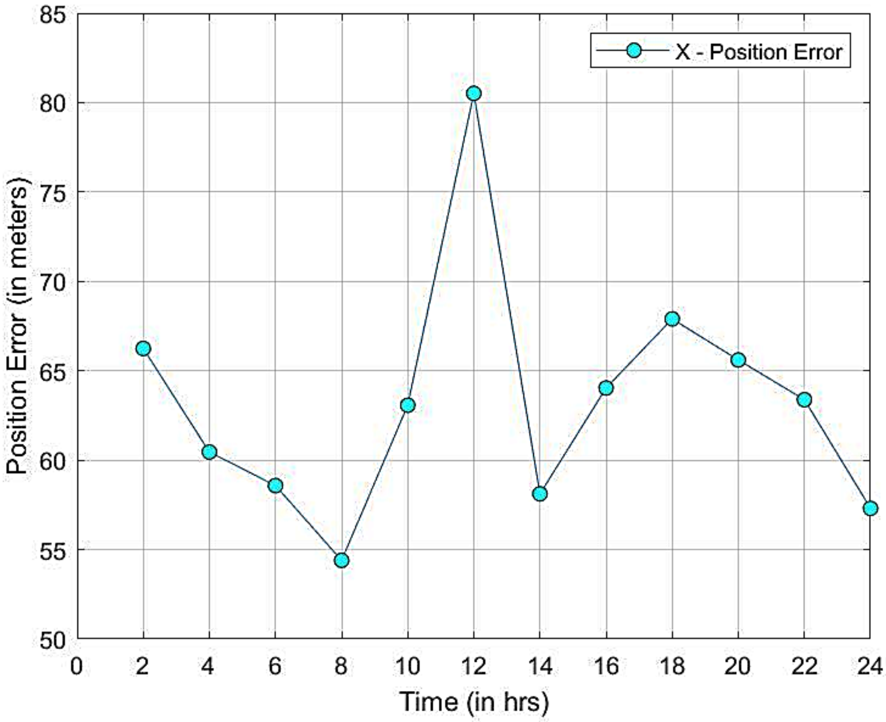

Figure 5:4 X- Position Error due to Kalman Filter Algorithm

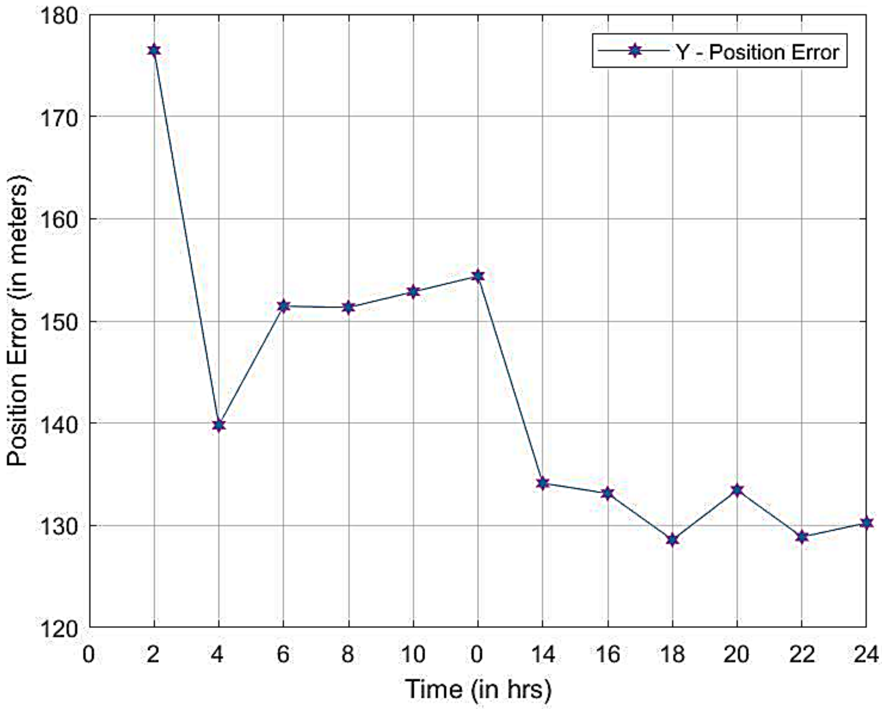

Figure 55:5 Y - Position Error due to Kalman Filter Algorithm

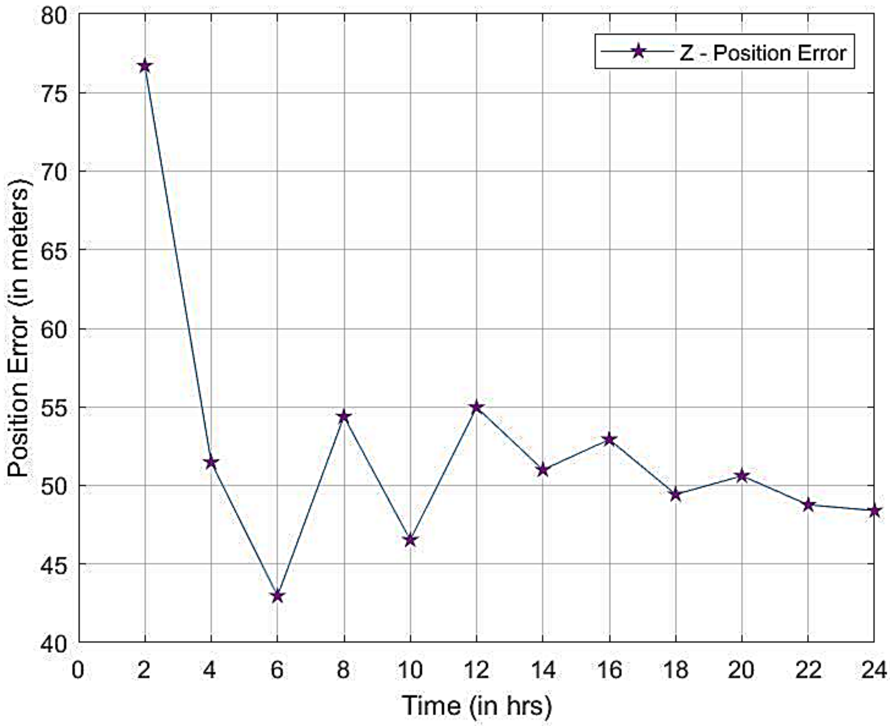

Figure 5:6 Z - Position Error due to Kalman Filter Algorithm

In TABLE 2, the mean position errors due to Kalman Filter Algorithm are presented. The mean position error due to Kalman Filter Algorithm is X = *63.30* m, Y = *142.88* m, and Z = *52.34* m.

5.3 FIREFLY ALGORITHM

Table 3 Estimated Position Errors by using Firefly Algorithm Optimization

Time (hrs)	X - Position Error (mts)	Y - Position Error (mts)	Z - Position Error (mts)
02:00	47.93834	134.7158	45.36232
04:00	49.81224	130.8506	46.1647
06:00	47.8887	130.9277	31.47469
08:00	52.10526	130.0110	43.24364
10:00	48.89236	135.6849	42.15711
12:00	48.65656	133.4049	45.97469
14:00	52.29074	134.1234	50.99792
16:00	53.21291	133.1206	52.9261
18:00	50.28349	128.5886	49.43039
20:00	52.40445	133.4324	50.61133
22:00	49.94254	128.8685	48.76521
24:00	46.53252	130.2463	48.39909
Mean	*49.99668*	*131.99789*	*46.29227*

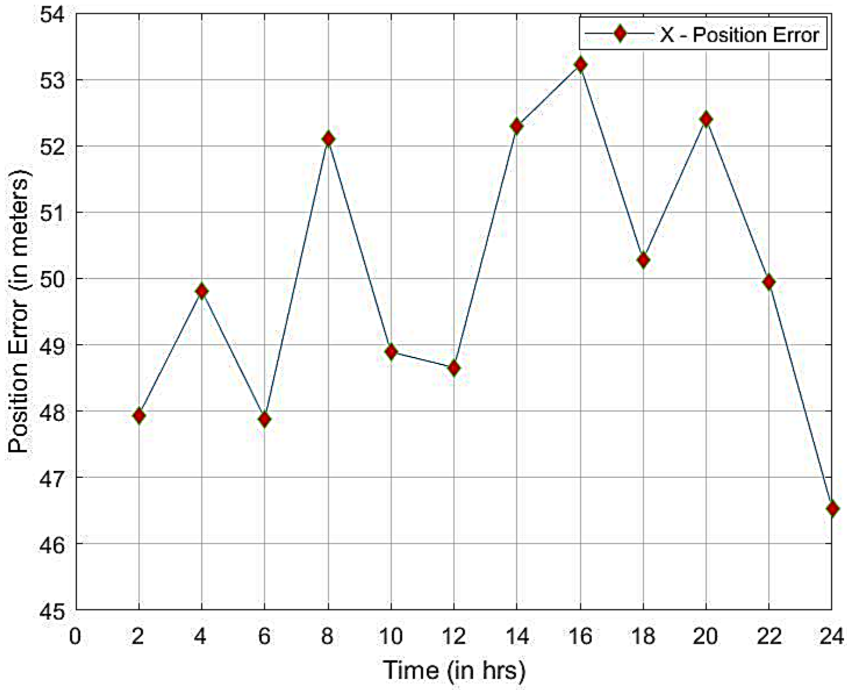

Figure 5:7 X - Position Error due to Firefly Algorithm

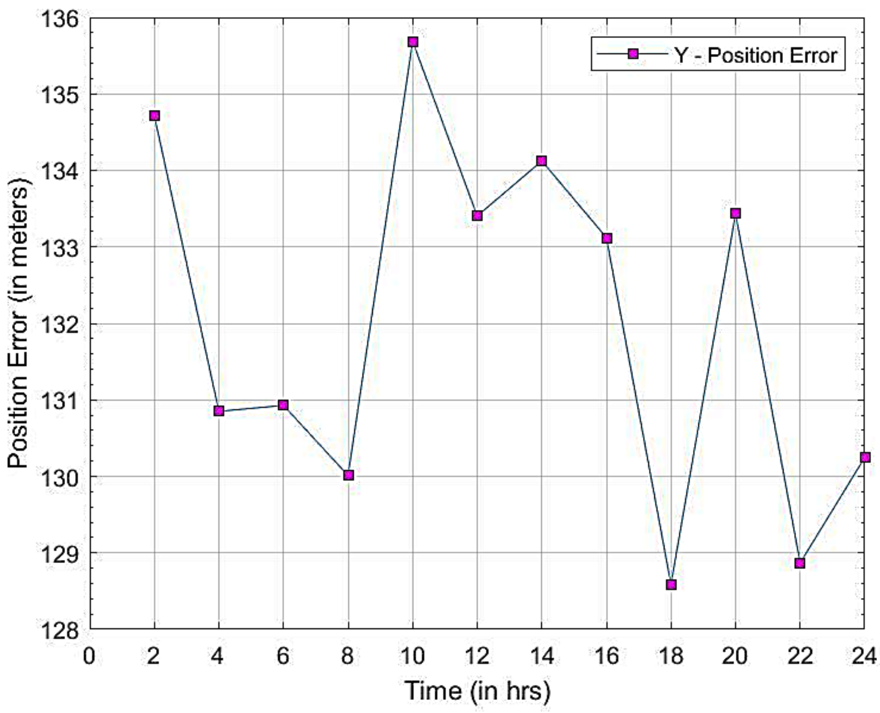

Figure 5:8 Y - Position Error due to Firefly Algorithm

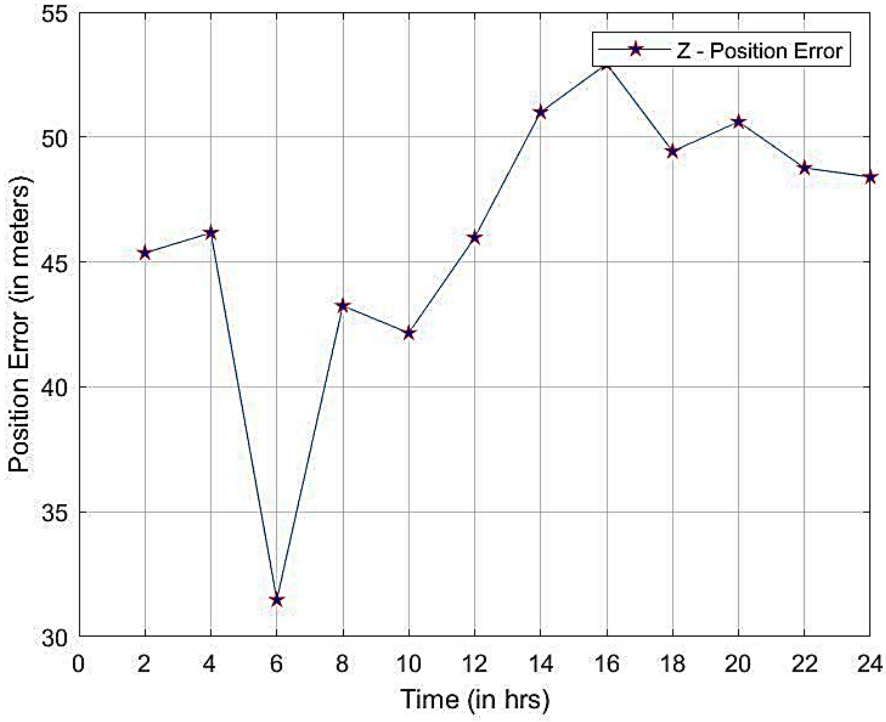

Figure 5:9 Z - Position Error due to Firefly Algorithm

In TABLE 3, the mean position errors due to Firefly Algorithm are presented. The mean position error due to Firefly Algorithm is X = *49.99* m, Y = *131.99* m, and Z = *46.29 m*.

5.4 GENETIC ALGORITHM

Table 4 Estimated Position Errors by using Genetic Algorithm Optimization

Time (hrs)	X - Position Error (mts)	Y - Position Error (mts)	Z - Position Error (mts)
02:00	34.02870	26.59867	8.90614
04:00	36.78520	19.33733	6.17722
06:00	41.16120	21.34750	4.01515
08:00	35.59528	16.19838	13.14042
10:00	27.78924	47.81619	14.47814
12:00	23.78154	56.83222	13.68455
14:00	34.15035	50.20541	14.02171
16:00	42.52991	35.48233	6.04318
18:00	39.01195	29.11703	9.97333
20:00	38.39038	24.70472	7.71369
22:00	35.87202	15.82729	11.51406
24:00	42.02838	11.24177	10.72202
Mean	*35.92701*	*29.55907*	*10.03247*

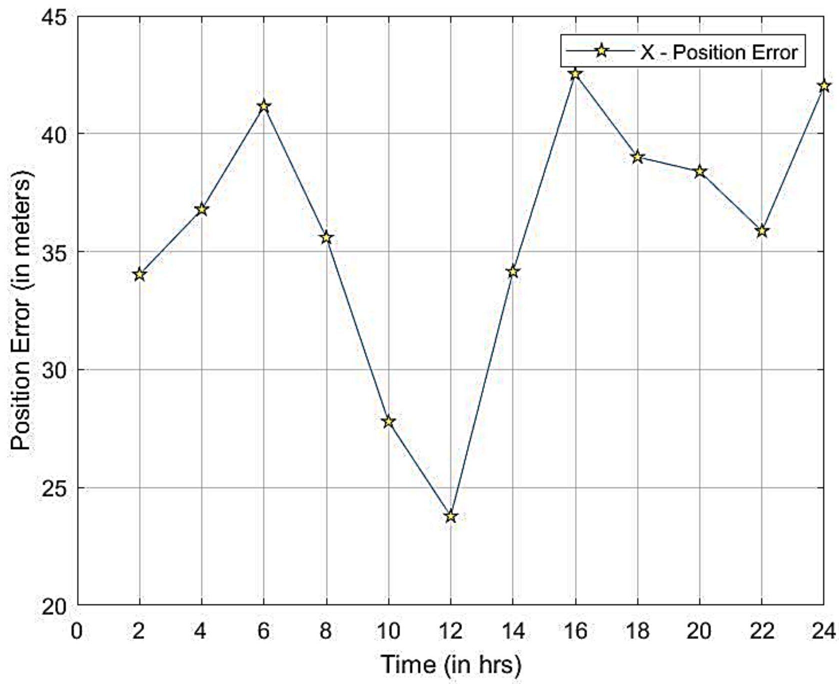

Figure 5:10 X - Position Error due to Genetic Algorithm

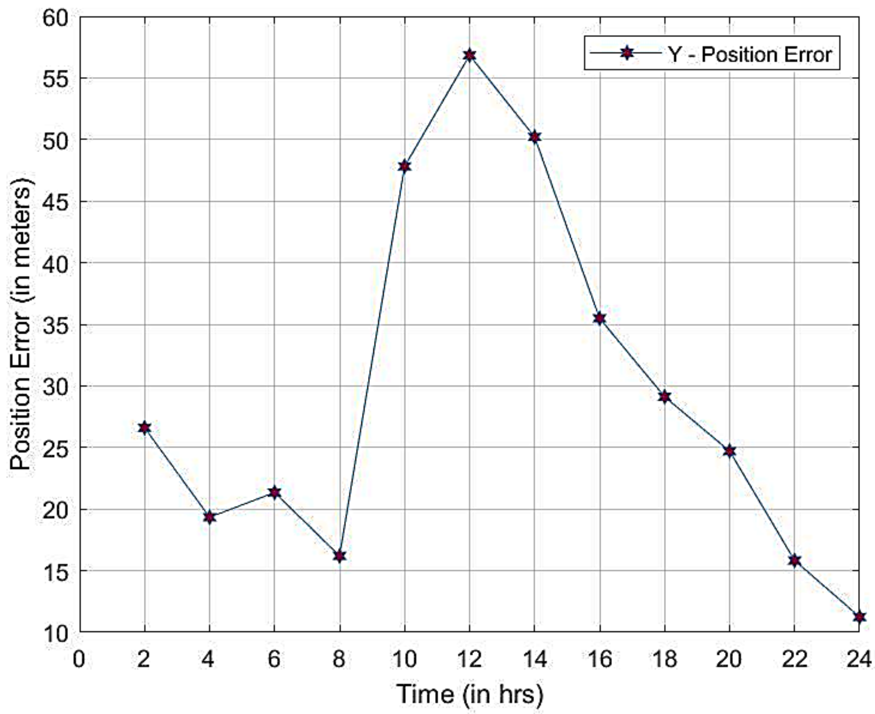

Figure 5:11 Y - Position Error due to Genetic Algorithm

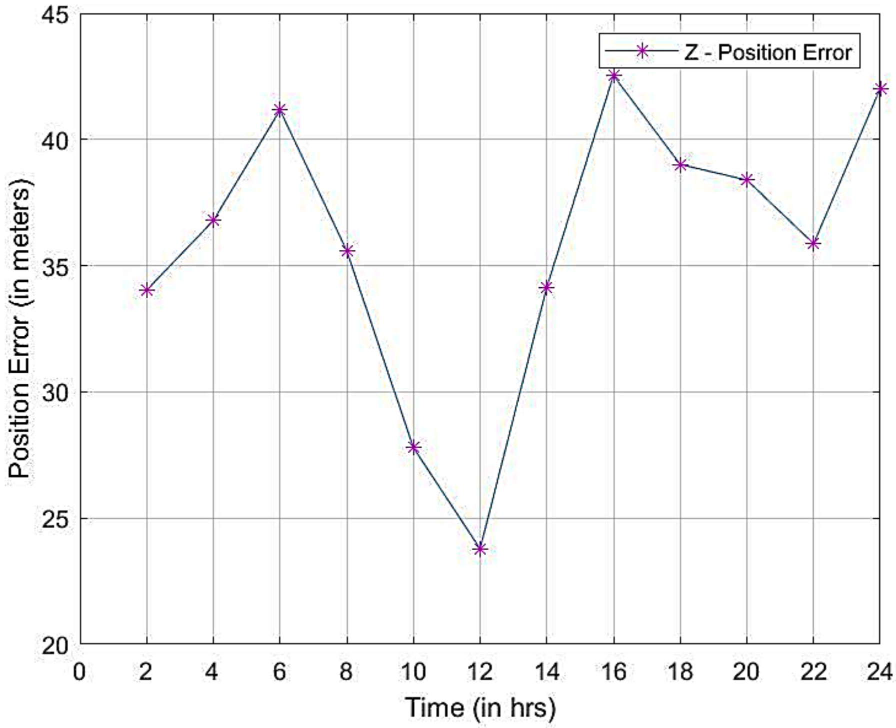

Figure 5:12 Z - Position Error due to Genetic Algorithm

In TABLE 4, the mean position errors due to Genetic Algorithm are presented. The mean position error due to Genetic Algorithm is X = *35.92* m, Y = *29.55* m, and Z = *10.03* m.

5.5 CMA-ES ALGORITHM

Table 5 Estimated Position Errors by using CMA-ES Optimization

Time (hrs)	**X - Position Error (mts)**	**Y - Position Error (mts)**	**Z - Position Error (mts)**	**Receiver Clock Bias (in ns)**
02:00	32.02870	23.59867	5.90614	2.3007
04:00	34.78520	15.33733	2.17722	9.4607
06:00	38.16120	15.34750	1.01515	1.5206
08:00	32.59528	13.19838	9.14042	1.4506
10:00	25.78924	43.81619	11.47814	8.0707
12:00	20.78154	54.83222	9.68455	8.6107
14:00	31.15035	47.20541	10.02171	3.6207
16:00	40.52991	30.48233	2.04318	1.3606
18:00	37.01195	26.11703	6.97333	5.3607
20:00	35.39038	20.70472	5.71369	1.5904
22:00	32.87202	11.82729	7.51406	7.9405
24:00	39.02838	7.24177	7.72202	7.9405
Mean	*33.34367*	*25.80907*	*6.61580*	*4.93561*

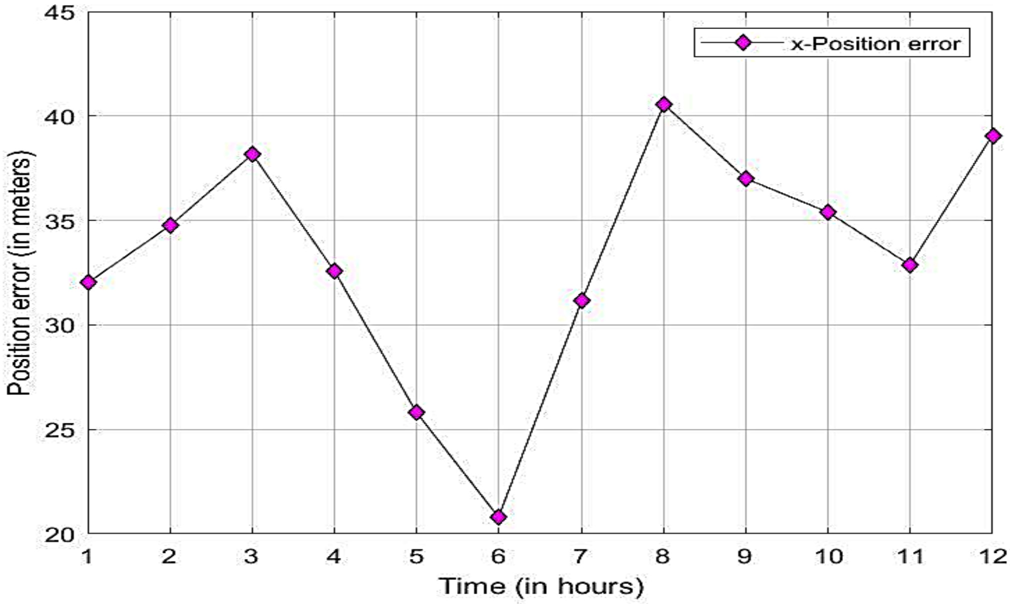

Figure 5:13 X - Position Error due to CMA-ES Algorithm

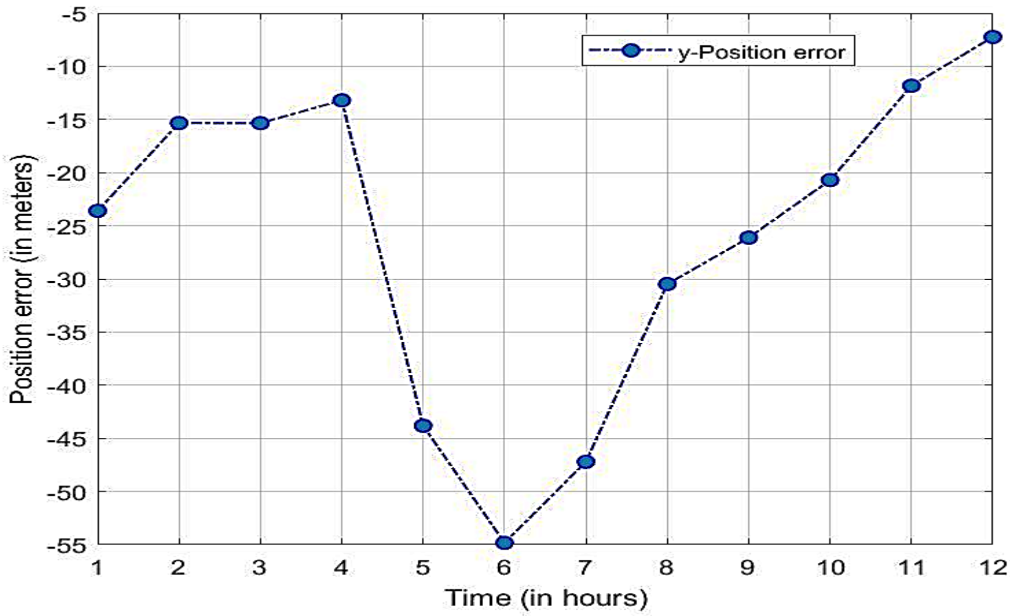

Figure 5:14 Y - Position Error due to CMA-ES Algorithm

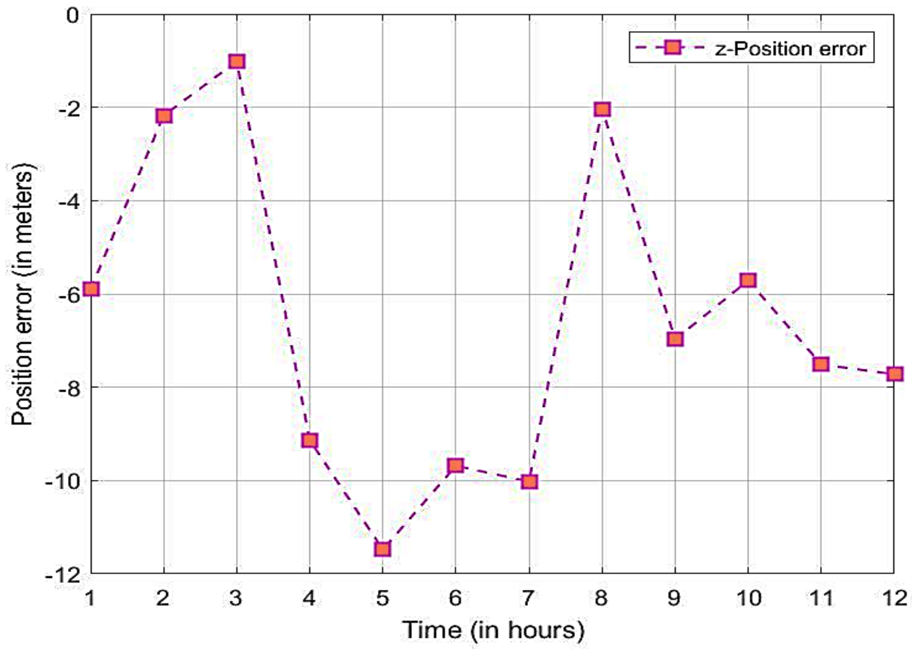

Figure 5:15 Z - Position Error due to CMA-ES Algorithm

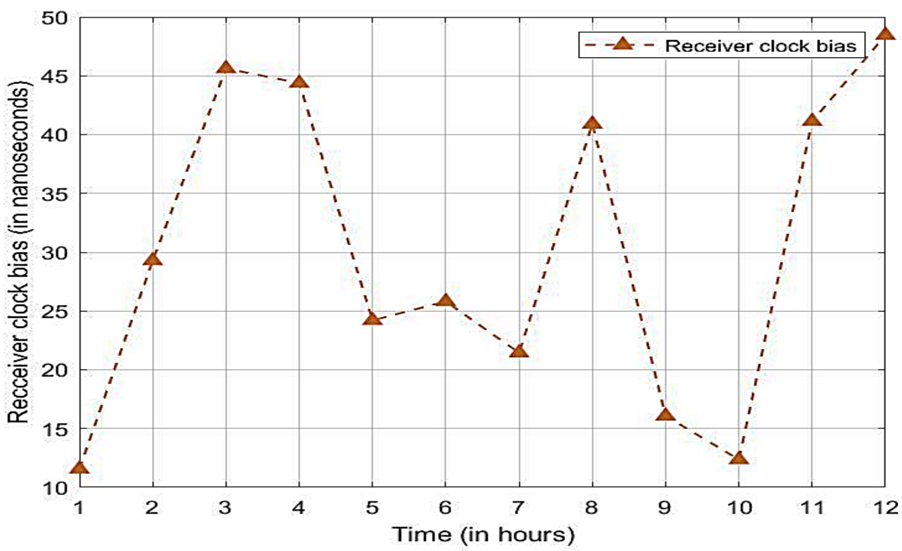

Figure 5:16 Clock Bias of Receiver due to CMA-ES Algorithm

In TABLE 5, the mean position errors due to CMA - ES Algorithm are presented. The mean position error due to Genetic Algorithm is X = *33.34* m, Y = *25.80* m, and Z = *6.61* m. And the receiver clock bias is *4.93561 ns.*

Table 6 Position Error Comparison

Algorithm	Mean Error		
	X – Position Error (mts)	Y - Position Error (mts)	Z - Position Error (mts)
Kalman Filter	*63.30062*	*142.88359*	*52.34574*
Firefly	*49.99668*	*131.99789*	*46.29227*
Genetic	*35.92701*	*29.55907*	*10.03247*
CMA - ES	*33.34367*	*25.80907*	*6.61580*

In TABLE 6, comparison of the mean position errors caused by Kalman Filter, Firefly, Genetic and CMA - ES algorithms are shown. The mean position error due to Kalman Filter algorithm is X = *63.30* m, Y = *142.88* m, and Z = *52.34* m , Firefly Algorithm is X = *49.99* m, Y = *131.99* m, and Z = *46.29,* Genetic Algorithm is X = *35.92* m, Y = *29.55* m, and Z = *10.03* m and in CMA – ES is X = *33.34* m, Y = *25.80* m, and Z = *6.61* m in x-y-, and z-directions respectively.

When the results are compared, it is clear that the mean position estimated by the CMA - ES algorithm provides more precise estimation of the GPS receiver's 3D position than the Kalman Filter, Firefly, and Genetic algorithms.

CONCLUSION

The CMA-ES algorithm is implemented on GPS data acquired in real-time from low-latitude areas of the Indian subcontinent. The estimated mean position errors by using CMA-ES are *33.34 in x - direction, 25.80 in y - direction, and 6.61 m in z - direction*, and estimated receiver clock bias is *4.93 ns*. The position reliability is further enhanced by taking into account every single word that was ignored as a correctable fault. As a result, CMA-ES is a suitable and better nature-inspired algorithm for estimating the location of any Low-latitude areas of the Indian subcontinent are home to a GPS receiver.

FUTURE SCOPE

In India's coastal regions, the application of CMA-ES for GPS receiver location estimate has a lot of potential in the future. Due to the existence of various challenges, such as structures, mountains, and dense vegetation, coastal locations are frequently difficult for GPS signal reception. CMA-ES can be used to optimize the position of GPS receivers to minimize the impact of these obstructions and improve the accuracy of GPS positioning. Furthermore, CMA-ES can be used to optimize the parameters of GPS positioning algorithms to improve their accuracy in the coastal region of India. The optimization can be performed by minimizing the difference between the estimated and actual positions obtained through ground truth measurements. It is important to note the potential of CMA-ES for estimating GPS receiver position in coastal areas of India. By adjusting the locations of GPS receivers and reference stations as well as the GPS positioning algorithms' parameters, the optimization algorithm can be used to increase GPS positioning accuracy.

REFERENCES

[1]. Covariance-tuned EKF Resampling Based Particle Filter, N Ashok Kumar, G Sasibhushana Rao, S Sudha Rani, Journal of Applied Science and Engineering, Volume – 25, Issue 4, Page 713-720.

[2]. GA Tuned Kalman Filter for Precise Positioning, Nalineekumari Arasavali, G Sasibhushana Rao, N Ashok Kumar, Microelectronics, Electromagnetics and Telecommunications, SPRINGER, 2021, Pages 285-292.

[3]. Development of Advanced Extended Kalman Filter for Precise Estimation of GPS Receiver Position, N Ashok Kumar, G Sasibhushana Rao, Nalineekumari Arasavali, IEEE, 2019.

[4]. Satellite Horizon Effects on Temporal GPS Receiver Position Accuracy over Coastal Area of South India, G Sasibhushana Rao, Lavanya Bagadi, N Ashok Kumar, SPRINGER, 2018/11 – Volume 817, Pages 1-9.

[5]. Firefly, Teaching Learning Based Optimization and Kalman Filter Methods for GPS Receiver Position Estimation, Ashok Kumar N B Lavanya, G Sasibhusana Rao, ELSEVIER, 2018/11, Volume – 143, Pages 892 – 898.

[6]. Novel BAT Algorithm for Position Estimation of A GPS Receiver Located in Coastal Region of Southern India, Ashok Kumar N, G.Sasibhushana Rao, ELSEVIER, 2018/11, Volume – 143, Pages 860-867.

[7]. P. J. G. Teunissen and O. Montenbruck (Eds.), Springer Handbook of Global Navigation Satellite Systems, 2017.

[8]. Xin-She Yang, "Nature-Inspired Optimization Algorithms", Elsevier, 1st edition, 2014; 141-155.

[9]. The CMA Evolution Strategy: A Tutorial, Nikolaus Hansen, June 28, 2011. *arXiv:1604.00772.*

[10]. Arnold, Dirk V., and Nikolaus Hansen. "A (1+ 1)-CMA-ES for constrained optimization." Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, 2010.

[11]. Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Osama Salah Eldin, CAIRO University, 06 March, 2016.

PUBLISHED PAPER STATUS