

Article

A Linear Regression-Based Methodology to Improve the Stability of a Low-Cost GPS Receiver Using the Precision Timing Signals from an Atomic Clock

Shilpa Manandhar ^{1,*}, Sneha Saravanan ² , Yu Song Meng ¹  and Yung Chuen Tan ¹ 

¹ National Metrology Centre (NMC), Agency for Science, Technology and Research (A*STAR), Singapore 637145, Singapore; meng_yusong@nmc.a-star.edu.sg (Y.S.M.); tan_yung_chuen@nmc.a-star.edu.sg (Y.C.T.)

² Department of Electronics and Instrumentation Engineering, Kumaraguru College of Technology, Coimbatore 641049, India; sneha.20ei@kct.ac.in

* Correspondence: shilpa_manandhar@nmc.a-star.edu.sg

Abstract: The global positioning system (GPS) is widely known for its applications in navigation, timing, and positioning. However, its accuracy can be greatly impacted by the performance of its receiver clocks, especially for a low-cost receiver equipped with lower-grade clocks like crystal oscillators. The objective of this study is to develop a model to improve the stability of a low-cost receiver. To achieve this, a machine-learning-based linear regression algorithm is proposed to predict the differences of the low-cost GPS receiver compared to the precision timing source. Experiments were conducted using low-cost receivers like Ublox and expensive receivers like Septentrio. The model was implemented and the clocks of low-cost receivers were steered. The outcomes demonstrate a notable enhancement in the stability of low-cost receivers after the corrections were applied. This improvement underscores the efficacy of the proposed model in enhancing the performance of low-cost GPS receivers. Consequently, these low-cost receivers can be cost-effectively utilized for various purposes, particularly in applications requiring the deployment of numerous GPS receivers to achieve extensive spatial coverage.



Citation: Manandhar, S.; Saravanan, S.; Meng, Y.S.; Tan, Y.C. A Linear Regression-Based Methodology to Improve the Stability of a Low-Cost GPS Receiver Using the Precision Timing Signals from an Atomic Clock. *Electronics* **2024**, *13*, 3321. <https://doi.org/10.3390/electronics13163321>

Academic Editor: Ping-Feng Pai

Received: 30 July 2024

Revised: 20 August 2024

Accepted: 20 August 2024

Published: 21 August 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: GPS; time and frequency; stability of clocks; clock bias errors; precision timing applications; atomic clocks; time transfer

1. Introduction

The global positioning system (GPS) stands out as one of the most extensively utilized technologies, mainly known for its applications in navigation. With the advancement in science, GPS has been shown to achieve an accuracy of up to several meters in an open sky condition; however, the accuracy deteriorates for an urban canyon setting [1]. Various factors influence the accuracy of a GPS system, with the performance of the receiver clock being a key determinant. Receivers employ diverse clocks, ranging from high-end chip-scale atomic clocks to lower-grade options, such as crystal oscillators.

In Figure 1, we observe the performance of a specific low-cost GPS receiver (Ublox), which uses a lower-grade clock. The data are collected over a day at a fixed location. They show that the maximum 2D error can go up to 40 m. Vertical positioning for these receivers is even more challenging. The vertical accuracy of a GPS receiver is directly linked to the stability of the receiver clock. The higher the receiver clock bias error, the higher the error in vertical positioning [2,3]. An error as big as that shown by Figure 1 is totally unacceptable, especially for autonomous vehicles. Real-time kinematic (RTK) signals or differential GPS (DGPS) are widely used techniques to improve the positioning accuracy of a GPS receiver [4,5]. However, these methods only improve the positioning accuracy but do not improve the clock stability by correcting the receiver clock errors. Our

proposed method here focuses on improving the stability of low-cost receiver clocks. Once the stability is improved, the positioning accuracy will be enhanced. RTK signals and DGPS could be applied further to enhance the precision of results.

The proposed methodology is to steer the GPS receiver periodically, hence improving its stability and accuracy. To achieve this, there are two main tasks: (a) generation of a precision timing source and (b) transfer of the steering information.

(a) Precision Timing Source: Atomic clocks have long been used as a source of precision time [6]. The accuracy of such timing is verified by comparing to the Universal Coordinated Time (UTC) generated and maintained by the Bureau of Weights and Measurement (BIPM) [7]. According to BIPM [8], cesium clocks are the current standards for time and frequency. Precision timing is generated from clocks that are traceable to the cesium standards. In our Time Lab at the National Metrology Centre (NMC) in Singapore, we maintain cesium clocks and hydrogen masers. The Singapore time scale is established by deriving frequency outputs from these clocks, which are meticulously preserved under optimal temperature and humidity conditions. The process of generation of precision timing based on a number of atomic clocks is almost similar for different laboratories [9–11]. However, they differ in the way they choose to combine the readings from these atomic clocks. It is important to analyze the precision time generation process. A detailed discussion on the time-scale generation process is presented in Section 2.

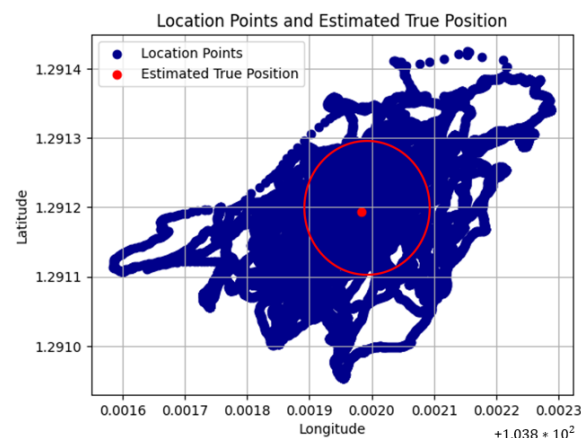


Figure 1. Scatter plot of the position recorded by Ublox for a randomly chosen day in a static condition for single point positioning (SPP). The latitude and longitude are in meters. The red circle represents a radius of 10 m error in 2D positioning.

(b) Time Transfer: Once the precision signals are generated from the lab, they can be used to correct the low-cost oscillators. The GPS time transfer technique can be applied for this purpose [12–14]. The GPS time transfer technique has previously been applied either in common view mode or all-in-view mode for the purpose of time comparison between any two laboratories [15,16]. This method has not been implemented yet for steering a low-cost GPS receiver. The paper will propose a machine-learning-based algorithm to explore the possibility of using this method to improve the stability of a low-cost GPS receiver. The details will be discussed in Section 3.2.

Contributions and Impacts

The major contribution of the paper is the model to steer the receiver clock of a low-cost receiver with the precision time information from labs. By doing so, it enhances the stability of the low-cost receiver and thus the positioning accuracy will be improved. As the proposed model is based on satellites and no physical connections are required, the advantage of the model is not only limited to improving the stability of a GPS receiver but any device that requires timing calibration can be benefited from the model. The proposed model has the capability of remotely transferring the time and frequency information.

In today's world of digitization and digitalization, such remote calibration capabilities are vital.

The proposed approach specifically targets low-cost receivers equipped with lower-grade clocks. By steering these clocks based on the devised model, the performance of such receivers can experience substantial improvement. This in turn will improve vertical positioning and the accuracy of overall positioning in general. The implications of this correction extends to applications such as autonomous vehicles, where precise positioning is imperative. Moreover, the correction of the receiver's clock performance can have a direct impact on the estimation of total GNSS signal delay. As the clock bias errors are improved, the signal delay estimation is expected to have less error. This is beneficial in climate-related studies like rainfall prediction, which relies on accurate estimation of atmospheric delay [17]. This enhancement is also particularly advantageous for studies requiring numerous GPS stations. For instance, GPS-based rainfall prediction, characterized by lower spatial resolution, necessitates the installation of multiple GPS receivers at diverse locations to enhance coverage. The current limitations posed by the high cost of receivers hinder this deployment strategy. However, the proposed methodology offers a solution by enhancing the performance of low-cost receivers, facilitating their cost-effective deployment across numerous locations.

2. Generation of Precision Timing Information

The correction of the low-cost oscillators like Ublox is based on the precision timing from the lab. This section will give an overview of the procedure involved in the generation of the precision timing in Singapore.

2.1. System Setup

There are two time and frequency laboratories under NMC in Singapore. They are located at 8 Clean Tech Loop (CT3) and Fusionopolis 2 (FP2). The lab at CT3 is the main lab traceable to *UTC* and the lab at FP2 is the backup lab traceable to *UTC(SG)*. The results discussed in this paper are with respect to the backup lab at FP2, where most of the R&D works are carried out. The hardware that we have at FP2 are shown in Figure 2. Here at the FP2 facility, we have three atomic clocks, which include two cesium clocks, and one hydrogen maser. We have one multi-channel measurement system (MMS), and one steering device (AOG-110, Microchip, Chandler, AZ, USA). We have higher-end GNSS equipment from Septentrio. This GNSS equipment has the option to be steered by an external clock as well.

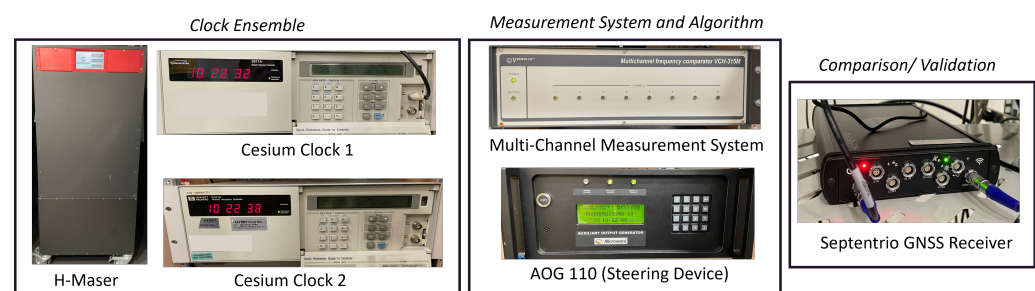


Figure 2. Hardware at Time and Frequency Lab of Singapore [located at Fusionopolis 2 (FP2)].

National Metrology Institutes (NMIs), like NMC, typically operate multiple atomic clocks, forming an ensemble [18] that contributes to the derivation of the $UTC(k)$ for a specific NMI, denoted as k . Ensemble timing capitalizes on the advantages offered by each clock within the ensemble. *AT1* is a very famous method to generate the ensemble timing that has been adopted by many NMIs [19]. ALGOS and Kalman filter-based methods are a few other techniques to generate the ensemble timings [20,21]. The next section will describe the time-scale algorithm that has been implemented for this paper.

2.2. Time-Scale Algorithm

Figure 3 shows the schematic of the system setup. It consists of all the hardware discussed in the previous section. As shown by the schematic, the MMS captures the time differences between a reference clock and the input clocks, with the H-maser serving as the reference in our case. Therefore MMS records two sets of differences, $x_r - x_1$ and $x_r - x_2$, differences between H-maser and cesium clock 1 and H-maser and cesium clock 2, respectively. These recorded differences are then used by a time-scale algorithm to calculate the ensemble time, x_{re} . For this paper, the AT1 algorithm is implemented.

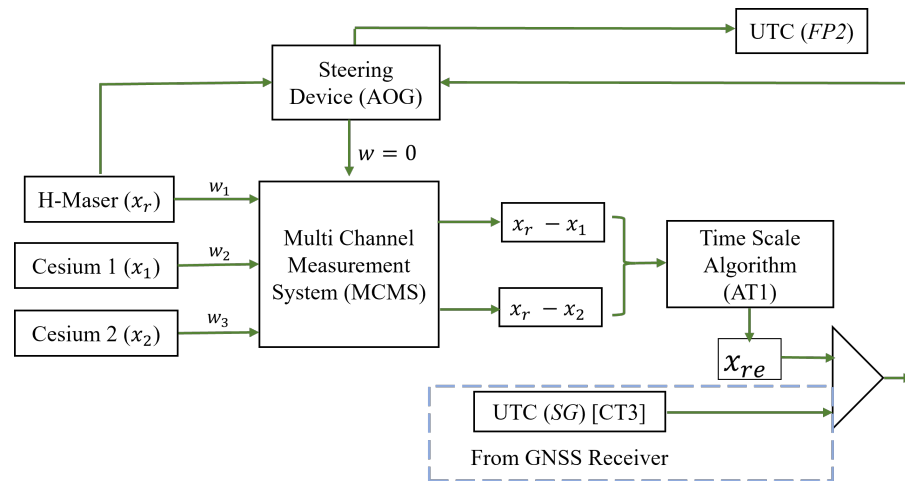


Figure 3. Flowchart of the system setup at the Time and Frequency Lab at FP2.

All time-scale algorithms share a common foundation, expressed by the general equation depicted in Equation (1).

$$x_{je}\hat{\tau}(k) = x_{je}(k-1) + y_{je}(k-1)\tau + 0.5d_{je}(k-1)\tau^2 \quad (1)$$

The equation shows the fundamental relationship used to estimate the behavior of a clock j with respect to an ensemble e , at time stamp k given the condition of clock at $k-1$. Here, x_{je} is the phase difference, y_{je} is the frequency difference, and d_{je} is the frequency aging of clock j with respect to the ensemble e . The AT1 algorithm is a recursive algorithm that can be used in the estimation of clock behavior. The procedural steps involved in the realization of the AT1 algorithm have been described in our earlier paper [22].

Determination of Clock Weights

In the AT1 algorithm, a crucial step involves determining the weights (w) assigned to each clock. These weights are generally inversely proportional to their errors i.e. clocks with higher errors receive lower weights. Different algorithms use different approaches for determining weights. Some rely on error variances [9]. Typically, an administrative weight limit is also put on the weights (like defining a maximum value of the weight that a clock in the ensemble can get) such that the ensemble is not biased to only one type of clock in the ensemble [20].

For this paper, the weights are determined based on the estimation error of each clock. Let A , B , and C represent the estimation errors and W_{HM} , W_{C1} , and W_{C2} represent the weights for the H-maser, cesium 1, and cesium 2, respectively. The weights and the errors are inversely proportional to each other: $W_{HM} \propto 1/A$, $W_{C1} \propto 1/B$, $W_{C2} \propto 1/C$. The sum of the weights equals 1, as shown by Equation (2). An administrative limit of 0.5 is imposed to ensure that the weight of any clock does not exceed 0.5. As an example, Equations (2) and (3) show how the weights for any given clock, e.g., hydrogen maser, are calculated.

$$W = K(W_{HM} + W_{C1} + W_{C2}) = 1 \quad (2)$$

$$K = ABC / (AB + BC + CA) \quad (3)$$

$$W_{HM} = K / A = BC / (AB + BC + CA) \quad (4)$$

The weights for the rest of the clocks can be estimated in a similar manner. The implementation of the algorithm results in the estimates of all clocks in the system with respect to the ensemble time i.e. x_{1e} , x_{2e} , and x_{re} for cesium clock 1, cesium clock 2, and H-maser, respectively.

2.3. Steering Algorithm

As illustrated by the system diagram in Figure 3, the steering process is facilitated by the AOG. This steering device takes in the signal from the H-maser and adjusts it based on the output provided by the time-scale algorithm (x_{re}). For this paper, frequency steering has been implemented, and the steering value s for a given timestamp k is calculated as shown in Equation (5). The process is automatic and is controlled by a script written in Python. The script generates the steering values every two-hours.

$$s(k) = (x_{re}(k) - x_{re}(k - t)) / t \quad (5)$$

Enhancements to the steering process can be achieved by incorporating information on the difference between AOG and *UTC(SG)*, as illustrated in the system diagram in Figure 3. This adjustment aims to refine the results obtained from our lab and FP2, and this optimization will be a focus of our upcoming work. In the meantime, the outcomes of AOG after the steering process, utilizing Equation (5), define the physical signal at FP2, denoted as the backup time for Singapore (*UTC(FP2)*). Subsequently, when this signal was compared to *UTC(SG)* for validation purposes, almost 82.5% of the differences were within ± 10 ns [23].

3. Data Processing & Methodology

This section describes the different data that have been used for devising the proposed model. Since the data come from two different sources, this section will introduce them separately. An overall methodology that has been implemented to achieve the objective will also be described.

3.1. Receiver Clock Offset w.r.t GPS Time

The time difference between a receiver clock and the satellites for a particular GPS receiver is documented in the Common GPS GLONASS Time Transfer Standard (CGGTTS) files. The Consultative Committee for Time and Frequency (CCTF) has published a set of guidelines based on which the CGGTTS files are created. These guidelines have been compiled with the aim of achieving a system that can transfer time with an accuracy of 1 ns or better. They can be applied to all available global navigation satellite system constellations, such as GPS, GLONASS, Beidou, QZSS, and Galileo. For this paper, we process these files to get information on the receiver clock performance w.r.t the satellite time.

The CGGTTS files record different important data types [24,25]. One of them is Refsys. This is the time difference between the receiver clock and the global navigation satellite system (GNSS) time. If the GPS constellation is considered then Refsys is the difference between the receiver clock and GPS time (GPST). The data available in CGGTTS files are generally 16 min apart and the values are reported in units of 0.1 ns.

We have used data from two different receivers with different grades of clocks; namely Ublox (Thalwil, Switzerland) and Septentrio (Leuven, Belgium) receivers. The Ublox that has been used for this paper is the Ublox from the M8T series [26]. It is a low-cost receiver steered by TCXO (temperature compensated crystal oscillator). However, it is very small in size and thus is portable and can be easily mounted in many locations. The Septentrio receiver, on the other hand, is expensive compared to Ublox. The Septentrio receiver model

PolaRx5TR has been used [27]. It is steered by Rubidium clocks and has the option to be steered by external clocks too.

The same Ublox was deployed at two different sites for data collection. The two sites were at Nanyang Technological University (NTU) and at Stirling Road. Figure 4 shows these two locations. The locations were chosen as per the convenience of the students to be able to record the data without intervention for as long as possible. The Septentrio receiver is located at the backup Time and Frequency Laboratory of the National Metrology Centre (NMC), Singapore (also shown in the Figure 4).

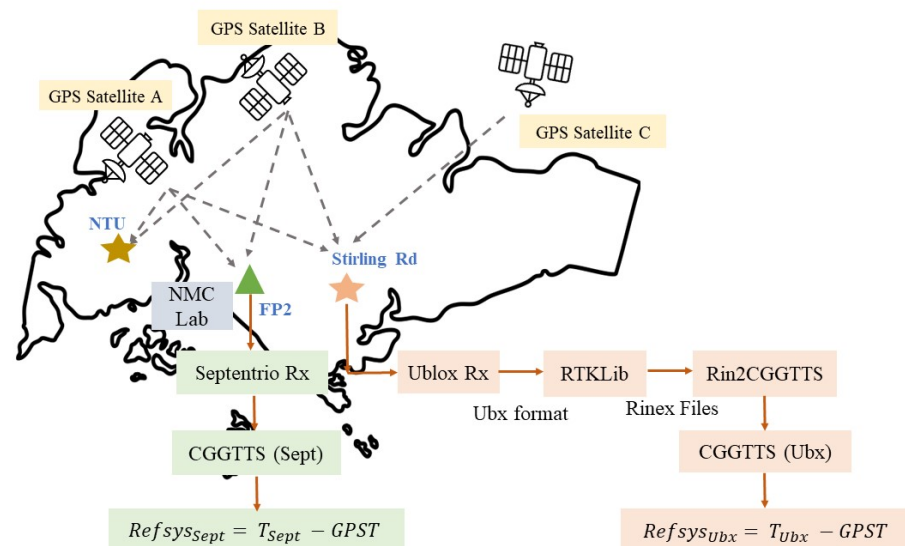


Figure 4. Map of Singapore showing the location of the receivers: Septentrio receiver at Time Lab in FP2 (in green triangle), Ublox receiver at NTU (in gold star), and Ublox receiver at Stirling (in orange star). The diagram also shows the flow of post-processing steps for a Ublox receiver.

Table 1 shows the common days that have been used for processing the results. The data from the NMC lab at Fusionopolis 2 for the corresponding date and time at both the Ublox locations (at NTU and Stirling Road) are used for the analysis.

Table 1. Data used for processing.

Location	Calendar Year	MJD
NTU	3 February 2024 to 5 April 2024	60343–60405
Stirling Rd	5 July 2023 to 1 September 2023	60130–60188

3.1.1. Ublox Data Processing

The Ublox records data in ubx format. It stores information that is useful in deriving the Refsys values. The flowchart in Figure 4 shows the general steps involved in getting the Refsys values from the raw ubx data. The ubx data are first converted into the receiver independent exchange (Rinex) observation and navigation files by using the rtklib software (version 2.4.3), which is freely available for use [28]. The Rinex observation files are then converted to the CGGTTS format by using the rin2cggts program. The program is distributed by BIPM for non-commercial purposes [24].

3.1.2. Septentrio Data Processing

The Septentrio receiver records the data in standard CGGTTS format. The data are recorded for different constellations with an identifier. Since only GPS constellation is being considered for this paper, only the CGGTTS file with GPS identifier will be processed. For Septentrio receivers, the receiver can be steered to an external reference clock. For

the experiments conducted for this paper, the Septentrio receiver will have an external reference, which is the precision timing generated at FP2.

An experiment was run to check on the data from the Septentrio receiver to study its performance in the presence of the external reference. For this experiment, a free-running cesium atomic clock was chosen as the external reference. Figure 5 shows the data with and without the external reference. The red dots are when the Septentrio receiver is running by its internal reference and the blue dots are when the external reference (from the FP2 Lab) is connected to the receiver. Almost the same number of days were considered for the comparison; however, there were some missing data for the case when the receiver is run by its internal clock. This is because there was a period when the instruments were shifted due to the relocation of the labs. The case was soon identified, and the devices were again working fine with no missing data after.

Figure 5 clearly shows that the distribution of the data with time is very much affected by the presence of an external reference. As expected, with the external reference, which is very precise, the data have a lesser range and less variation when compared to the case with the internal clock only. Also, there is a clear separation between the two when the amplitudes of the data are compared. The Refsys values in the presence of the external reference are more than 20 times smaller. For applications where the receiver clock bias errors are critical, like positioning, such differences can cause a huge impact.

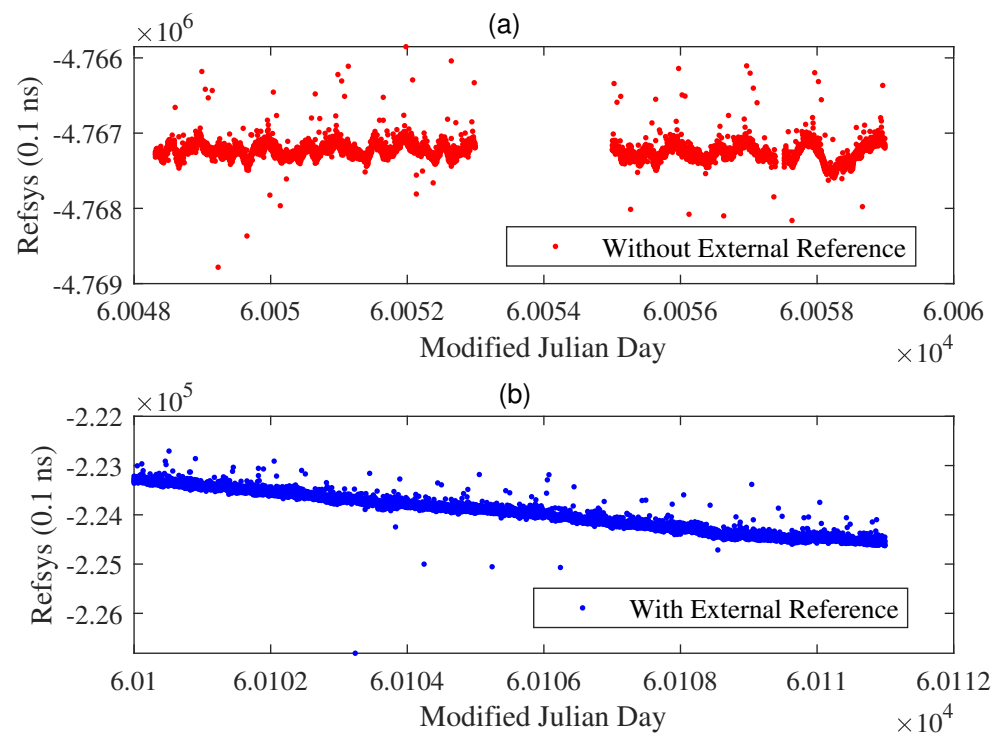


Figure 5. Results (a) before and (b) after synchronization to the external reference. The legend in each figure represents the condition. The graph with red dots is for “before” and the graph with blue dots is for “after” the synchronization.

3.2. Methodology

This section describes the procedure that is used in deriving the receiver clock differences between the two receivers Septentrio and Ublox. Both the receivers record the differences with respect to the GPST. $Refsys_{Sept}$ denotes the differences between the data from Septentrio and GPST, and similarly, $Refsys_{Ubx}$ denotes the differences between the data from Ublox and GPST. Here, we see that the GPST is the common time reference between the two receivers. If this common GPST is subtracted, we can compute the differ-

ences between the Septentrio and the Ublox receiver. This can be done by following either of the two approaches: *common view method* and *all-in-view method* [15,16].

Figure 4 shows that there are two satellites (GPS satellites A and B) that are viewed from both the receivers at that instant, whereas only GPS satellite C is viewed by the Ublox receiver. As the name suggests, if only common satellites (in this case A and B) are used, then it is termed as common view. If the data from all satellites are averaged first and then the difference is calculated, it is termed the all-in-view method. For this paper, the common view method will be implemented. However, there are not many differences between the two methods for the context of this paper as the distance between the two receiver locations is very short. The same satellites are visible almost all the time by both the stations.

4. Results and Analysis

This section will discuss the results processed for the Ublox receivers [located at two locations and the Time Lab at FP2]. A difference-correction algorithm will be introduced to correct the errors of a Ublox receiver. For all the experiments conducted for this section, the Septentrio receiver is connected to the external time reference provided by the FP2 lab (as described in Section 2).

4.1. Performance Comparison between Septentrio and Ublox Receivers

It is expected for the Septentrio receiver to perform better compared to the Ublox receiver. It is interesting to observe how the data for the Ublox fluctuates compared to the Septentrio. This gives us an opportunity to study and correct such fluctuations with reference to the stable output from Septentrio.

Figure 6 shows the respective Refsys values for the receivers at NTU, at Stirling Road, and at FP2. The x-axis of all three plots shows number of days. Each boxplot of one day shows the distribution of Refsys values. The Refsys values here are for all the satellites visible for that particular day. We can note two important variations here. Firstly, the daily variations: as Refsys represents the difference between the receiver clock and the GPST, it must be the same irrespective of the time of the day and the visible satellites. However, we observe that there are variations for each day for both receivers. This might be because of the satellite clock errors. Different satellites have their own clock errors. Thus, the Refsys calculated from Sat 1, for example, might have a difference compared to when calculated from Sat 10. The boxplot shows such discrepancies.

Secondly, a greater variation can be seen in the data when compared across the different days. It can be observed that the data vary quite distinctly with days for these two types of receivers. This is due to the fact that the receiver clock performance changes with time. The clocks have aging that affects their stability and performance. The Refsys data for Septentrio show a linear trend in the performance. Similar behavior is shown by the performance of the atomic clocks. Since the Septentrio receiver is disciplined by precision timing traceable to the cesium standards, it follows a similar pattern. No such linear trend is observed for the case of the Ublox receiver. The Refsys data for both the locations (NTU and Stirling Road) show that the performance is degrading with time but there is no fixed pattern. This was also observed for the case when Septentrio was freely running (without an external time source) in Figure 5. Moreover, the changing weather conditions for different days can add on to such variations.

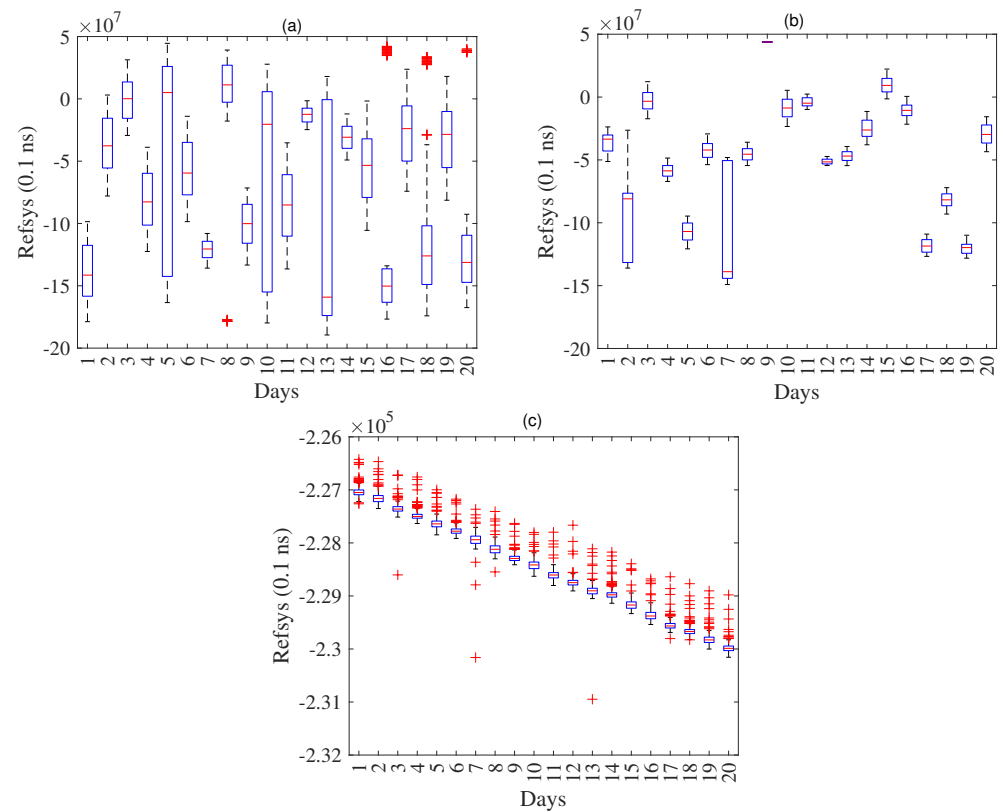


Figure 6. Boxplot to compare the data from Septentrio and Ublox receivers for (a) NTU, (b) Stirling Rd, and (c) Septentrio receiver. The x-axes for all plots are the number of days, as shown by Table 1 and the y-axes for all the plots are the respective Refsys values.

When the absolute amplitudes of these two types of receivers are compared, it shows that the performance of the Ublox receiver is very poor. The values for both the locations (NTU and Stirling Road) are poorer compared to the case when the Septentrio receiver was freely running. These higher errors can lead to greater uncertainty in the results produced by the Ublox receivers. This is one of the reasons that there is a greater variation in vertical height information when calculated using Ublox receivers [2]. This study shows that it is very important to build a method to correct such clock bias errors. The next section will propose the solution.

4.2. Data-Driven Correction Model

As established in the earlier section, the performance of a low-cost GPS receiver like Ublox is highly unstable and has higher clock-bias errors which can be translated into and not limited to longitudinal and latitudinal errors, vertical height errors, and errors in tropospheric delays. This section explores a way to correct the performance of such low-cost GPS receivers.

The GPS receivers have the advantage that they are connected to different receivers remotely via satellites. At a given instant, time differences between different GPS receivers can be compared based on the satellites in view. As explained in Section 3.2, this can be achieved by either the common view method or the all-in-view method. For this paper, the common view method is implemented.

At any given instant, all the Refsys values from two receivers, Septentrio ($Refsys_{Sept}$) and Ublox ($Refsys_{Ubx}$), are recorded. A machine-learning model is implemented that takes the differences between ($Refsys_{Sept}$) and ($Refsys_{Ubx}$) and predicts the correction factor for the Ublox receivers. The same procedure is repeated for the data from two Ublox locations (at NTU and at Stirling Road).

There are different machine-learning models that can be explored for difference prediction. We compared the few well-known ones. Support vector machine (SVM) is a powerful model that has been successfully applied in various real-life applications [29]. However, SVMs have some limitations that may make them less suitable for certain scenarios. One of the main issues with SVMs is their high time and memory training complexities, which depend on the size of the training set [29]. This can be problematic when dealing with large datasets, as it can lead to longer training times. While a k-nearest neighbors algorithm (KNN) is simple and can capture local patterns, it is sensitive to noise and requires careful tuning of the k parameter. In the case of aligning Refsys, where the relationship might not be strictly local, it may not perform optimally [30]. Random forest is also not suitable for this situation because it introduces unnecessary complexity. Random forest could lead to overfitting and is computationally more intensive compared to the task in hand [31]. Therefore, we explore a linear regression model, which can work best for our data.

Linear Regression Model

Linear regression is used in this context because it can provide a quick and relatively efficient solution to correct $Refsys_{ubx}$ with the $Refsys_{sept}$ values. The model works well when there is a linear relationship between the variables, and it converges quickly. Given that the data have a lot of outlier values, a more robust model like ridge regression or Huber regression could have been considered as well. However, in this case, linear regression seems to provide satisfactory results with a lower training time and is able to handle the linear relationship between the two datasets effectively.

Figure 7 shows the flowchart of the program that has been implemented for the paper. Python programming language has been used for this implementation. The data are pre-processed and any outliers are removed. The algorithm then calculates the differences between the $Refsys_{ubx}$ and $Refsys_{sept}$. These differences are the features for this machine-learning algorithm. In the next step, the model is trained with the difference values to predict the correction factor. A total of 80% of the data is split into the training set, and 20% is allocated for the validation (test) set. The linear regression model is then initialized and trained with the training dataset. The trained regressor will then be used to predict the differences for the test data. The predicted data will then be used to correct the test data. In this case, the Ublox receiver's timings will be corrected by adding these predicted differences.

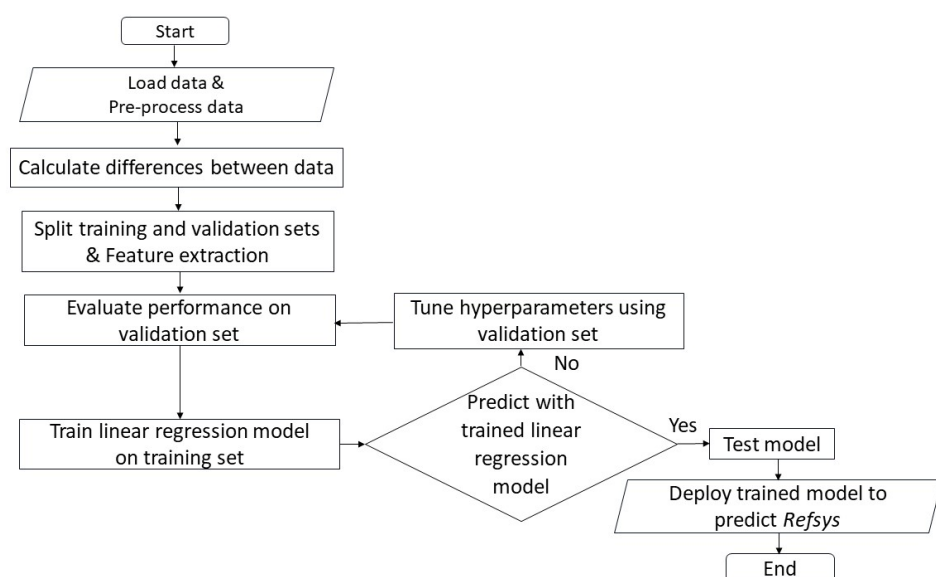


Figure 7. Flowchart of the proposed machine-learning-based correction model.

Here, the hyperparameters for the linear regression model are adjusted primarily through the selection of data splits and reproducibility settings. Specifically, the test size parameter defines the split between training and test data, while random state ensures consistent results across runs. Along with these, grid search and regularization are employed to optimize these parameters, enhancing the model's ability to correct $Refsys_{ubx}$ values and improve alignment with $Refsys_{sept}$ values.

4.3. Performance Comparison

Generally, the performance of a clock is reported in terms of its stability. Allan deviation is the most common form to express such stability. The lower the value of the Allan deviation, the higher the stability. Stable 32 is a well-known software that helps to generate such stability results [32]. This software is used to analyze the results for this paper too.

The stability graphs are plotted for the cases before the application of the correction factor and after the predicted correction factors are applied. In Figure 8, the dashed curves show the stability of the receiver before the application of the correction factor and the solid curves show the stability of the receiver after the application of the correction factor. The red and the blue colors are for the data from NTU and Stirling Road, respectively. It can be noted that the stability of the Ublox for the two locations look very similar for both the cases of before and after the application of the correction factor.

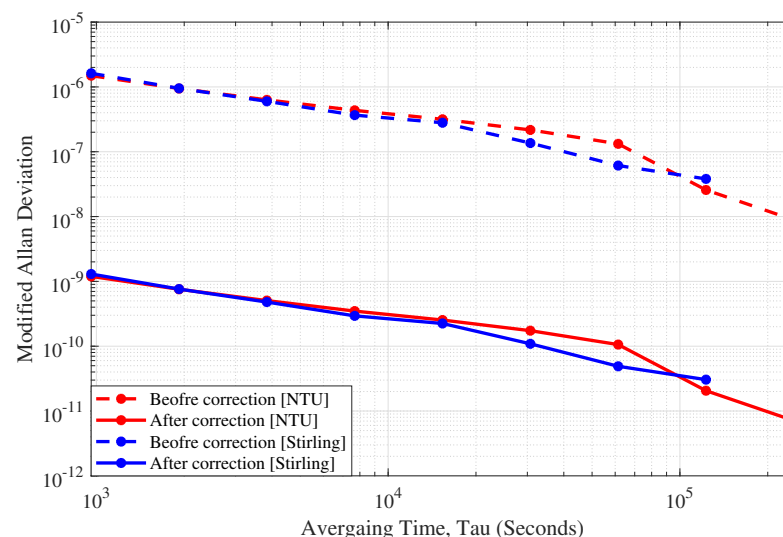


Figure 8. Modified Allan deviation for the $Refsys_{ubx}$ values. The red curve is for before the application of the correction and the blue curve is for after the correction has been applied.

There is a clear shift in the stability curves after the predicted correction factors were applied. The stability is improved for all the averaging times. This shows that the linear-regression-based model is able to improve the performance of a low-cost oscillator. For an averaging time of 1 day (x-axis reading of 86,400 s), the stability is improved by 10^3 times. This huge improvement in the stability validates that a simple data-driven algorithm-based steering is very much effective in correcting the performance of a low-cost oscillator.

5. Conclusions and Future Work

This paper identifies the significance of improving the stability of a low-cost GPS receiver. An improved low-cost receiver has a potential to provide location information as precisely and accurately as high-cost receivers. Thus, the spatial coverage could be increased by deploying a number of low-cost receivers. This leads to a better determination of location. The applications of a low-cost GPS receiver with improved stability are not only limited to positioning but the receiver could also be used for weather-related studies.

To achieve this, the paper proposed a machine-learning algorithm to predict the time differences between a low-cost GPS receiver and a higher-end receiver. The predicted differences were then used to correct the errors associated with the low-cost receiver. This then improved the stability of the receiver. Results show that for an averaging time of 1 day, the stability is improved by almost 10^3 times.

Moreover, the proposed methodology can extend its application in remote calibration of atomic clocks. Traditionally, the atomic clocks are transported to the calibration center, but with the proposed methodology, the clock can be calibrated remotely via GPS satellites. This saves both cost and time for the companies that need the calibration services.

In future work, the Ublox data, after being corrected, will be used in deriving location information and an improvement in the accuracy of the navigation will be shown. Also, the prediction of the differences will be done for different time steps in the future (say 6 h advance in time) and the effects of it will be studied.

Author Contributions: Conceptualization, S.M.; methodology, S.S. and S.M.; software, S.S.; validation, S.M., Y.S.M. and Y.C.T.; formal analysis, S.M. and Y.S.M.; investigation, S.M., S.S. and Y.C.T.; resources, Y.S.M.; writing—original draft preparation, S.M. and S.S.; writing—review and editing, S.M., Y.S.M. and Y.C.T.; visualization, S.M. and Y.S.M.; supervision, S.M. and Y.S.M.; project administration, S.M.; funding acquisition, S.M. and Y.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Agency for Science, Technology and Research (A*STAR) under Project No. 222D800033 [Career Development Fund], and the National Research Foundation, Singapore and A*STAR under its Quantum Engineering Programme (NRF2021-QEP2-01-P03).

Data Availability Statement: The data collected for this research involve the time and frequency data from the cesium clocks of NMC, which are not allowed to be shared publicly.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jiménez-Martínez, M.J.; Farjas-Abadia, M.; Quesada-Olmo, N. An Approach to Improving GNSS Positioning Accuracy Using Several GNSS Devices. *Remote Sens.* **2021**, *13*, 1149. [CrossRef]
2. Misra, P.; Pratt, M.; Bruke, B.; Ferrantio, R. Adaptive modeling of receiver clock for meter-level DGPS vertical positioning. In Proceedings of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS 1995), Palm Springs, CA, USA, 12–15 September 1995; pp. 1127–1135.
3. Manandhar, S.; Meng, Y.S. Improvement in Vertical Positioning with GPS Receiver Clock Steered by Precise Time Reference. In Proceedings of the 2021 IEEE USNC-URSI Radio Science Meeting (Joint with AP-S Symposium), Singapore, 4–10 September 2021; pp. 82–83.
4. Niu, Z.; Zhao, X.; Sun, J.; Tao, L.; Zhu, B. A Continuous Positioning Algorithm Based on RTK and VI-SLAM with Smartphones. *IEEE Access* **2020**, *8*, 185638–185650. [CrossRef]
5. Ku, D.Y.; Patel, A. Kinematic State Estimation Using Multiple DGPS/MEMS-IMU Sensors. *IEEE Sens. Lett.* **2020**, *4*, 1–4. [CrossRef]
6. Jones, R.H.; Tryon, P.V. Estimating Time from Atomic Clocks. *J. Res. Natl. Bur. Stand.* **1983**, *88*, 17. [CrossRef] [PubMed]
7. Arias, F.; Jiang, Z.; Lewandowski, W.; Petit, G. BIPM comparison of time transfer techniques. In Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition, Vancouver, BC, Canada, 29–31 August 2005.
8. BIPM Technical Services: Time Metrology. Available online: <https://www.bipm.org/en/time-metrology> (accessed on 30 June 2024).
9. López-Romero, J.M.; Díaz-Muñoz, N. Progress in the generation of the UTC(CNM) in terms of a virtual clock. *Metrologia* **2008**, *45*, S59. [CrossRef]
10. Patriza, T.; Claudine, T. Comparative Study of Time Scale Algorithms. *Metrologia* **2005**, *28*, 57.
11. Zhao, S.; Dong, S.; Qu, L.; Yuan, H.; Bai, S.; Wu, W.; Wang, X.; Yin, D. A new steering strategy for UTC(NTSC) based on hydrogen maser. In Proceedings of the IEEE International Frequency Control Symposium, IFCS, New Orleans, LA, USA, 9–12 May 2016.
12. Matsakis, D.; Defraigne, P.; Banerjee, P. Precise Time and Frequency Transfer. *Radio Sci. Bull.* **2014**, *351*, 29–44.
13. Lewandowski, W.; Azoubib, J.; Klepczynski, W.J. GPS: Primary tool for time transfer. *Proc. IEEE* **1999**, *87*, 163–172. [CrossRef]
14. Lee, S.W.; Lee, C.B.; Yang, S.H. A New Approach for Steering UTC(KRIS). *IEEE Trans. Instrum. Meas.* **2009**, *58*, 1247–1251.
15. Wang, J.L.; Huang, S.Y.; Liao, C.S. Time and frequency transfer system using GNSS receiver. *Radio Sci.* **2014**, *49*, 1171–1182. [CrossRef]
16. Weiss, M.A.; Petit, G.; Jiang, Z. A comparison of GPS common-view time transfer to all-in-view. In Proceedings of the 2005 IEEE International Frequency Control Symposium and Exposition, Vancouver, BC, Canada, 29–31 August 2005.

17. Manandhar, S.; Lee, Y.H.; Meng, Y.S.; Yuan, F.; Ong, J.T. GPS-Derived PWV for Rainfall Nowcasting in Tropical Region. *IEEE Trans. Geosci. Remote Sens.* **2018**, *8*, 4835–4844. [\[CrossRef\]](#)
18. Borregaard, J.; Sørensen, A.S. Efficient Atomic Clocks Operated with Several Atomic Ensembles. *Phys. Rev. Lett.* **2013**, *111*, 090802. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Weis, M.; Weissert, T. A time-scale algorithm for post-processing: AT1 plus frequency variance. In Proceedings of the Conference on Precision Electromagnetic Measurements, Ottawa, ON, Canada, 11–14 June 1990.
20. Levine, J. Realizing UTC(NIST) at a remote location. *Metrologia* **2008**, *45*, S23–S33. [\[CrossRef\]](#)
21. Song, H.; Dong, S.; Qu, L.; Wang, X.; Guo, D. A robust Kalman filter time scale algorithm with data anomaly. *J. Instrum.* **2021**, *16*, P06032. [\[CrossRef\]](#)
22. Manandhar, S.; Meng, Y.S. Study on the Effect of using Different Weighting Techniques in a Time Scale Algorithm to Generate an Ensemble Time. In Proceedings of the 3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC), Gran Canaria, Spain, 29 May–3 June 2022; pp. 1–4.
23. Manandhar, S.; Chai, J.Z.; Meng, Y.S.; Tan, Y.C. Performance Comparison of AT1 Algorithm for a Smaller Ensemble of Atomic Clocks. In Proceedings of the URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC), Gran Canaria, Spain, 19–24 May 2024.
24. Defraigne, P.; Petit, G. CGGTTS- Version 2E: An extended standard for GNSS Time Transfer. *Metrologia* **2015**, *52*, G1. [\[CrossRef\]](#)
25. CGGTTS Data Format. Available online: <https://www.cnmc.usff.navy.mil/Our-Commands/United-States-Naval-Observatory/Precise-Time-Department/Global-Positioning-System/GPS-Timing-Data-and-Information/> (accessed on 30 June 2024).
26. UBLOX:NEO/LEA-M8T Series. Available online: <https://www.u-blox.com/en/product/neolea-m8t-series> (accessed on 30 June 2024).
27. Septentrio PolaRx5TR. Available online: <https://www.septentrio.com/en/products/gps/gnss-reference-receivers/polarx-5tr> (accessed on 30 June 2024).
28. RTKLIB: An Open Source Program Package for GNSS Positioning- Version 2.4.3. Available online: <https://www.rtklib.com/> (accessed on 30 June 2024).
29. Nalepa, J.; Kawulok, M. Selecting training sets for support vector machines: A review. *Artif. Intell. Rev.* **2019**, *52*, 857–900. [\[CrossRef\]](#)
30. Zeniarja, J.; Ukhifahdhina, A.; Salam, A. Diagnosis of heart disease using k-nearest neighbor method based on forward selection. *J. Appl. Intell. Syst.* **2020**, *4*, 39–47. [\[CrossRef\]](#)
31. Assuncao, J.; Fernandes, P.; Lopes, L.; Normey, S. Distributed stochastic aware random forests—Efficient data mining for big data. In Proceedings of the IEEE International Congress on Big Data, Santa Clara, CA, USA, 6–9 October 2013.
32. Stable32—Version 1.62. Available online: <http://www.stable32.com/> (accessed on 30 June 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.