University of North Dakota
UND Scholarly Commons

Theses and Dissertations | Theses, Dissertations, and Senior Projects

January 2022

An Exploratory Analysis Of A Time Synchronization Protocol For UAS

Marcos Fernandez-Tous

Follow this and additional works at: https://commons.und.edu/theses

Recommended Citation

Fernandez-Tous, Marcos, "An Exploratory Analysis Of A Time Synchronization Protocol For UAS" (2022). *Theses and Dissertations*. 4257.
https://commons.und.edu/theses/4257

This Dissertation is brought to you for free and open access by the Theses, Dissertations, and Senior Projects at UND Scholarly Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UND Scholarly Commons. For more information, please contact und.commons@library.und.edu.

AN EXPLORATORY ANALYSIS
OF A TIME SYNCHRONIZATION PROTOCOL FOR UAS

by

Marcos Fernandez-Tous

Master of Science, Polytechnic University of Madrid (Spain), 2001

A Dissertation

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfilment of the requirements

for the degree of

Doctor of Philosophy in Aerospace Sciences

Grand Forks, North Dakota

May

2022

Copyright 2022 Marcos Fernandez-Tous

DocuSign Envelope ID: 00D4104C-1C24-48DE-AD47-D17FAFECBE05

Name: Marcos Fernandez-Tous

Degree: Doctor of Philosophy

This document, submitted in partial fulfillment of the requirements for the degree from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done and is hereby approved.

DocuSigned by:
Joseph Vacek
6B77155E1233416...
Dr. Joseph Vacek

DocuSigned by:
7A363CB8EE5F433...
Dr. Pablo de Leon

DocuSigned by:
Dr. Mark Askelson
831C5247AA12468...
Dr. Mark Askelson

DocuSigned by:
Paul Snyder
869C2AE7086B4AB...
Paul Snyder

DocuSigned by:
Robert Thorson
CFBDD8BF5EFB4B3...
Robert Thorson

This document is being submitted by the appointed advisory committee as having met all the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

DocuSigned by:
Chris Nelson
2E0AF088C733403...
Chris Nelson
Dean of the School of Graduate Studies

3/14/2022
Date

PERMISSION

Title An Exploratory Analysis of a Time Synchronization Protocol for UAS

Department Aviation

Degree Doctor of Philosophy

In presenting this dissertation in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my dissertation work or, in his absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this dissertation or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my dissertation.

Marcos Fernandez-Tous

May 15, 2022

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND ACRONYMS

Abbreviation / Acronym	**Definition**
3GPP	3rd Generation Partnership Project
4G	Fourth generation of cellular broadband technology
5G	Fifth generation of cellular broadband technology
ANSP	Air Navigation Service Provider
BIPM	Bureau International des Poids et Mesures (International Bureau of Weights and Measures)
BVLOS	Beyond Visual Line of Sight
CFIT	Controlled Flight Into terrain
CRB	Cramér-Rao Bound
CSMA	Carrier Sense Multiple Access
FAA	Federal Aviation Administration
FCC	Federal Communications Commission
GCS	Ground Control Station
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronic Engineers

INS	Inertial Navigation System
IPP	Integrated Pilot Program
LORAN	Long Range Navigation
LTE	Long Term Evolution
MEMS	Micro-Electrical Mechanical Systems
MLE	Maximum Likelihood Estimation
NAS	National Airspace
NLOS	Non-Line of Sight
NPUASTS	Northern Plains UAS Test Site
NTP	Network Time Protocol
NTSB	National Transport Safety Board
OLS	Ordinary Least Squares
OSI	Open System Interconnection
PTP	Precision Time Protocol
RBS	Reference Broadcast Synchronization
RMS	Root Mean Squared
ROS	Receiver Only Synchronization
RPIC	Remote Pilot In Command
SLRM	Simple Linear Regression Model
SRA	System Risk Assessment
TDMA	Time Division Multiple Access
TPSN	Timing Sync Protocol for Sensor Networks
TS	Technical Specification

UAS	Unmanned Aircraft System
UAT	Universal Access Transceiver
UAV	Unmanned Aerial Vehicle
UTC	Universal Coordinated Time
WSN	Wireless Sensor Network

ACKNOWLEDGMENTS

Confía en el tiempo, que suele dar dulces salidas a muchas amargas dificultades.

[Trust time; it usually provides a sweet way out of many bitter challenges.]

–Miguel de Cervantes, *El Ingenioso Hidalgo Don Quijote de la Mancha*

I would like to dedicate this work to the people from whom, along these four years, I have learnt to trust time. It is not only about the knowledge or the craft of research making. It is also, in my understanding, about the art of putting one's energy to work, harnessing setbacks, and unceasingly aiming at the highest standards of quality to produce the outcome of which I cannot be prouder.

To Robert Thorson who, in trusting me when I was in most need of it, has been a reference in this quest towards a PhD in space studies. To Dr. Joseph Vacek, who represents the best example of what he once told me about the friendliness, warmth, and unconditional support of the people in this wonderful part of the world. To Dr. Mark Askelson, much indebted for all the suggestions he made during the scarce time he has. To Paul Snyder and to Dr. Pablo de Leon, whose trust towards me throughout the years has gone beyond their acceptance to participate as dissertation committee members. Finally, I would also like to thank Dr. Saleh Faruque, who gave me the idea for the research.

In this brief note, I would also like to acknowledge my parents and my two sisters. Their presence rises in the horizon, as I write, like silent light beams across the unfathomable waters of the Atlantic. You sowed the seeds of the person I am in the process of becoming.

And to Bridie Moroney, who already knows that without her support during my first years in the United States, this story might have been played over a different tune.

To my wife, Mary, and to my daughter, Nora.

You are always on my mind.

ABSTRACT

This dissertation provides a numerical analysis of a Receiver Only Synchronization (ROS) protocol which is proposed for use by Unmanned Aircraft Systems (UAS) in Beyond Visual Line of Sight (BVLOS) operations. The use of ROS protocols could reinforce current technologies that enable transmission over 5G cell networks, decreasing latency issues and enabling the incorporation of an increased number of UAS to the network, without loss of accuracy. A minimum squared error (MSE)-based accuracy of clock offset and clock skew estimations was obtained using the number of iterations and number of observations as independent parameters. Although the model converged after only four iterations, the number of observations needed was considerably large, of no less than about 250. The noise, introduced in the system through the first residual, the correlation parameter and the disturbance terms, was assumed to be autocorrelated. Previous studies suggested that correlated noise might be typical in multipath scenarios, or in case of damaged antennas. Four noise distributions: gaussian, exponential, gamma and Weibull were considered. Each of them is adapted to different noise sources in the OSI model. Dispersion of results in the first case, the only case with zero mean, was checked against the Cramér-Rao Bound (CRB) limit. Results confirmed that the scheme proposed was fully efficient. Moreover, results with the other three cases were less promising, thus demonstrating that only zero mean distributions could deliver good results. This fact would limit the proposed scheme application in multipath scenarios, where echoes of previous signals

may reach the receiver at delayed times. In the second part, a wake/sleep scheme was imposed on the model, concluding that for wake/sleep ratios below 92/08 results were not accurate at p=.05 level. The study also evaluated the impact of noise levels in the time domain and showed that above -2dB in time a substantial contribution of error terms disturbed the initial estimations significantly. The tests were performed in Matlab®. Based on the results, three venues confirming the assumptions made were proposed for future work. Some final reflections on the use of 5G in aviation brought the present dissertation to a close.

I. INTRODUCTION

Imagine the deployment of a cluster of Unmanned Aircraft Systems (UAS), (i.e., a multi-agent or a swarm structure) with the purpose to inspect a pipeline section, or any other long-linear infrastructure, located close to a populated area. Transmissions encompass command and control (C2) communications as well as synchronous payload information, such as video footage of the pipeline and current potential hazards nearby. The flight profiles are such that some UAS are eventually operated far from the Visual Observer, in which is known as Beyond Visual Line of Sight (BVLOS) operations. Since this makes direct links from the drones to the ground control station (GCS) unsteady, local cell phone networks are used as a relay. The high capacity associated with these networks, especially when using LTE/5G standards, may be of great benefit to the service provided, which needs to transmit a significant amount of information in real time. Let us further imagine that one UAS is hovering over point A, from where it must transmit information to the operator. Before doing that, however, transmitter and receiver processors must set a common time framework, so that interrogation/responses messages do not collide in the channel. Thus, communication standards establish a process called synchronization, which essentially pre-sets the clocks in the transmitter and receiver clicking simultaneously. Now, another UAS in point B that needs to establish communications with either GCS or the A drone will also go through a similar synchronization process. This process could work based on GNSS, whose signal might be weak in certain areas, or using what is known as a synchronization protocol. In such protocols, a set of timestamped messages (observations) is exchanged, in order to estimate each other clocks' offsets. However, further broadcasting of new messages add pressure on the frequency band, which is already busy with payload information. And busy lines mean numerous attempts to re-connect. Gradually, latency issues in the form of

video glitches begin to show in the UAS' ground control station (GCS). Eventually, outages pop up, and the remote pilot in command (RPIC) loses the UAS signal altogether. This safety concern could be resolved if we addressed the synchronization process in a different manner.

Imagine that a "silent" synchronization protocol was used instead. A protocol which would not need to send additional messages: We could achieve a minimum impact of these background processes on the busy communication link, while simultaneously not impairing the synchronization robustness. One of the synchronization protocols that has been studied in recent years is the Receiver Only Synchronization (ROS) protocol. It has the advantage over other protocols that it exploits the timestamps of the messages exchanged with one drone only to synchronize any additional drone in its vicinity without adding messages on the stack. Although many protocols have been proposed in the last decade to synchronize different types of nodes, none of them has been specifically designed with UAS needs in mind, to the best of the author's knowledge. This dissertation proposes such a protocol and provides a qualitative evaluation of its adaptation to UAS. It also assesses the impact of different noise sources on its robustness. Finally, it introduces a wake/sleep cycle, whereby frequency band occupation can be reduced even further, while simultaneously helping the UAS platform save energy resources.

As a prelude to the requirements of such a specific synchronization scheme, let us set the background for the ubiquitous presence of drones in our society.

UAS Status and the Question of BVLOS Operations

UAS, commonly known as drones, are aircraft intended to be operated without human intervention from within or on the aircraft (Small Unmanned Aircraft Systems, 2016). These UAS' features compel the inclusion of associated elements such as ground control (Peniel &

Granshaw, 2018) to allow the vehicle operation from a remote site. Today, we are witnessing an explosion of drone applications in a swelling trend which is not expected to slow down in the foreseeable future (Bartsch et al., 2017). As of May 04, 2021, the FAA reported over 873,500 drones registered in the U.S., of which roughly 40% are for commercial purposes and 60% are recreational, and nearly 226,000 pilots certified to operate them (Federal Aviation Administration, 2021). Parcel delivery (Marinelli et al., 2018), farmland surveillance, or oil pipelines monitoring (Otto et al., 2018) are just a few of the many applications of these devices. Drones are becoming ubiquitous in our landscape, irrespective of whether we live in an urban or rural environment. For these reasons, we can conclude that drones' presence will continue to grow in common use.

After the Huerta v. Pirker case (2014), the National Transport Safety Board (NTSB) forced the federal aviation authority in the United States, i.e. the Federal Aviation Administration (FAA), to publish regulatory material over drones. The result came to be 14 CFR part 107 on small UAS, published in June 2016 (Small Unmanned Aircraft Systems, 2016), which covers operations of civil drones weighing less than 55 pounds on takeoff. This venue provided a regulatory framework for an increasing number of drones' uses in our National Airspace (NAS). However, it also became apparent that the initial regulatory boundaries would eventually need to be trespassed: For instance, the general rule stipulates that it is unlawful to operate drones in BVLOS, something which, for instance, seriously limits the range to provide parcel delivery. Fortunately, this limitation may be circumvented through the issuance of a special waiver.

The process through which the FAA may grant a BVLOS waiver constitutes a complex interaction between the applicant and the authorities, involving the development of a robust System Risk Assessment (SRA). As the number of waivers applications increases, the FAA is

realizing that the approval process should evolve from what was stipulated in 8040.6 Order (Federal Aviation Administration, 2019) on UAS safety risk management towards a swifter, evidence-based process. This paradigm would need the implementation of new technology to increase the confidence levels between applicant and approving authority.

In this sense, in the last six years several enablers have been considered to realize the implementation of BVLOS, such as Automatic Dependent Surveillance Broadcast (ADS-B), or the use of communication satellites (Balsi et al., 2019). Indeed, when there is no direct line of sight between the platform and the Ground Control System (GCS), ensuring a trustworthy link with the ground operator becomes a challenge. With the explosion of 5G applications, there was a rush to propose this standard as one of the most reliable means to relay information between the airframe and the GCS, and for a cause: 5G could provide up to 100 times higher bit rate and five times lower latency compared to regular 4G/LTE (Wu et al., 2021). However, these speeds impose a burden on synchronization processes, which must be accomplished in a very short time, down to the order of one microsecond (Third Generation Partnership Project, 2021c). The FAA and the Federal Communications Commission (FCC) are still working to assess new BVLOS technical enablers within two initiatives, namely the Integrated Pilot Program (IPP) and the Program Experimenter License. These programs have set the ground to investigate the feasibility of cellular-connected drones in the National Airspace (NAS).

The IPP implementation, enacted in 2012 through the FAA Modernization and Reform Act, began an effort aimed at investigating, among others, BVLOS solutions for drones (section 332c of the FAA Modernization and Reform Act, 2012). This is the main mission of one of the participants, the North Dakota DoT, who to this purpose set up the Northern Plains UAS Test Site (NPUASTS). Moreover, the FAA has recently signed Memoranda of Understanding (MoU)

with private business, such as Verizon in 2021, to investigate the feasibility of cellular-connected drones flying BVLOS operations (Moody, 2021).

Similarly, the FCC established the Program Experimenter License in 2013 to speed up tests at specific sites without this agency's explicit authorization. Since then, more than 2,000 licenses have been granted yearly for universities and research centers to experiment with new communications technologies, such as 5G (Federal Communications Commission, 2017). For instance, this allowed North Carolina State University and Northeast University experimenting with frequency bands allocated to 5G cell phone services and assess their feasibility for drones C2 services (Federal Communications Commission, 2021a). The FCC also grants waivers to private initiatives in order to investigate the use of these cellular bands for drones' connectivity (Federal Communications Commission, 2021b). As a result, the FCC issued two notices requesting the public opinion on a petition to use 5GHz (Federal Communications Commission, 2021c) and 450 MHz (Federal Communications Commission, 2021d) band for UAS C2 services.

Pros and Cons of LTE/5G as a BVLOS Solution

The use of LTE /5G ground stations as an assisting network to establish drones' communications in BVLOS operations has several major advantages over other communication links. Some of these advantages are: ubiquitous accessibility, enhanced performance, ease of monitoring and management, robust navigation through the GPS system, and cost-effectiveness. For this reason, numerous business initiatives are or will be using LTE/5G technology to help deploy BVLOS operations to provide their services. AT&T and Uber, for instance, have set up a joint venture on cargo drones flying BVLOS (Fletcher, 2019). The former has also inaugurated an innovation studio specifically targeting drone 5G connectivity (Spires, 2021). Finally,

Skyward is a service for drones which, among other things, will allow them to connect via Verizon's network (Scott, 2021). Time synchronization, however, remains a key issue for these systems. Standards bodies such as the Third Generation Partnership Project recognizes this, when requesting a GNSS-based timing reference (Third Generation Partnership Project [3GPP], 2021b). A highly accurate system relying on satellite-based atomic clocks, GNSS-based timings could in principle solve the issue of 5G busy channels, but they also have other concerns that make them a less than optimum solution. For instance, the signals could be intentionally or unintentionally jammed. Therefore, 3GPP also indicates that potential degradation of the GNSS signal should be counteracted with backup systems for time resiliency (Third Generation Partnership Project [3GPP], 2021b). Thus, alternative, protocol-based synchronization methods have also been explicitly discussed for asynchronous networks (Son & Lee, 2007).

In addition to LTE synchronization issues, there are other aspects that challenge the use of LTE in UAS communications: The need to standardize regulations at an international level[1], LTE downlink and uplink interference[2] (Ivancic et al., 2019), and LTE/5G interference with radar altimeters used in aviation[3]. Although the present dissertation does not analyze these aspects, they are still recognized as potential roadblocks on the path to make LTE/5G the standard in BVLOS communications.

---

[1] The International Civil Aviation Organization (ICAO) and the European Aviation Safety Agency (EASA) are two international bodies with which some kind of cooperation will be expected in the future.

[2] Qualcomm Technologies Inc. (2017) issued a very interesting report on this subject.

[3] Interference of LTE/5G frequencies with radar altimeters in manned aircraft has been reported recently (Radio Technical Commission for Aeronautics, 2020).

The present section discussed the feasibility and limitations of LTE/5G links for UAS. At this point, the reader may question the necessity of such a system: Why would LTE/5G be necessary, after all? 5G standards mean, of course, faster communications. With them, the need of more accurate transmission processes is unavoidable, if we want to keep latency issues at a low level. But what would the drawbacks be of using less accurate clocks? The following section turns onto this question, by presenting some applications in need of a highly accurate time synchronization system.

Background: The Importance of Time Synchronization

UAS clusters, be it in a swarm or multi-agent structure, constitute an example of what is known as Wireless Sensor Networks (WSNs): a set of defined elements interconnected through non-physical means. Some of these WSN require highly reliable synchronization methods to operate as expected. A list of their applications is presented in Serpedin and Chaudhari (2009). Their list, complemented with additional information from works published during the last decade, include:

- Data fusion and TDMA use.
- Power management.
- Transmission scheduling.
- Localization protocols.
- Security protocols.
- Tracking protocols.

Data fusion and TDMA Use

Many processes require data merging from different sensors. To achieve a minimum energy consumption in the transmission process, an efficient packet scheduling is necessary. Thus, some researchers have turned their attention towards techniques such as Time Division Multiple Access (TDMA) (Akila et al., 2017). TDMA may exploit the classification of data packets based on their urgency to be transmitted and assign timeslots to each node. These timeslots are repeated cyclically, so that in every cycle each node has a determined number of "windows of opportunity" to transmit their information. In order to minimize the risk of broadcast overlap, which might translate into unacceptable delays and waste of energy, these slot timespans are tight, constrained to a few milliseconds, including a guard time to allow for synchronization errors. The higher the synchronization accuracy, the narrower this guard time can be without increasing the interference probability, and hence more time slots can fit in one cycle: Consequently, the capacity of the communication channel is increased. In summary, TDMA systems are as efficient as the time synchronization process, and hence the importance of this process for a good system performance.

TDMA may also work in the case of random access to the communications channel. Here, the probability of two transmitters using the same slot simultaneously may be minimized by simply increasing the number of timeslots. However, the process does not guarantee that the channel is free whenever a node –a drone in this dissertation's case– needs to transmit information. Some algorithms, based on nearly-conflict-free assignment of slots, tend to minimize this problem (Ergen & Varaiya, 2009). Ergen and Varaiya's approach, for instance, consists of defining a multi-hop system, where a schedule is pre-built that avoids simultaneous transmission of adjacent nodes. Other algorithms such as Carrier Sense Multiple Access (CSMA)

can minimize collusion risk using pre-sensing techniques to check whether the channel is already busy. However, no synchronization process would entirely remove the possibility that third-party communications interfere with our signal, and if they did, the solution would come at the cost of additional delay (Abedi & Pourhasani, 2021).

TDMA techniques are used in the Automatic Dependent Surveillance Broadcast (ADS-B) system, which is required for most pilot aircraft in the U.S. airspace. Indeed, its Universal Access Transceiver (UAT) mode makes use of TDMA to transmit information packages between airspace users.

Power management

Power management, a particularly sensitive aspect for drones, would also greatly benefit from highly accurate synchronization processes. Establishing any communication link involves a series of steps –radio transmission, sensing and data processing– which utilizes power resources at various degrees (Nikolić et al., 2014). It is also generally assumed that sensing activities constitute the "cheapest" step, in terms of power consumption (Nikolić et al., 2014). Hence the existing literature, in an effort to minimize scarce resources, has focused on reducing radio transmissions while simultaneously increasing sensing times (Nikolić et al., 2014). An optimum solution might even include wake/sleep schemes, whereby a node is ready to receive the information precisely when another node is about to send it. Thus, accurate timing involving good synchronization between nodes is key for the system to work efficiently.

The need to maximize power savings brings forward a concept that will be explored in the present dissertation. Broadcasting information is more energy expensive than receiving the broadcast. However, as Nikolić et al. (2014) showed, the energy savings are less impressive if

they come at a cost of extending sensing times. Therefore, the benefit of extending purely sensing activities for the sake of energy savings might be minimal. To counteract this effect, Nikolić et al. (2014) proposed the use of a duty cycle, whereby sleep modes are introduced during which the information received in a previous period is used to keep clocks synchronized. As the sleep modes get longer, however, there is a risk of clocks offsetting, and hence losing the synchronization between nodes. Therefore, a certain amount of wake modes must be interleaved to "catch up" with clock drift. In summary, it is the combination of sleep and wake (or active) modes what makes a duty cycle-based, synchronization system efficient.

Transmission Scheduling

The scientific community has shown interest in delivering increasingly efficient wireless networks. As a result, a significant number of schemes based on scheduled transmissions similar to TDMA have been developed in recent years. TDMA schemes, as indicated, are made of a series of distributed slots throughout time. Other schemes exist, which consist of a distribution of different resources, such as the emitted power (Lyu et al., 2017). In all these schemes, control stabilization is key for an appropriate transmission. Stabilization methods aim at selecting the optimum times during which the transmissions have the higher chances to reach the receiver. To achieve this, Lyu et al. (2017) proposed a feedback control mechanism to minimize the waste of time and resources, and to avoid transmissions during those periods when the channel is unreliable because of the relative position of the nodes, weather phenomena or random processes.

Localization Protocols

Serpedin and Chaudhari (2009) mentioned additional methods that would benefit from a highly accurate synchronization protocol. They classified these methods into localization, security, and tracking protocols.

Localization protocols aim at enhancing the information provided by the sensors, which transmit information about an event of interest, by adding precise information on "where" the event happens. Hence for these applications, the event location must match the recorded information with a high degree of accuracy, lest the user loses information on where exactly a given event occurred. Chelouah et al. (2018) provided an exhaustive survey of localization algorithms.

Security Protocols

Penttinen (2016) has recently analyzed security risks affecting WSNs. He also proposes several protection methods against cyber-attacks, jammers, and similar potential sources of link interruptions. Penttinen (2016, p. 257) provided a list of security-related items to be considered in the security process. Among them, he cited Timing over Packet (ToP), a method to organize the delivery of data packages over a specific scheduled known only by the transmitter and receiver. Timing over Packet allows to minimize the risk of security attacks through these transmission bursts at fixed times, in a similar manner that a sparrow would avoid being trapped by a bird of prey by only flying to the next tree, if and when necessary, and taking the shortest route to do so. Any efficient ToP system would need highly synchronized clocks, so that the receiver is ready to capture the information whenever the corresponding sender is in transmission mode.

Tracking Protocols

Yi et al. (2015) proposed a tracking technique which can be applied with non-concurrent (non-synchronized) signals of transmitting beacons for submerged receivers, such as submarines, where two-way communications may be seriously degraded. The authors showed that synchronized clocks increase the accuracy of tracking processes dramatically. Their results target tracking methods for vehicles moving below the water and are therefore not directly transferrable to the UAS case. However, they constitute a practical example of the importance of time synchronization in localization methods based on the differences in arrival times of signals originated by two different beacons. Interestingly, two of the oldest navigation systems that aviation has used in the past, OMEGA and LORAN-C, are also based on TDoA techniques.

This section showed that there is a need for an accurate synchronization protocol specifically tailored to UAS characteristics. A fair question arises, however, on why the use of GPS time synchronization service, although very accurate, might not be enough. After all, practically all UAS use GPS for location purposes today, and for a reason. Indeed, the precision of a GPS clock, measured as the root mean square (RMS) value, is of about 5 nsec (Senior et al, 2008). Unfortunately, the communications latency in these systems increase significantly due to relativistic, ionospheric and tropospheric effects (Solomon et al., 2011). These researchers estimated the synchronization process time using GPS to be somewhere between 100 msec and 1 sec. For drones such as the Insitu's ScanEagle flying at their maximum speed of about 45 m/sec, this means that the initial position of the airframe may be off for about 45 m, something unacceptable in the vicinity of airports or restricted zones. The next section describes how GPS time synchronization works.

GPS Time Synchronization

Sensor networks equipped with a GPS receiver have a distinctive opportunity to synchronize their clocks with a very precise system: the GPS atomic clocks installed in the satellites. These clocks are adjusted, save for an accumulated number of lapped seconds, with the Coordinated Universal Time (UTC), which is established by the Bureau International des Poids et Mesures (BIPM) (Senior et al., 2008) and is widely used for time measurements around the world.

Time synchronization using GPS, or any other GNSS system such as the European's Galileo, operate under the principle of messages' timestamping. The receiver just needs to evaluate the travel time and subtract this information to the current local time to know what the clock's difference, or clock offset, is in order to adjust its own clock pace. The travel time is deduced by knowing the distance from the transmitting satellite, which in turn is calculated through the satellite's position using its ephemeris data. GPS.gov (2019) cites some of the applications that use GPS timing in their systems.

GPS uses Cesium (Cs) or Rubidium (Rb) atomic clocks in their satellites (Senior et al., 2008), which are one of the most stable oscillators, as these clocks really are, ever built. These oscillators provide the basis for the GPS messages' timestamps. Because the satellites' orbit speed approaches 14,000 km/h, relativistic corrections must be introduced to keep the reference functional (Zhang et al., 2006). Other corrections due to ionospheric effects must also be accounted for (Rose et al., 2014). These clocks' precision is of the order of nanoseconds. Since it takes three nsec, or $3*10^{-9}$ sec, for a radio wave to travel one meter, a precision of, say, 10 m could be achieved if the clocks did not diverge more than 30 nsec from a given reference.

GPS time is a scale which begins at midnight, or 0h Coordinated Universal Time (UTC), on January the 5th to the 6th, 1980. The difference between GPS and UTC frames is currently established at about 18 sec, due to the leap seconds the UTC adds periodically to account for perturbations from different sources such as the Earth's rotation. Given that this GPS-UTC difference is fixed, the satellites' coverage allows GPS to be the world provider of "UTC time" par excellence (Panfilo & Arias, 2019).

The Problem with GPS as a Synchronization Source

GPS world coverage provides an obvious advantage for the setting of a common timeframe for wireless sensor networks (WSN). Unfortunately, it also suffers from several drawbacks, which pose a huge pressure on GPS to perform as the reference system for time synchronization. Three of these drawbacks are explained below:

1. As it has already mentioned, failure on synchronizing the clocks before transmission may result in unacceptable latency levels. Current standards set a latency in communicating links of between 1 sec and 40 msec for C2 communications in non-automatic flight, depending on the control mode (3rd Generation Partnership Project [3GPP], 2021a). This latency, evaluated as the time lapse between the generation of a message in the transmitter and it being received by another node, is the maximum accepted delay of a signal. One of the main contributors to latency is an inefficient time synchronization system. For instance, in the case of busy channels, the time synchronization process keeps trying to transmit until the clocks of the systems involved in the communication processes are ticking simultaneously, within an accepted degree of accuracy, and

C2 or payload communications can be established. The GPS system may not be able to comply with these standard latency times, at least when used as a sole means. Therefore, a complementary system is required, and so it is recognized by 3GPP (3rd Generation Partnership Project [3GPP], 2021b).

2. GPS is under the control of the Department of Defense (Federal Aviation Administration1, n.d.), and not under an Air Navigation Service Providers (ANSP) such as the FAA in the U.S. This fact puts a serious hurdle in the service continuity. Maintenance outages are not rare events and need to be considered (Eier & Sharples, 2009). GPS can also be turned off for strategic reasons at any time. Thus, some unmanned vehicles using GPS receivers may be equipped with inertial systems (INS) as well, which may provide a backup for short interruptions. Other researchers propose the use of micro-electrical mechanical systems (MEMS) technology to resolve uncertainties during outage periods of the GPS signal (Mostafa et al., 2019). However, the use of any type of hardware increases the cost of the navigation system, where a software-based application might do the work just as well. This is the solution proposed the in the present dissertation.
3. GPS has also been the target of intentional interference (jamming) with simple devices that emit just over 30dB in power (Sun & Amin, 2005).

GPS weaknesses presented above may impair signal continuity during critical operations. Sometimes, manned aircraft suffer from GPS signal losses due to various cases (e.g., jamming or spoofing). This was the case when an aircraft was trying to land in El Paso International Airport in May 2020, and the pilot was forced to perform a landing in an alternative runway which was

not equipped with vertical guidance, hence in a higher Controlled Flight Into Terrain (CFIT) maneuver. Fortunately, the plane made it to the ground without further consequences. (Harris, 2021). Hundreds of similar anomalies in GPS signal reception, where reported to the FAA within just a few months of 2017 and 2018 (Harris, 2021). The situation could be even more critical if a UAS lost the GPS navigational reference near an active runway.

The impact of some of these drawbacks may be reduced with the aid of complementary, if not alternative, synchronization schemes. The next section presents such a scheme, in the form of a protocol that establishes a two-way messages exchange between two sources.

Time Synchronization Protocols

The problems identified in the previous section could be minimized with an efficient time synchronization protocol. Such a protocol would benefit from the following characteristics:

- It would be independent from GPS, which means that it would be inherently invulnerable to any maintenance outages that the GNSS systems go through periodically. By "inherently" this author means that a protocol does not need to go through maintenance outages. However, in certain hierarchical protocols such as the Precision Time Protocol (PTP) discussed below, some nodes connect directly with time sources that do need periodical maintenance of some kind (Institute of Electrical and Electronics Engineers, 2019). In this case, WSN nodes will unavoidably experience interrupted transmissions.
- Synchronization protocols do not need to be managed by any specific entity who may determine the signal availability based on factors unrelated to the synchronization process. This is particularly true, although not exclusive, for the

synchronization type known as internal synchronization. Internal synchronization, as opposed to external synchronization, means that clocks synchronize among themselves, creating a common reference which does not need to be linked to UTC (Swain & Hansdah, 2015).

- Because of their local application, synchronization protocols may be less prone to an intentional jamming aiming at a more global scale. This fact may provide an additional security barrier and strengthen the communications linkage. However, clock synchronization protocols are not completely free from this threat and hence some solutions have been proposed to minimize the impact of jamming on resources consumption and time delays (Pajic & Mangharam, 2010).
- Some protocols are specifically adapted for minimum energy consumption, using low duty cycles with a smart distribution of active and idle modes. See, for instance, Nur et al. (2017).

Synchronization is a process that must be sustained through time. Noises of varied natures affect initial estimations and must be considered in the models. Basically, noise can be attributed to internal and external causes. The former case is related with the performance of the clock oscillators in the individual processors. The latter case can be affected by a variety of causes that may impact the quality of transmissions. The following two sections provide a summary of both of these cases.

Internal Noise Sources: Jittering in Clock Oscillators

In order to understand why a clock loses its synchronization through time, we will take a closer look at local oscillators, which are the devices responsible for clock *tickings*. An exhaustive introduction to them can be found in classic books such as Frerking (1978). The most common oscillators have a core of quartz crystal (Bottom, 1982, pp. 11-37). These crystals originate the oscillation by means of the piezo-electric effect, whereby dipoles convert mechanical vibrations of the crystals into an electrical signal. The mechanical vibration comes in the form of a set of harmonics, also known as resonant modes, of a fundamental frequency. This fundamental frequency is characteristic of each crystal unit. When excited in these modes, ideal quartz crystals achieve a highly regular output signal. Bottom (1982, p.9) mentions that some crystal units are reliable even above 150 MHz. In order to sustain the oscillatory phenomenon, part of these crystals' output, in the form of a voltage, is fed back to the input using an amplifier (Bottom, 1982, pp. 50-62). Last, an analog to digital converter transforms the remnant output into a periodic, normally squared-shaped, signal. Unfortunately, although their stability ranks among the highest found in nature, quartz crystals are not perfect, either. Bottom (1982) provides some of the reasons for these imperfections, which manifest in the form of jitter, as mentioned by Lee and Chin (2016):

- Impurities in the crystal structure. Some of these impurities can be removed or minimized when the crystal is brought under an intense electric field at high temperatures (Bottom, 1982, pp. 23-24). This process has been applied to clocks that are being used in high ionizing environments, such as in the Global

Positioning System (GPS) satellites. A quartz crystal that has undergone such a process is called a swept crystal.

- The generation of thermal stresses may change the distance between crystal nodes and their angles (Bottom, 1982, pp. 47-48). As a result of these changes, the resonant properties of the crystals are modified. Basically, the crystal undergoes a process of time diversion, by which the frequency is not kept at a regular pace, i.e., it spaces apart or contracts through time.
- Aging. Bottom (1982) dedicates a full chapter to analyze how this phenomenon affects the performance of quartz crystals (pp. 208-219). The consequence of aging is revealed through:
    - Surface deterioration, which has been studied in what is known as the Bremsstrahlung effect. Broken bonds may appear on a surface layer in a strained position, which alter the electrical field transmitted by the crystal.
    - Surface contamination, which is particularly critical when volatile materials are present in the crystal holder.
    - Some electrodes in charge of transmitting the electrical field to the crystal can also wear out with time, which results in imperfect transmission.
    - Molecular rearrangements within the crystal may also occur. This effect is accelerated with rising temperatures.
    - The leaking of crystal holders through time, which are filled with N_2 or He, is unavoidable. As a consequence of this leakage, clock drifting may occur.

Temporal, random jittering in clocks are the cause of clock offsets and skews (Zhang et al., 2020). Clock offsets manifest as differences in their instantaneous readings, whereas linear, gradual diversion of these differences through time, are known as clock skews. The model proposed in this dissertation makes an extensive analysis of clocks offset and skew estimations.

External Noise Sources

Even in the ideal case of zero offsets in the oscillators themselves, transmissions suffer from a variety of noise effects that may alter their throughput. These effects impact the receiving time of otherwise regular clock tickings. Kopetz and Schwabl (1989, as cited in Elson et al., 2002) assign these time variations to four periods of the communication process:

- Send time, or the time spent to construct the message. This includes the time it takes the message to be transferred from the host to its network interface.
- Access time, or the delay incurred in waiting to access the transmission channel. In the case of TDMA models, for instance, a saturated channel may delay the broadcast by several milliseconds.
- Propagation time, sometimes called transmission time, which accounts for the travel time across the channel. In the case of WSN, this channel is the atmosphere, where signals travel at a finite speed close to the speed of light c.
- Receive time, or the time it takes to generate a reception signal in the receiver.

To understand the nature of the delays produced in each of these periods, we must first consider how network communication works. The Open System Interconnection (OSI) model is a valid conceptual model used by engineers to understand how information packets are managed

in a given network. Although OSI is not a standard, it provides a simple approach to understand how information transmissions work. A brief introduction to OSI is offered below.

Zimmermann (1980) defines the OSI model as: "a structuring technique which permits the network of Open Systems to be viewed as logically composed of a succession of layers, each wrapping the lower layers and isolating them from the higher layers […]" (p. 426). We could thus imagine a given architecture as composed of seven distinct but interconnected layers. If we consider that each layer adds value to the services provided by precedent layers, a layer hierarchy is built up. From top to bottom, i.e., from the most complex to the less value-added layer, these layers are represented in Table 1:

Table 1

The Seven Layers of the OSI Model

Layer 7	Application layer, where received synchronization messages are processed to get an estimate of clock offsets
Layer 6	Presentation layer
Layer 5	Session layer
Layer 4	Transport layer, where noise may be generated
Layer 3	Network layer, where noise may be generated
Layer 2	Data link layer, where noise may be generated
Layer 1	Physical layer

Many sources describe the specific functions of each layer in detail, such as Zimmermann (1980). The Data Link Layer is where the means to establish, maintain, and release data links between entities are produced (Zimmermann, 1980). In the Network Layer, the message to be transported is given a defined structure adapted to the channel (Zimmermann, 1980). Finally, the

Transport Layer provides the physical transport service through the means of components, such as electrical circuits (Zimmermann, 1980). These components' performance may be affected by diverse phenomena, such as extreme temperatures or aging and, therefore, introduce delays in the system. Mapping these layers to Kopetz and Schwabl's (1989, as cited in Elson et al., 2002) scheme specified above would allow us to establish the following relations:

- Send time delay takes place mostly at the Data Link layer. Send times are not considered to be an important contribution to the total delay in the scheme proposed in the present dissertation.
- Access time delay takes place mainly at the Network Layer. This type of delay may happen, for instance, when a TDMA scheme is applied with a random access to the available time slots. A source which is searching for an available slot to send information may experience a significant amount of delay when the channel is busy. Pinto and Almeida (2018) briefly described this problem and proposed a delay-tolerant solution over Commercial Off-The Shelf (COTS) WiFi.
- Propagation time delay takes place at the Transport layer. In the protocols proposed in the present dissertation, nodes are considered either fixed or travelling at low speed and hence, the propagation time delay is constant in every transmission.
- Finally, delays related to receive time take place at the Network and Data Link layers. As with send time delay, receive time delays are not considered important contributions to the system total delay.

The last many years have witnessed a surge of synchronization protocols adapted to wireless sensors: Sundaraman et al. (2005); Faizulkhakov (2007); Rhee et al. (2009); and Yiğitler et al. (2020) are some of them. However, to the best of the author's knowledge, so far none of the research performed has focused on the special characteristics of the UAS. The lack of works that could help understanding the effects of noise and clock jitters hinders our efforts to enhance the quality of command and control (C2) and payload transmissions for drones. As these devices become more ubiquitous in our society, we can no longer overlook this need. This is the vision to which the present dissertation is committed.

The Present Work

This work analyzes qualitatively a time synchronization protocol for Unmanned Aircraft Systems (UAS) that may replace or complement GPS-based synchronization systems. This protocol could also complement synchronization systems in LTE networks to decrease time latencies, to which 5G-based services are particularly sensitive:

- First, it tests a proof of concept for a synchronization protocol specifically tailored to UAS's needs, which minimizes the channel use and therefore also the latency in the system. As shown later, this feature makes Receiver Only Synchronization (ROS) protocols a very interesting choice. A simulation developed in a MATLAB®'s R2021a version is presented, and results thereof are discussed. Results reveal how many iterations and observations in the recursive scheme are sufficient. These estimations are compared to the real values through a recursive scheme over regression analyses. The scheme is based on Chatterjee and Venkateswaran's (2015) work, but the evaluation performed in

the present dissertation includes a higher number of parameters and analyzes the effect of using different noise models.

- In the second part, a string of sleep modes, when results are extrapolated through time, is introduced in the scheme, and its effect on the model accuracy is evaluated.

Problem Statement

There is a need to provide efficient time synchronization between UAS clocks, to enable low-latency communications between the UAS and the Ground Control System (GCS), or between UAS in flight. Current UAS software uses GPS as its main source to synchronize clocks. However, as discussed earlier, GPS has some drawbacks that reduces its efficiency. The use of an efficient time synchronization protocol may contribute to signal stabilization and reduce time latency, particularly in areas or periods of poor GPS coverage. With the surge of commercial UAS operations, the need to provide this stability is key to ensure that drone operations in BVLOS remain safe.

Purpose of the Study

The purpose of the present dissertation is to analyze the applicability of a Receiver Only Synchronization (ROS) protocol to UAS. This applicability is discussed under two perspectives: Accuracy and resources utilization. First, it proposes a recursive algorithm to estimate the clock offset and clock skew with respect to the reference node, normally the GCS. Second, it applies a duty cycle to minimize the use of channel resources. The duty cycle is composed of wake and sleep modes. During the wake mode, the UAS overhears broadcasts exchanges between a second UAS and the GCS, in order to synchronize the internal clock with the system reference. A UAS in sleep mode discontinues the receiver, and the internal processor extrapolates the information

received up to then. The duration of these idle modes is determined by the need to keep the accuracy of the estimations bounded with respect to the true values.

Research Questions

The present study aims at answering the following questions:

1. Which parameters characterize a recursive algorithm that estimates the clock offset and clock skew over a ROS synchronization protocol using a duty cycle?
2. Which of these parameters' values provide an optimum result in terms of accuracy and resources utilization?
3. How does the choice of the noise model affect the efficiency of the selected scheme?

Proposed Methods

This work will develop and run simulations on MATLAB® to obtain results. MATLAB® owner is Mathworks.

Methodology Rationale

MATLAB® is widely used in academic STEM programs and is considered an ideal platform that allows for an initial assessment of a broad selection of engineering and scientific problems. The results of the simulations will be smoothed out by using recursive runs, which minimize the effect of random parameters introduced in the system.

Although it is reasonable to use MATLAB® to obtain theoretical estimates, the results of field tests may provide validity on some aspects of the construct and its content. For instance,

this study assumes that the random portion of delays are autocorrelated (see below). This assumption should be confirmed with real data. Also, field tests may provide evidence of new variables that influence the results, in which case they should be incorporated in the model.

This work provides the grounds on which the field tests may build on. The present author believes that they may also help validate the results obtained here.

Research Assumptions and Limitations

This research is framed within a set of assumptions listed below, whose rationale may be found in different chapters of this dissertation:

- When applying the Receiver Only Synchronization (ROS) model to the proposed synchronization protocol, the author assumes that the UAS involved are within the network coverage of each other.
- The random portion of delays are one-lagged, linearly auto-correlated. That is, they follow a linear scheme whereby a delay at an instant *i* is proportional to the delay in the previous instant *i-1*. In mathematical language, $\omega_i = \rho\omega_{i-1} + \nu_i$, with $\{\nu_i\}$ being the disturbances, distributed according to a given statistics. The correlation coefficient must be $|\rho| < 1$ for the solutions to be converge to a finite value. Some comments on this –strong– assumption are made in the conclusions analysis corresponding to the first part of the tests –Chapter 4.
- Node A in the model (see Figure 5) transmits at regular intervals, in other words, the set $\{T_{1,i}^{A}\}$ is composed of timestamps equally spaced in time.

This chapter discussed the importance of a robust time synchronization system to strengthen UAS connectivity, particularly when BVLOS operations are involved. One of the most critical aspects is the need to keep communications latency below a specific limit, which GPS cannot guarantee, at least everywhere all the time. Several examples have been analyzed that demonstrate the problem's currency. The next chapter summarizes the research literature in the field.

II. LITERATURE REVIEW

Whereas the preceding chapter claimed an unfulfilled need for accurate synchronization protocols specifically tailored to UAS, this chapter explores recent contributions in the field. During the last decade, many works have appeared that proposed new synchronization protocols for Wireless Sensor Networks (WSN), each of them constituting solutions that meet specific requirements. Also, the existing literature is periodically augmented with the publication of surveys that suggest various ways to categorize these proposals.

One of the earliest examples of such surveys is in Sundaraman et al. (2005). They compared nine existing synchronization protocols from a quantitative–such as precision and network size, as well as a qualitative perspective, precision, accuracy, cost as a measure of energy efficiency and complexity. Faizulkhakov's (2007) work has been considered as the first effort to provide a consistent classification of synchronization protocols (Yiğitler et al., 2020). Rhee et al. (2009) expanded the classification types from Sundaraman et al. (2005) and provided a comparison between six most common protocols in the period. Last, Yiğitler et al. (2020) provides a recent perspective on different surveys appeared in the last 15 years –included those mentioned here–. In general, it is observed that each survey brings the attention to different protocol schemes depending on the trends of the period and thus, although they are all similar in broad terms, some of the classification criteria are still in an evolutionary process. It is not the intention of this section to provide a thorough analysis of all the surveys appeared recently –for an example of such a list, see Yiğitler et al. (2020). Rather, the intention is to highlight the novelty of the topic through the evolution of the classification schemes. In this sense, the following table compares the criteria presented in works only four years apart, those of Sundaraman et al. (2005), Faizulkhakov (2007) and Rhee et al. (2009). Note that the different

classifications are not consistent, and a given group of one author may include a set of different groups identified by another. In this sense, Faizulkhakov's (2007) apparent short list may be misleading.

Table 2

Comparison of Classification Schemes for Synchronization Protocols from Three Different Authors

	Sundaraman et al. (2005)	Faizulkhakov (2007)	Rhee et al. (2009)
Master-slave vs. peer-to-peer	**X**		**X**
Internal vs. external	**X**		**X**
Probabilistic vs. deterministic	**X**		**X**
Sender-to-receiver vs. receiver-to-receiver	**X**		**X**
Clock correction vs. untethered clocks	**X**		**X**
Pairwise vs. network-wide			**X**
Application dependent features			**X**
Stationary networks vs. mobile networks			**X**
MAC-layer-based approach vs. synchronization by request			**X**
Lifetime: permanent synch vs. synch by request		**X**	
Synch scale: complete		**X**	

vs. partial	
Synch with shift and frequency synch	**X**
Synch of clocks and translation of timescale	**X**

In the search for the best of these protocols for UAS needs, the present author selected the Receiver Only Synchronization (ROS) model proposed by Chatterjee and Venkateswaran (2015). Reasons of efficiency and global resources utilization led us to make this choice:

1. First, several studies have proven that increasing data rates decreases the data delivery ratio (Kumar et al., 2012, or Meera et al., 2017) because of messages collision in the channel. Thus, a protocol such as ROS would be of utmost interest, since it is based on overhearing messages, rather than on increasing broadcasts –thereby increasing the probability of messages collision, to reach clocks' full synchronization.
2. Second, these same studies have proven the global energy spending in the network increases exponentially with the number of nodes added to it (Meera et al., 2017). A way to reduce this tendency would consist on reducing the radio transmissions in favor of sensing activities, which are generally assumed to be less resources consuming (Nikolić et al., 2014). The ROS model exploits this feature (see below).

The ROS model, as used here, will be enhanced with the incorporation of a duty cycle, whereby active and idle/sleep modes follow in succession. This cycle will allow for an optimum balance between accuracy and resources consumption. The present analysis will make use of

specific statistical tools, and certain assumptions will be made on the so-called delay model. To these tools, we dedicate the second part of the present chapter.

The structure of the present literature review is composed of the following items, as presented in the following sections:

- A list of the most important time synchronization protocols, and an exploration of alternative means to cover existing gaps in the GPS timing service.
- The presentation of the Minimum squared Error (MSE) and Maximum Likelihood Estimators (MLE), as tools to evaluate the proposed model.
- Models used in the existing literature for signal delays, and clock offset and clock skew.

Time Synchronization Protocols

Clock synchronization can be defined as "the process of ensuring that physically distributed processors have a common notion of time" (Sundaraman et al., 2005). For our purposes, this harmonization process is achieved through a specific protocol, i.e., an *exchange of messages* between an emitter and a receiver, or an information source and a sink. This characteristic is what differentiates protocols from other synchronization processes like those GPS-based. Although, as shown later, the statistical theory underlining some of these protocols' rationale go back to the 1950s, most of them have recently become the subject of intense review, catching up with a rush for WSN applications (Sundaraman et al., 2005). Serpedin and Chaudhari's (2009) book constitutes one of the clearest and most exhaustive approaches to synchronization protocols in the field. Although following their approach, this analysis provides updates with works of more recent publication.

As shown above, researchers have been using diverse criteria to classify synchronization protocols. Taxonomies can be very different from source to source. However, for this dissertation's purpose, it is enough to use a simpler approach, proposed in Serpedin and Chaudhari (2009). These authors specified four criteria, namely:

- Master-slave vs. peer-to-peer.
- Clock correcting vs. untethered clock.
- Synchronization approach.
- Pairwise synchronization vs. network-wide synchronization.

Master-Slave vs. Peer-to-Peer

The fundamental difference between these protocol types is that, for the first of them, the nodes constituting the WSN need to be organized in a hierarchical manner prior to exchanging messages. Once they are organized in this way, a local clock synchronizes with its immediate "superior". On the contrary, no such hierarchy exists in peer-to-peer synchronization protocols, where any node can synchronize with any other node in the network (Serpedin & Chaudhari, 2009).

Precision Time Protocol (PTP), which has been widely used to synchronize clocks in both wired and wireless networks (Iordache & Marghescu, 2012), constitutes a representative case to understand the operation of a master-slave protocol, Rhee et al. (2009) mentioned. The first PTP standard was published by the Institute of Electrical and Electronic Engineers (IEEE) in 2008 and has been updated at least twice since then. Its last version was coded IEEE Std 1588™-2019 (Institute of Electrical and Electronics Engineers, 2019). As stated before, a master-slave

protocol such as PTP comprises two processes (i.e., hierarchy configuration and synchronization) and it works as follows.

Hierarchy is established between a root clock, which in PTP is called the Grandmasters clock, and ordinary or boundary clocks. The network selects master clocks based on an algorithm which considers, among other things, the stability of a given clock, its accuracy, and other information such as “priority” of the node, which may be configured externally. The hierarchy thus established constitutes the topological backbone of the network. Once the hierarchy has been established, the synchronization process can start, following any of the approaches presented below. For instance, in the case of PTP two sets of messages are exchanged between the Grandmasters clock and a boundary clock, whereby the delay introduced in the system can be deduced and the local clock drift may be corrected. Indeed, to a certain extent, a GPS-based synchronization process is of a master-slave type: A GPS atomic clock acts as the Grandmaster, to which all local clocks are synchronized. Unfortunately, because of the need to establish a hierarchy before the synchronization process can start (unless the hierarchy is inherent to the system, such as in GPS), most of the master-slave protocols are mostly applied in networks for which resource consumption, both in energy and time terms, is less concerning (Yiğitler et al., 2020). This is the reason why PTP is widely used in computer networks. However, this feature makes them less ideal for UAS applications.

Peer-to-peer alternatives, which do not rely on a hierarchic structure, may be more helpful when time and other resources are limited. Indeed, since the discovery of natural process whereby local synchronization may lead to synchronization on a global scale, such as in the paradigmatic case of fireflies[4] (Buck, 1988), the development of such schemes has brought the interest of many scientists. The scientific community has proposed myriads of peer-to-peer synchronization protocols for a wide variety of purposes. Peer-to-peer protocols are appealing because they are easier to implement: To begin with, there is no need to construct a specific topology first, since no master clock is needed, which contributes to resources savings. Peer-to-peer schemes are, so to speak, more "democratic". Unfortunately, these protocols may suffer from an inherent loss of accuracy: As Lamonaca et al. (2017) indicate, whenever a non-synchronized node plugs into the network it introduces perturbations which may alter the clocks' local stability. This event occurs because the clock skew of the new member averages with the rest, resulting in a clock ticking at a different pace, at least locally. This perturbation, in the form of clock offsets, propagates from node to node across the network, until it eventually fades away. Unfortunately, as more devices connect to the system simultaneously, dampening effects have a longer relaxation time. This effect compromises the continuity of the time reference.

Considering UAS needs at a local level, the potential disadvantages of using a peer-to-peer protocol overcome the complexities of a master-slave configuration. Indeed, none of the time synchronization applications exposed in the precedent chapter must keep UAS

[4] Buck discovered in 1938 that flash communication between a species of fireflies living in Southeast Asia reach a synchronous timing pattern which can be determinant for the species survival. The cited reference is a revisit of the topic, 50 years later.

synchronized with a global time reference such as GPS or UTC. It is enough to synchronize those nodes that need to communicate with each other at a local level. For these reasons, the protocol proposed in the present dissertation is of the peer-to-peer type.

Clock Correcting vs. Untethered Clocks

In clock correcting protocols, the evaluation obtained each time a synchronization process runs is immediately used to modify the clock function, so that two clocks are synchronized at all times. In the case of untethered clocks, on the contrary, the differences obtained after every run are kept in a table, and every time a clock reference is needed the information from the local clock includes the last update noted down (Serpedin & Chaudhari, 2009). This process, of course, does not prevent eventual corrections of the local clock function, although at a much lower rate than in clock correcting protocols. Some authors (Wang et al., 2017) consider that processing time with untethered clocks is simpler and cheaper, from the resources point of view, than clock correcting protocols. This is because, in the former case, the synchronization process takes place while data packages are conveyed, instead of passing the information on to specific synchronization packages, which increase the messages overhead (Wang et al., 2017). However, Wang et al. also stated that untethered clocks result in less accurate processes when a global time reference is needed consistently through time, instead of on a per packet basis. Interestingly, these same authors claimed that synchronization approaches based on overhearing messages, such as in the case of the Receiver Only Synchronization (ROS), are inherently inaccurate because they do not adjust the local clock. The present author cannot find any reason why this should be so. The nature of a protocol based on overhearing mechanisms does not need to be related to the way offset estimations are used to update the local

clocks. The present author notes this as a possible confusion in Wang et al. (2017), which otherwise constitutes a remarkable contribution to the existing literature.

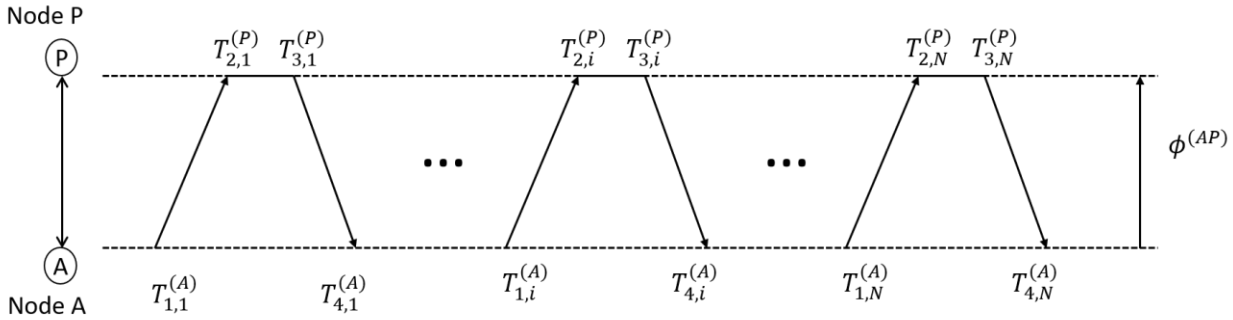
Synchronization approach

Undoubtedly, this is the most common classification criterium for synchronization protocols. Serpedin and Chaudhari (2009) organized several protocols based on the synchronization approach they follow:

Sender-Receiver: Of all the schemes proposed, this is the most popular. In a sender-receiver protocol, two nodes synchronize directly by successive messages exchange. The model can be represented as follows:

Figure 1

Clock synchronization model for a sender-receiver protocol



Note. Adapted from “Synchronization in wireless sensor networks”, by E. Serpedin and Q. Chaudhari, 2009, Cambridge University Press.

In this model, two nodes synchronize with each other in a cycle of N sets of message exchanges, which are called loops. Each of these loops has the same structure: Node A sends a

timestamped message to node P, $T_{1,i}^{(A)}$, which in turn timestamps its receiving time after its local clock as $T_{2,i}^{(P)}$. After a certain lapse, node P sends the response at time $T_{3,i}^{(P)}$, which A receives at its local time $T_{4,i}^{(A)}$. $\phi^{(AP)}$ represents the delay in the transmission, which is traditionally subdivided into fixed and random components. The final section in this chapter analyses the nature of these components in more detail.

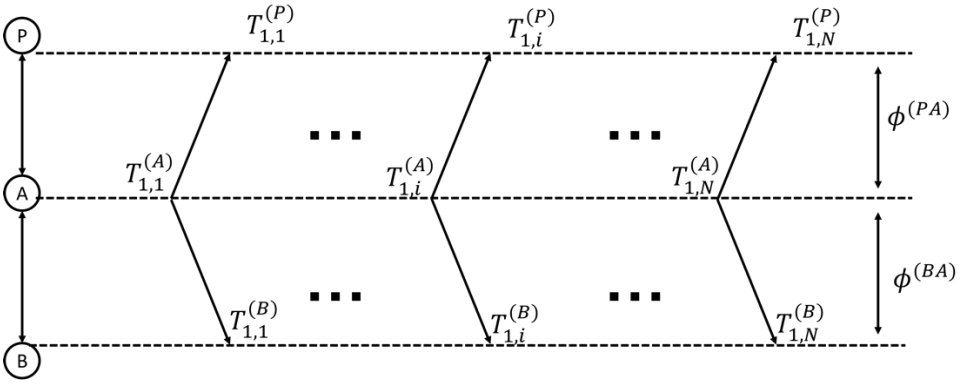
The Network Time Protocol (NTP), together with its several versions since it was first proposed in 1985, uses a sender-receiver scheme (Mills, 1991). Chaudhari et al. (2010) noted that this point-to-point scheme leads to a significant communication overhead. This overhead may eventually lead to latency issues, for which some relief could be provided using an implicit mixed master-slave/peer-to-peer protocol. The scheme, suggested in Chaudhari et al. (2010), would work in two phases as follows: First, a reference station broadcasts timestamped messages. Second, boundary clocks listen to these messages and use them in a peer-to-peer communication scheme to synchronize clocks between themselves. By avoiding direct synchronization with the reference "master" station, the messages overhead can be reduced. However, the need to broadcast messages between neighboring boundary clocks remains a concerning issue. This may balance out the method advantages. Chaudhari et al. (2010) work include an analysis of clock offset estimations using exponential delay distributions.

Receiver-Receiver: Serpedin and Chaudhari (2009) inform that, in these one-way timing models, a reference node broadcasts signals which are received by boundary nodes. These nodes add corrections based on the reception time. Hence the two receiving nodes share a common timeframe and are thus synchronized between themselves. This synchronization is indirect, in

contraposition to the sender-receiver scheme, and it has been proposed in the literature for energy-constrained WSNs. A typical receiver-receiver scheme is represented in Figure 2.

Figure 2

Clock synchronization model for a receiver-receiver protocol



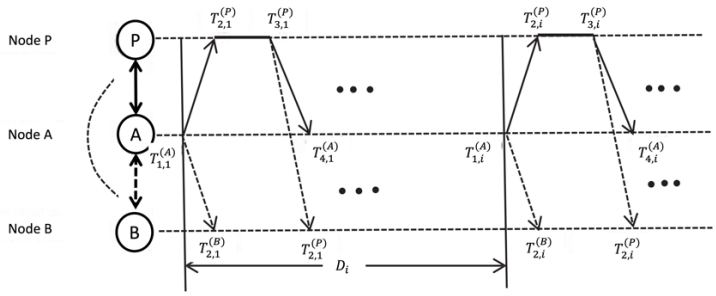
In this model, some knowledge on the delays structure and distance between nodes is needed, which can be evaluated using the corresponding responses (not represented in Figure 2). Note as well that, in the receiver-receiver model, node A acts as an anchor node, and as such its clock provides the reference to the network, normally obtained through GPS. Lee and Chin (2016) presented a receiver-receiver protocol scheme with which they obtained very accurate estimations. To evaluate the delay, they used two synchronization messages which, due to the model constrains, should not have a time gap between them of less than 40 msec. This time gap, represented by ξ_i in their model, is constant, and should be known beforehand. Moreover, the time lapse should not constitute a problem, since it is linked to the clock frequency and the desired clock skew resolution: Using higher frequencies would lead to shorter time lapses. In any case, their results, of the order of 10^{-5} minimum squared error (MSE) of $\hat{y}$, are exceptional. Our model showed similar results. However, they came at the cost of increasing the number of

observations –see Figure 11. Finally, Lee and Chin (2016) analyzed the results using an exponential distribution for the random portion of delays, with a mean of 3.33 μsec. The clock jitter is modeled using a normal distribution with a standard deviation of $\pm 60\ psec$, compared to our much more conservative value of 0.2 for the disturbances' $\{v_i\}$ standard deviation, the same value as in Chatterjee and Venkateswaran (2015), and 0.25 for the first value of the residuals, ω_1 of the same order as the clock skew in our model. Noise effects are captured in the scheme below through the *d* and ψ parameters, see for instance equation (25).

Receiver Only Synchronization: To the best of the author's knowledge, the ROS model was first introduced in a seminal work published in 2007 (Noh & Serpedin, 2007). In many aspects, ROS is a mixed model between the sender-receiver and receiver-receiver protocols introduced previously. The model aims at synchronizing nodes that only need to listen to the sender-receiver synchronization of a pair of two additional nodes. Note that, in this case, the use of a master node, although possible, is not necessary. The ROS protocol is presented below:

Figure 3

Receiver-only synchronization protocol model



Note. In this figure, node A is the anchor node, whereas node P represents a drone. By overhearing messages exchanged between nodes A and P, a second drone in node B can synchronize with drone P, and therefore with A, without the need of further broadcasts. Adapted from "An efficient statistical approach for time synchronization in wireless sensor networks", by A. Chatterjee and P. Venkateswaran, 2015, International Journal of Communication Systems, 29, p.724. (DOI: 10.1002/dac.2944).

ROS model also requires that nodes B and P be within reach of node A as well as within mutual reach, but the synchronization between A and P is direct in this case. In other words, if a third node can overhear a sender-receiver synchronization protocol between two nodes within its reach, it can synchronize with them without the need of more broadcasts (i.e., without the need to spend more energy). The ROS model is analyzed in further detail in the next chapter, and simulation results will be compared to Noh and Serpedin's (2007).

It must be noted that Noh and Serpedin's (2007) work consider that the random portions of timing delays are independent and identically gaussian-distributed, $\mathcal{N}\left(\mu, \frac{\sigma^2}{2}\right)$. This assumption produces a solution without the need of any iterative algorithm. Eight years later, Chatterjee and Venkateswaran (2015) took Noh and Serpedin's work one step further when they assumed that the difference between two successive delays values were first-order autocorrelated. Their solution required the use of some statistical tools to solve autocorrelated series, which they found in a work by Cochrane and Orcutt published in 1949 within the field of econometry. The present dissertation builds on Chatterjee and Venkateswaran's (2015) work.

Pairwise synchronization vs. network-wide synchronization

Serpedin and Chaudhari (2009) provide a fourth classification for synchronization protocol models. Whereas pairwise synchronization only deals with synchronization of two nodes, network-wide synchronization aims at establishing a common framework between all nodes in the system. Nature itself presents several examples of network-wide synchronization, such as in the case of the fireflies mentioned above (Buck, 1988).

After giving this brief discussion on synchronization protocols and explaining the reasons to pursue the analysis of an ROS protocol, we now turn our attention to the tools that will allow us to evaluate to what extent the ROS protocol fits our expectations in terms of accuracy and power consumption. The measurement tool used to compare the effects of different parameter values is the Mean Squared Error (MSE), of which a detailed introduction is provided in the next section.

Mean Squared Error and Maximum Likelihood Estimators

This section presents a description of the Mean Squared Error (MSE) estimator, as well as the conditions upon which it resolves as a Maximum Likelihood Estimation (MLE). The proofs provided can be found in many statistical sources. Some analysis of the last research works in the matter are also included.

Mean Squared Error

The MSE estimator has been widely used to analyze the accuracy of a certain algorithm with respect to the expected values. Let us imagine that we have a set of N estimated values of a given measurement, $\hat{y}_i$, with the subindex *i* ranging from 1 to N. These estimated values have been obtained through a certain algorithm, $\mathcal{A}$, which may take the form of an iterative process, such as is the case in the present study. Moreover, let us imagine that we know the exact values that correspond to each estimation, denoted by y_i. In this case, the MSE would provide a number which indicates how good the behavior of $\mathcal{A}$ is to estimate y_i. The MSE would take the following form:

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (1)$$

In other words, the MSE is the mean of the squared differences, or residuals, of $\mathcal{A}$ in trying to predict the exact values y_i. The lower the MSE value, the better the approximation of $\mathcal{A}$ is. The key is then to find upon which conditions the MSE value is lowest.

Let us now assume that the algorithm $\mathcal{A}$ works in the following manner: For every independent variable introduced, x_i, we obtain a result y_i which is linearly related with x_i, that is:

$$y_i = a + bx_i \quad (2)$$

It is further assumed that $\mathcal{A}$ is not perfect. Hence, the application of $\mathcal{A}$ introduces some errors in the system in the form of residuals $\{\varepsilon_i\}$. These residuals correspond to the noise which

originates in the transmission process (Kopetz and Schwabl 1989, as cited in Elson et al., 2002), as discussed towards the end of the preceding chapter. As a result, the estimations results are not the true values of y_i, hence are represented differently: $\hat{y}_i$. In this case, the linear regression of y_i over x_i, i.e. equation (2), must be corrected to:

$$\hat{y}_i = a + bx_i + \varepsilon_i \tag{3}$$

…where *a* is known as the intercept and *b* as the slope of the curve. This model is called the Simple Linear Regression Model (SLRM), since it is made of only two linearly related parameters: *a* and *b*. Carl Friedrich Gauss (1777-1855) was the first mathematician to propose a method for estimating these values (Snedecor & Cochran, 1967, p. 147). Regression analysis is a successful tool to analyze relationships between economic variables, and hence the description of the method presented below has been developed and can be found in several books in the statistics and econometric fields. Introducing equation (2) and equation (3) into equation (1) we obtain another expression for the MSE as (see, for instance, Snedecor & Cochran, 1967):

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(\varepsilon_i)^2 \tag{4}$$

Equation (4) is the mathematical expression of the concept of mean squared errors.

The replacement of (3) into (4) allows us to obtain an alternative expression:

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(a + bx_i - \hat{y}_i)^2 \tag{5}$$

The use of equation (5) to obtain the unknown parameters *a* and *b* is also known as the Ordinary Least Squares (OLS) method. In other words, OLS is a linear least squares method to find the estimates of $\hat{a}$ and $\hat{b}$ through equation (5). We can do this by differentiating MSE with respect to these parameters and set the resultant to 0. If we do this, we will obtain the following values (Pindyck & Rubinfeld, 1998, or Warner, 2013):

$$\hat{b} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2}, \text{ and} \quad (6)$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x}, \quad (7)$$

…where $\bar{y} = \frac{\sum y_i}{N}$ and $\bar{x} = \frac{\sum x_i}{N}$. The present work will use these results in the simulations performed in MATLAB®.

The intercept and slope of the curve in equation (2) correspond to the offset and skew concepts introduced towards the end of Chapter 1. As such, they have their origin in the random jitter of the synchronizing clocks –see chapter 1.

MSE as a Maximum Likelihood Estimation

In order to obtain an estimation of the parameters $\hat{a}$ and $\hat{b}$, we have implicitly introduced two assumptions that are important to highlight now.

We know that $\hat{a}$ and $\hat{b}$ values given by equations (7) and (6) respectively minimize the MSE expression given in equation (1). Moreover, it can be demonstrated (Snedecor & Cochran, 1967, p. 506) that, under certain circumstances, the MSE method corresponds to the maximum likelihood estimation (MLE), and therefore gives the lowest value possible. These conditions are

included in the Gauss-Markov theorem (Thejll & Schmith, 2005), and they basically mean that the error terms, or residuals $\{\varepsilon_i\}$ must have no structure. For the error terms to have no structure: a) their distribution in equation (3) must have zero mean– some authors add a stronger condition stating that the distribution must also be normal, but there is no consensus on this requirement– and b) each element is independent and identically distributed, sometimes abbreviated i.i.d. This last requirement is equivalent to say that: a) the population is made of a sufficiently large number of elements, and b) the samples obtained do not depend on the order they are extracted from that population, or alternatively the residuals are independent from x_i, $E\left[\varepsilon_i, x_j\right] = 0 \; \forall i, j$. In the case of i.i.d., we can ensure that, for a sufficient number of elements N, estimates $\hat{a}$ and $\hat{b}$ converge to the true values. In mathematical terms: $E[\hat{a}] = a$ and $E\left[\hat{b}\right] = b$. In this situation, we say that the statistics are unbiased. On the contrary, if the error distribution $\{\varepsilon_i\}$ do not comply with these two conditions simultaneously, the results are biased, i.e. $E[\hat{a}] = \tilde{a}$ and $E\left[\hat{b}\right] = \tilde{b}$ with $|\tilde{a} - a| > \delta_1$ and $\left|\tilde{b} - b\right| > \delta_2$ for certain δ_1 and δ_2 finites.

In summary, under the two conditions on the residuals $\{\varepsilon_i\}$ exposed above, the results of the estimators introduced by equations (6) and (7) are unbiased. As a corollary, we can state that they are also *efficient*, since their variance is smaller than that of any other unbiased estimator (Pindyck & Rubinfeld, 1998). The present dissertation will demonstrate this last point using the Cramér-Rao Bound (CRB) –see below.

For those cases where the residuals are autocorrelated, that is, $\varepsilon_i = f(\varepsilon_{i-1})$, we cannot ensure that the estimators represented by equations (6) and (7) are efficient. In fact, we cannot even ensure they are unbiased (Pindyck & Rubinfeld, 1998). Cochrane and Orcutt (1949)

proposed a smart solution to convert autocorrelated systems into an equivalent system with uncorrelated error terms, which is presented in the next section.

Linear autocorrelation in the residuals

Durbin and Watson (1950 and 1951) worked on a test statistic that helped identify correlation behavior in a given series of numbers, which in our model will be the residuals previously introduced, ε_t. Their statistic indicates the residuals' degree of correlation:

$$d = \frac{\sum_{t=2}^{T}(\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=2}^{T} \varepsilon_t{}^2} \tag{8}$$

The *d* statistic roughly corresponds to $2(1-\hat{\rho})$, and therefore ranges between 0 and 4, with a 2 value indicating no correlation. That is, for *d* sufficiently away from 2, an expression with the following structure could be used to obtain the successive values of ε_t:

$$\varepsilon_i = \hat{\rho}\varepsilon_{i-1} + \nu_i \tag{9}$$

In equation (9), ν_i is the disturbance term, which corresponds to white noise, i.e., $\nu_i \sim \mathcal{N}(0, \sigma^2)$. Note also that Durban-Watson statistic *d* is independent of $\hat{\rho}$ and σ.

As indicated earlier, the existence of autocorrelated residuals does not allow us to use the Ordinary Least Squares (OLS) method as an MLE for the problem of obtaining the intercept and slope in equation (2). To solve this issue, Cochrane and Orcutt (1949) presented a clean and smart solution that we will use in our model. Let us retake equation (2), for which we know that the residuals are autocorrelated: that is, they follow equation (9). We then subtract $\hat{\rho}\hat{y}_{i-1} = \hat{\rho}\hat{a} + \hat{\rho}\hat{b}\hat{x}_{i-1} + \hat{\rho}\hat{\varepsilon}_{i-1}$ to equation (2) to obtain the following expression:

$$\hat{y}_i - \hat{\rho}\hat{y}_{i-1} = \hat{a} - \hat{\rho}\hat{a} + \hat{b}(\hat{x}_i - \hat{\rho}\hat{x}_{i-1}) + \varepsilon_i - \hat{\rho}\hat{\varepsilon}_{i-1} \quad (10)$$

We then make the following change of variables:

$$\hat{y}_i^* = \hat{y}_i - \hat{\rho}\hat{y}_{i-1} \quad (11)$$

$$\hat{a}^* = \hat{a} - \hat{\rho}\hat{a} \quad (12)$$

$$\hat{x}_i^* = \hat{x}_i - \hat{\rho}\hat{x}_{i-1} \quad (13)$$

$$\nu_i = \varepsilon_i - \hat{\rho}\hat{\varepsilon}_{i-1} \quad (14)$$

Note that equation (14) is equation (9) put in a different order.

Introducing these new variables into equation (10), a new expression results:

$$\hat{y}_i^* = \hat{a}^* + \hat{b}\hat{x}_i^* + \nu_i \quad (15)$$

Equation (15) has the same structure as equation (3), but in this case the deviations, called disturbances here, are distributed with zero mean, $E[\{\nu_i\}] = 0$. This would allow us to consider MSE as the MLE and apply it to obtain $\hat{b}$ and the new variable $\hat{a}^*$ using equations (6) and (7). Obviously, we should then undo the change to obtain $\hat{a}$ from $\hat{a}^*$, but in order to do this we would need an estimate of $\hat{\rho}$ first. Cochrane and Orcutt (1949) also dealt with this issue of estimating several parameters simultaneously, and proposed an iterative process that is presented here:

1. Use equation (3) to obtain $\hat{a}$, $\hat{b}$ and $\hat{\varepsilon}_i$.
2. Introduce the values of $\hat{\varepsilon}_i$ obtained previously into equation (9) to get $\hat{\rho}$.

3. Use $\hat{\rho}$ to make the appropriate transformations, from $\hat{y}_i$, $\hat{x}_i$ and $\hat{a}$ into $\hat{y}_i^*$, $\hat{x}_i^*$ and $\hat{a}^*$, respectively, using equations (11), (12) and (13).
4. Introduce the new variables into equation (15) and obtain an estimate for the disturbances, $\hat{v}_i$.
5. Feed step 2 with $\hat{v}_i$ values and get a new value of the correlation coefficient $\hat{\rho}_{new}$.
6. Repeat steps 2 through 5 until the differences between successive estimations of $\hat{a}$ and $\hat{b}$ are "sufficiently small".

Some authors have highlighted the fact that Cochrane and Orcutt's (1949) iterative process might lead to local, instead of global, minima (Oxley & Roberts, 1982). These authors mention other colleagues' works that proposed the use of a grid to find $\hat{\rho}$ minima (Hildreth & Lu, 1969, as cited in Oxley & Roberts, 1982). However, in this case the processing time might get significantly longer. The present author acknowledges the inconvenience of potential local minima but, for the sake of simplicity, will not pursue the grid approach.

Cochrane and Orcutt's (1949) scheme also presents an issue with the initial values, since the new variables $\{y_i^*\}$ and $\{x_i^*\}$ are not defined for i=1. They solve it by making the first term of the series equal to 0. However, other works have proposed different initial values. Dielman (1985), and Pyndick and Rubinfeld (1989), for instance, propose the following alternative:

Equation (9) can be used to obtain the variance of the first term in the series, i.e., ε_1:

$$var(\varepsilon_1) = \frac{\sigma^2}{1-\rho^2} \tag{16}$$

As equation (16) shows, the variance of the first term of the residual may be very big in cases of high correlation, $|\rho| \approx 1$. A useful transformation for this first term would be (Pindyck & Rubinfeld, 1989):

$$\varepsilon_1^* = \sqrt{1-\rho^2}\varepsilon_1 \tag{17}$$

One way to get to the expression in equation (17) is to use the following transformation for these first terms:

$$y_1^* = \sqrt{1-\rho^2}y_1 \tag{18}$$

$$x_1^* = \sqrt{1-\rho^2}x_1 \tag{19}$$

The expression for these initial values will also be adopted in the present work. However, this adoption must be handled with care: If, during any of the iterations, the estimated value $|\hat{\rho}| > 1$, $\hat{y}_1^*$ and $\hat{x}_1^*$ obtained are complex values, and the process becomes unstable. In order to avoid that, a condition was introduced in the recursive scheme whereby whenever $|\hat{\rho}| > 1$, it was replaced by a middle value between extremes $\hat{\rho} = 0.5$, since $0 < \hat{\rho} < 1$.

With an estimation of the parameters $\hat{a}$ and $\hat{b}$, we can obtain their variance as the number of the sample elements taken increases from 2 to *N* as:

$$var\ (\hat{x}) = \frac{\sum(x_i - mean(x))^2}{n-1} \tag{20}$$

…with $\hat{x}$ replacing either $\hat{a}$ or $\hat{b}$. The next step is to evaluate whether the parameters' estimations comply with the Cramér-Rao bound, which is a property of all MLEs. The next section introduces this concept.

Cramér-Rao Bound

The Cramér-Rao Bound (CRB) constitutes a lower minimum for the variance of an unbiased estimator. In his work, Cramér tried to answer two related questions: "How should we best use the data to form estimates?" and "What do we mean by the "best" estimates?" (Cramér, 1999, p. 473). He answers the second question by stating that the "best" estimates are those that do not fall "too far" from a given value, considered to be the real value in the case of unbiased estimators. In other words, the estimates are concentrated around the real value. The minimum variance could in principle be used as a good statistic to evaluate this concentration. Cramér (1999) demonstrates that this minimum variance can never fall below a specific limit, which came to be known as the Cramér-Rao Bound (CRB). What follows is a summary of how to obtain the CRB in the case of using minimum square errors as a maximum likelihood estimator. The information has been extracted from Cramér (1999), and Serpedin and Chaudhari (2009).

Cramér-Rao approach start with the biparametric function $y(x;\alpha)$ of a variable $\{x_i\}$ in the model described by equation (2). The function *y* is determined by two unknown parameters, *a* –the intercept– and *b* –the slope–, with unbiased estimators $\hat{a}$ and $\hat{b}$, respectively. Cramér (1999) demonstrates that the variance of an estimator α (here α replaces either *a* or *b*) complies with the following inequality:

$$E(\hat{\alpha}-\alpha)^2 \geq \frac{1}{n\int_{-\infty}^{\infty}\left(\frac{\partial \log y}{\partial \alpha}\right)^2 y(x;\alpha)dx} \tag{21}$$

In our case, the equation (above) is resolved to give the following results for each parameter (adapted from Serpedin and Chaudhari, 2009):

$$var(\hat{a}) \geq \frac{\sigma^2 \sum_{i=1}^{N} x_i^2}{N\sum_{i=1}^{N} x_i^2 - \left[\sum_{i=1}^{N} x_i\right]^2} \tag{22}$$

$$var(\hat{b}) \geq \frac{\sigma^2 N}{N\sum_{i=1}^{N} x_i^2 - \left[\sum_{i=1}^{N} x_i\right]^2} \tag{23}$$

In equation (22) and equation (23), σ is the standard deviation of the disturbances, $v_i \sim \mathcal{N}(0,\sigma^2)$. Thus, the values of the variance obtained for each of the parameters will be compared with the Cramér-Rao lower bound. Afterwards, the simulation results with each of the parameters' values will be sorted out by their distance from the CRB.

In summary, CRB can be used to evaluate how spread the results of an estimation are, compared to their theoretical minimum. For instance, if we obtain a set of estimations which are accurate –the mean is the true value—but their variation is too large, the chances of getting meaningful results decrease in small sample sizes. In this sense, CRB have been traditionally used in the literature to compare different statistics. However, it is important to remember that CRB provides the lower variation bound as long as these statistics are unbiased. A detailed presentation of the Cramér-Rao bound (CRB) can be found in two independent papers, one introduced by C. R. Rao (1945), and the other one year later by H. Cramér (1999, original work published 1946).

The Origin of Time Delays

This section provides a description of the time delays introduced in the transmission process $\{\varepsilon_i\}$ and proposes some statistical models that have been traditionally used to represent them.

It is common to think of signal delays as composed of a fixed and a random portion. Random delays, due to their very nature, have a statistical distribution through time. Many works take a gaussian distribution to represent random delays, but the justification behind these decisions is often missing. It is true that some statistic models have certain features that make them more appropriate to represent specific delays: The probability density function of an exponential distribution, for instance, does not take negative values, hence it is not appropriate to model phenomena that may advance the arrival time of a signal –negative values. Following Luo et al. (2016), the present work will analyze the efficiency over the simulation results of four statistics: gaussian, exponential, gamma, and Weibull distributions. Delays' analyses of a stochastic nature are not new: Works exist in a variety of fields, such as epidemics (Caraballo et al., 2020), commensalism (Deng, 2019), transport models (Chen & Cassandras, 2018), and communication signals. The last set constitute the object of our interest.

Residuals $\{\varepsilon_i\}$ in equation (9) represent noise that alter the time of arrival of signals to the receiver. Even signals broadcast every, say, 1 msec, does not mean that they reach the receiver at this same interval. Noise effects may alter their period at different stages –see the OSI model at the end of chapter 1. Several statistics have been used in the literature to model this variability. The most typical of them is the gaussian distribution. Of all the models considered here, it is the only one that can have positive and negative values, and hence susceptible of having a zero

mean. The interest of gaussian distributions lies in the fact that the average value of samples of an unknown population becomes gaussian as long as we use a sufficient number of elements N to calculate this average. This is the idea stated in the central limit theorem: "if [a] random variable X has mean μ and variance σ^2, then the sampling distribution of [the mean] $\bar{X}$ becomes approximately normal with mean μ and variance σ^2/N as N increases" (Pindyck & Rubinfeld, 1998, p. 28). Note that, for small samples, where the application of the central limit theorem results in wide dispersion of results, we cannot estimate the population mean assuming a normal distribution, and we must consider alternative models. Also, gaussian distributions may be best used when no assumptions are made on noise sources. However, if delays are the major component of noise generated, gaussian distributions might not be our best choice. Delays always result in additional time, and therefore the residuals $\{\varepsilon_i\}$ in equation (9) are strictly positive. Exponential, gamma and Weibull distributions can only take positive values; hence they would fit these cases better.

Exponentially distributed noise is the second most common type found in the literature. Chen et al. (2020) obtained good correlation of delays when using an exponential model in Beyond Line of Sight (BVLOS) scenarios for GPS L1 coarse acquisition (CA) signals. This could provide some grounds to believe that signals transmitted between two nodes when there is not direct line of sight result in exponentially distributed noise.

To the best of the author's knowledge, models based on gamma distribution of delays applied to communication signals are scarce. Kim et al. (2009) is one of them. Gamma distribution of delays have good correlation in multipath environments (Chen et al., 2020).

Finally, studies specifically based on the Weibull distribution, although sparser still, also exist: Ahmad et al. (2010) work is based on a two-way synchronization protocol, which is different than the protocol proposed here. Thus, comparison of their results with this dissertation's must be done with care. Ahmad et al. (2010) used a Weibull distribution with a shape parameter $\beta=1$, and showed that MSE results are lower than if the exponential distribution was considered. Weibull distributions might be good models for noise generated from a large set of scatterers distributed randomly in space (Yacoub, 2007).

This chapter analyzed the existing literature on time synchronization protocols. Following Serpedin and Chaudhari's (2009) classification, we presented some of the most recurrent protocol types that were developed within the last years. Of them, the ROS protocol has been selected for a deeper analysis in the form of Matlab® simulations. To this we dedicate the next chapter. When studying the efficiency of a ROS protocol, some statistical tools are necessary, and therefore an introduction to them was performed in the third section. This chapter ended with an exploration of the main causes of clock delays in wireless communications and provided brief notes on the treatment the existing literature gives to the last ones.

III. THE PROOF OF CONCEPT

This chapter presents the proof of concept for a time synchronization protocol which meets the needs of UAS operating in BVLOS. The proof focuses on the accuracy achieved on the estimations of key synchronization parameters (i.e., clock offset and clock skew). Tests were performed, and their results and corresponding discussion are presented in ensuing Chapter IV for the characterization of the proposed model, and Chapter V, which is dedicated to the application of a duty cycle. This chapter is divided into the following three sections:

- The first section settles definitions of concepts that have already been used and will continue to apply to the rest of the dissertation.
- The second section presents a detailed analysis of the ROS protocol. It includes the characterization of an algorithm used in the simulation.
- The third step describes the MATLAB® simulations that were subsequently performed. These simulations have been divided into two phases:
    - Characterization of the proof of concept for the protocol, and
    - Variability of the results when a duty cycle scheme is introduced.

Definition of Terms

This author considers that a set of key definitions is of necessity, particularly since the use of some of these terms is not consistent across the existing literature. Some of the concepts included in the list have been introduced in the previous two chapters. Appropriate references for the definitions are provided when necessary.

Time synchronization: “Time synchronization is a procedure for providing a common notion of time across a distributed system” (Serpedin and Chaudhari, 2009, p. 3).

Time synchronization protocol: A pre-defined set of timestamped messages exchanged between two sources in order to achieve time synchronization. Each protocol is uniquely defined by a specific structure, which compresses the order of the messages and the number and rank of participants in the messages exchange. Sundarman et al. (2005) summarize the requirements for clock synchronization protocols in the following four aspects:

- “The protocol should cope with unreliable network transmission and unbounded messages latencies.
- When synchronizing two nodes, each node must be able to estimate the local time on the other node’s clock. […]
- Time must never run backward. […]
- Synchronization overhead must not degrade [the] system performance.” (p.285).

Clock: “A device that can provide a measurement of the passage of time since a [defined origin of the timescale]”. (Adapted from Institute of Electrical and Electronic Engineers, 2019, p. 18).

Clock drift: The phenomenon by which two clocks that share the same origin of the timescale do not share the same reading at a later instant. In other words, clock drift is manifested by differing mismatches between clock readings through time. Some sources define clock drift as a derivative of second order of the clock offset over time (Rhee et al., 2009; Yiğitler et al., 2020). However, this last definition will not be used in the ensuing discussion.

Clock offset: The difference between the readings of two clocks at any given instant. Sundaraman et al. (2005) provide a more restrictive definition, whey they state that a clock offset is "[…] the difference between the time reported by a clock and the real time" (Sundaraman et al., 2005, p. 285). Clock offset may be measured in seconds.

Clock skew: The rate at which the readings of two clocks drift apart. Mathematically, it can also be defined as the ratio of the clock offset and the time to reach this difference. For instance, a processor with a clock skew of 10^{-6} would mean that there is an offset with respect to a reference framework of one additional second every 11.5 days approximately. Note that clock skew may vary with time. Again, Sundaraman et al. (2005) restrict the definition to "the difference in the frequencies of the clock and the perfect clock" (Sundarman et al., 2005, p.285). Clock skew units may be represented by [increasing drift] sec /sec [absolute time].

Duty cycle: For our purposes, duty cycle is a combination of one active mode and one successive idle mode. This dissertation assumes that duty cycles repeat indefinitely.

Active mode: A number of successive loops during which the local clock is actively overhearing a broadcast exchange between node P and node B. Also known as wake mode.

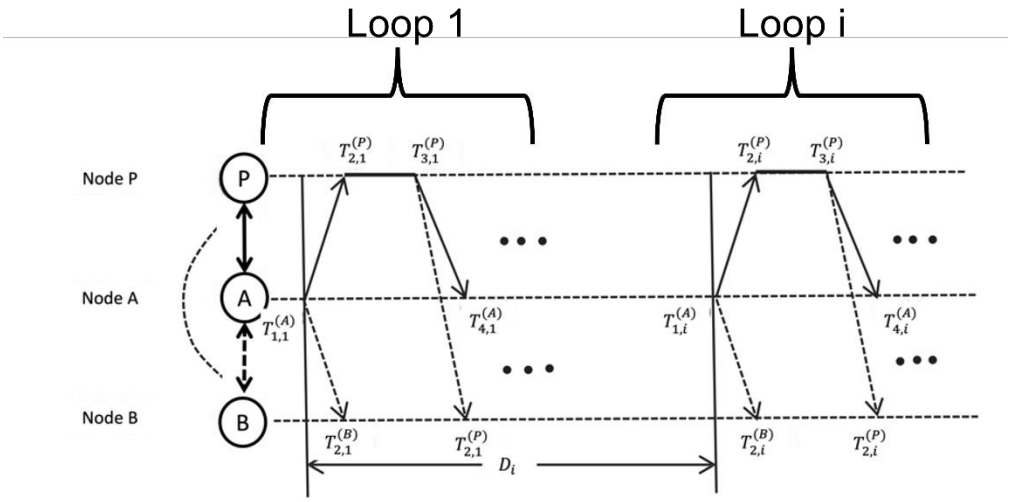
Idle mode: A number of successive loops during which the platform turns off its sensor and the local clock extrapolates the information obtained from the last active mode to estimate its offset and skew. Also known as sleep mode.

Loop: A number of exchanged messages with a pre-defined pattern. Active modes are made of a set of loops. A loop is made of the set of messages necessary to update estimates on

clock offsets and skew. Note that, in the ROS model, several loops are necessary to determine a clock drift.

Figure 4

Loops in the ROS model



Note. In the ROS model, a loop is composed of two messages: one from A to P, and the other from P to A. Note that, although B receives these messages as well, it is not the main addressee of A-messages. Adapted from "An efficient statistical approach for time synchronization in wireless sensor networks", by A. Chatterjee and P. Venkateswaran, 2015, International Journal of Communication Systems, 29, p.724. (DOI: 10.1002/dac.2944).

Iterations: Two types of iterations are performed in the model. This dissertation differentiates between first-level and second-level iterations:

First-level iterations: Number of program's runs over which to average. A minimum number of runs is necessary to smooth out the results obtained, making them independent of initial random variable and thus guarantee the validity of the conclusions reached.

Second-level iterations: Number of iterations of the algorithm that are necessary to obtain two successive estimations differing in less than a given value. These estimations are made on the clock offset and clock skew. The iterative model used converges within a few number of iterations.

The following statistical terms are closely related, and yet different. For the sake of clarity, a distinction needs to be made between them.

Errors: The difference between an estimated value and a reference. Error is a generic term, which can be either refer to outliers, residuals or disturbances.

Outliers: For our purposes, an outlier in linear regression analysis is the distance of the real value to the value of the model represented by a curve (a straight line).

Residuals: Residuals may refer to the error terms in the linear regression model, represented by ε_i in equation (3). In our model, these error terms are equivalent to the difference of random delays in the ROS scheme, $\omega_i = \mathrm{X}_i^{(AP)} - \mathrm{X}_i^{(AB)}$. Therefore, although the same deduction to obtain ε_i is valid for ω_i, the last symbol will be used henceforth, in line with existing works in the field.

Here, as for Chatterjee and Venkateswaran (2015), residuals present a linear autocorrelation of the first order, i.e. $\omega_i = \rho\omega_{i-1} + \nu_i$.

Disturbance: Disturbances are the error terms of the residuals in the autocorrelation model. They correspond to the ν_i terms in equation (9).

The ROS Protocol

This section introduces the ROS protocol used. The discussion relies heavily on statistical concepts introduced in the preceding chapter, such as the Minimum Squared Error (MSE).

Chapter II correlated any imperfections in the crystal quartz at the core of a clock or even certain disturbances in the transmission channel results on a clock behavior diverging from a regular pattern to which it was initially adjusted. The simplest mathematical model, and the most commonly applied, provides a linear relationship between the time as seen from a particular clock τ_j, and an absolute time t, identified with the UTC or any other standard time used as a reference in a local framework:

$$\tau_j(t) := a_j t + b_j \tag{24}$$

Equation (24) may be found in multiple sources, such as in Freris et al. (2011). In their paper, these authors define a_j as the ratio at which two clocks speed with respect to each other, whereas b_j is the difference in their displays at a given initial time *t=0*. In accordance with the definitions provided above, they correspond to the clock skew and offset concept, respectively.

At this point, it is important to expand a bit on the relative nature of equation (24): Due to its linear behavior, the structure of this equation is valid irrespective of whether the source to measure clock drift is an absolute clock like UTC or any other clock in the network: Only the value of the offset and skew parameters would change. In the first case, the difference in time between clock *j* and UTC is represented by b_j, whereas the number of ticks that clock *j* makes for each UTC tick is represented by a_j. In the second case, b_j would represent the difference between clock *j* and UTC offset on one side, and the offset between the local reference clock and UTC on

the other side, i.e., the UTC reference disappears. Moreover, a change of relative frames is also immediate. Hence, absolute and relative synchronization (Freris et al., 2011) are equivalent terms. Consequently, the existence of absolute clocks in the network is not essential for the process. Another way to see this is from a relativistic perspective, where references established are always conventional. Put simply: Any two clocks chosen at random may set up a common frame valid between them.

A protocol has been previously defined as a pre-defined set of timestamped messages exchanged between two sources. This exchange is useful in that it helps determine the local clock offset and skew, and thereby achieves synchronization with other clocks in the network. Today, an interest exists in developing increasingly more efficient, accurate and/or simple synchronization protocols for many applications. As a result, protocols models continue piling up, and it is easy to lose sight of the forest for the trees. Chapter 2 of this dissertation makes a succinct exploration of some of these models. The protocol presented here belongs to the Receiver Only Synchronization (ROS) family, and it can be found in Chatterjee and Venkateswaran (2015). The main reasons to choose a ROS protocol over any other scheme are: First, they work with no additional messages exchanged, thus reducing the probabilities of interference in the channel, particularly in multi-agent or swarm UAS compositions transmitting simultaneously. Second, it is capable of minimizing energy consumption, which is important since UAS are typically powered by relatively small capacity batteries due to aerodynamic limit that preclude carrying larger ones.

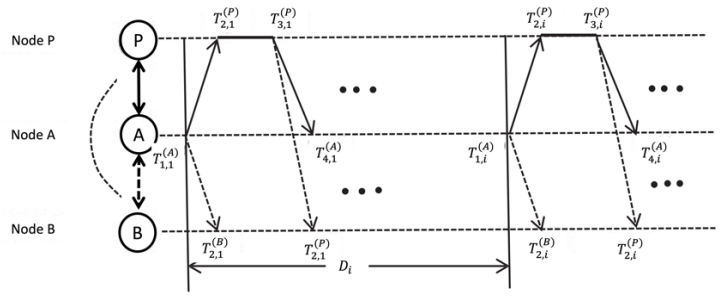
A ROS model is composed of three nodes (see Figure 5):

- *Node A*, representing an anchor node, normally a ground station which transmits timestamped messages at regular intervals. The anchor node is assumed to be synchronized with GPS to provide a general time reference. Else, node A may also provide a local reference.
- *Node P*, representing a UAS platform that exchanges messages with node A, following any of the sender-receiver synchronization schemes present in the literature.
- *Node B*, representing a UAS platform which listens to these messages and synchronizes with node P, and therefore with A, following a ROS synchronization scheme.

ROS protocols allow for synchronization between nodes B and P with no message exchange between them, which helps saving energy resources. As noted in the introduction, these resources consumption are essential in a UAS-based Wireless Sensor Networks (WSN). Although this protocol was schematically represented in Figure 4, it is repeated below as a reminder:

Figure 5

Receiver Only Synchronization (ROS) Protocol



Note. In this figure, node A is the anchor node, whereas node P communicates with Node A. By listening to nodes A-P exchange, node B can synchronize with node P without sending a single message. From "An efficient statistical approach for time synchronization in wireless sensor networks", by A. Chatterjee and P. Venkateswaran, 2015, International Journal of Communication Systems, 29, p.724. (DOI: 10.1002/dac.2944).

In Figure 5, we define one loop as any of the following sequence of observations, repeated in time: Node A sends a timestamped message $T_{1,1}^{(A)}$ to node P, which receives and timestamps it in turn according to its local clock, producing $T_{2,1}^{(P)}$. Also, $T_{1,1}^{(A)}$ reaches node B, which timestamps it as $T_{2,1}^{(B)}$. However, neither of these three times will be the same, due to inherent delays in the process –see the distribution of delays in the OSI model summarizing Zimmermmann (1980) above–. In a second phase, node P timestamps the response message at $T_{3,1}^{(P)}$, which also includes $T_{2,1}^{(P)}$. Node A receives this response and timestamps it at $T_{4,1}^{(A)}$. Finally,

node B receives the response from P to A and extracts $T_{2,1}^{(B)}$. The loop is thus complete, and another loop begins after a pre-specified time with a new message sent from node A. Let us for now concentrate on $T_{1,1}^{(A)}$ and $T_{2,1}^{(P)}$, i.e., the time when the first message was produced/left node A as seen by A, and the time it reached node P, as seen by P. In Figure 5, *i* represents the i-th loop, at the end of which B has collected $2i$ timestamps. We represent both sets as $\left\{T_{1,i}^{(A)}\right\}$ and $\left\{T_{2,i}^{(A)}\right\}$.

To the formula which relates the timestamps between nodes P and A –see equation (24) – random delays must be added to the transmission. These delays, of varied natures as indicated in preceding chapters, are commonly classified into random (X) and fixed delays (d). A significant contribution to the fixed delays d is due to the finite speed at which wireless signals travel across a given medium. For a given loop *i* (see Figure 5), equation (24) can be rewritten as:

$$T_{2,i}^{(P)} = T_{1,i}^{(A)} + \theta_{offset}^{(AP)} + \theta_{skew}^{(AP)}\left(T_{1,i}^{(A)} - T_{1,1}^{(A)}\right) + d^{(AP)} + X_i^{(AP)} \quad (25)$$

To reach the expression in equation (25), we have replaced a_j and $\mathrm{b_j}$ in equation (24) by $\theta_{skew}^{(AP)}$ and $\theta_{offset}^{(AP)}$, respectively, since this notation serves as a reminder of the offset and skew values defined between nodes A and P. It is also worth observing that fixed delays $d^{(AP)}$ do not carry any subindex, since they are assumed to be constant along the loops. Last, the relative time *t* is defined as the difference shown in the timestamps between the signal reaching P in the i-th loop and the signal reaching P in the first, i.e., $T_{1,i}^{(A)} - T_{1,1}^{(A)}$.

Conversely, replacing P by B in equation (25) would allow us to obtain the same formula for the clock in node B. We then subtract this equation from equation (25) to get:

$$T_{2,i}^{(P)} - T_{2,i}^{(B)} = \theta_{offset}^{(BP)} + \theta_{skew}^{(BP)} \left(T_{1,i}^{(A)} - T_{1,1}^{(A)}\right) + d^{(AP)} - d^{(AB)} + X_i^{(AP)} - X_i^{(AB)} \qquad (26)$$

Note that, using the property already mentioned on time relativity, $\theta_{\text{offset}}^{(\text{AP})} - \theta_{\text{offset}}^{(\text{AB})}$ and $\theta_{\text{skew}}^{(\text{AP})} - \theta_{\text{skew}}^{(\text{AB})}$ have been replaced by $\theta_{offset}^{(BP)}$ and $\theta_{skew}^{(BP)}$, respectively. Equation (26) has thus the same structure as equation (25). Moreover, since P transmits $T_{2,i}^{(P)}$, if the random delays were known distributions, node B could evaluate both its clock offset and skew immediately. Unfortunately, this is an assumption too strong to be made, and we must further elaborate on equation (26) to obtain a more practical expression.

First, with respect to the fixed portion of delays $\text{d}^{(\text{AP})} - \text{d}^{(AB)}$, it is usual to integrate them into the left side of the equation, and define a new variable, y_i, given by $y_i = T_{2,i}^{(P)} - T_{2,i}^{(B)} - \left(\text{d}^{(\text{AP})} - \text{d}^{(AB)}\right)$. Whereas in a sender-receiver synchronization approach it is commonly assumed that the fixed delays cancel each other out, it is not possible to extend this hypothesis to the present case, i.e., in general $\text{d}^{(\text{AP})} - \text{d}^{(AB)} \neq 0$. This is mainly because the distance between nodes A-P may be different than the distance between nodes A-B. Further calling the residuals $\omega_i = \text{X}_i^{(AP)} - \text{X}_i^{(AB)}$, a simplified equation results in:

$$y_i = \theta_{offset}^{(BP)} + \theta_{skew}^{(BP)} \left(T_{1,i}^{(A)} - T_{1,1}^{(A)}\right) + \omega_i \qquad (27)$$

Equation (27) is linear in the $T_{1,i}^{(A)} - T_{1,1}^{(A)}$ parameters. Therefore, the application of MSE to obtain $\theta_{offset}^{(BP)}$ and $\theta_{skew}^{(BP)}$ using equations (6) and (7) would be feasible, as long as the residuals ω_i are independent and identically distributed (i.i.d.), as we saw in the preceding chapter. Our assumptions, on the contrary, is that residuals may follow a certain autoregressive delay model,

which for the sake of simplicity we assume to be linear and of the first order. The next section presents the algorithm to solve for $\hat{\theta}_{offset}^{(BP)}$ and $\hat{\theta}_{skew}^{(BP)}$.

The Algorithm

Mathematically, the simplest autoregressive model of the first order to represent residuals is:

$$w_i = \rho\omega_{i-1} + \nu_i \tag{28}$$

Equation (28) is an example of what is known as a Markov scheme (Cochrane & Orcutt, 1949). Markov schemes are those sequences where the state of one event depends on the state attained in the previous event. In our case, the value of the residual at time *i* depends on the value of the residual at time *i-1*. We will assume with Chatterjee and Venkateswaran (2015) that the disturbance term ν_i is an i.i.d., gaussian variable of the type $\mathcal{N}(0, \tau^2)$, and the standard deviation τ is unknown. As shown later, the noise level was established at -3dB with respect to the signal $T_{1,i}^{(A)}$. Since ω_i variable depends on the value of ω_{i-1}, the least squares method is not valid to obtain the linear regression parameters in equation (27) (Cochrane & Orcutt, 1949). Our objective is to evaluate this equation, which can also be represented as:

$$y_i = \theta_{offset}^{(BP)} + \theta_{skew}^{(BP)} * x_i + \omega_i \tag{29}$$

…replacing $T_{1,i}^{(A)} - T_{1,1}^{(A)}$ by x_i. We then apply the same change of variables presented in equations (11), (12) and (13) to obtain an equation equivalent to (15):

$$y_i^* = \theta_{offset}^* + \theta_{skew}^{(BP)} \left[T_{1,i}^{(A*)} - T_{1,1}^{(A*)} \right] + \nu_i \quad (30)$$

This is a linear regression model that can be solved using MSE. However, because the estimate we need to obtain is $\theta_{offset}^{(BP)}$ rather than θ_{offset}^*, and because θ_{offset}^* is also dependent on the regression parameter ρ, which is another unknown, we need a recursive algorithm to solve equation (30). This algorithm was presented in Chapter II. Adapted to the ROS scheme, it resolves following five steps:

1. Estimate an initial value of the autoregressive coefficient $\hat{\rho}$ using equation (28). This estimation is made of three steps:
    a. Estimate the linear regression coefficients $\hat{\theta}_{offset}$, $\hat{\theta}_{skew}$ by ordinary least squares using equations (6) and (7).
    b. Obtain the resulting series of residuals $\hat{\omega}_i$.
    c. Using these residuals $\hat{\omega}_i$ and least squares methods, estimate the autoregressive parameter $\hat{\rho}$ using the following formula, adapted from Warner (2013):

$$\hat{\rho} = \frac{E[(\omega_i - \bar{\omega})(\omega_{i-1} - \bar{\omega}]}{E[(\omega_i - \bar{\omega})^2]} \quad (31)$$

    An equivalent and easier way to obtain $\hat{\rho}$ is to use equation (28) with any of the *i* values. Alternatively, one could use this equation for all *i* values, and obtain the averaged $\hat{\rho}$, which is what was done in the second part of the tests.
2. Since $T_{x,y}$ and, therefore, x_i are known from the timestamped messages received, evaluate $y_i^*, T_{1,i}^{(A*)}$ and $T_{1,1}^{(A*)}$ using $\hat{\rho}$.

3. Introduce the estimated parameters calculated in steps 2 and 3 into equation (30) and obtain a second estimate of $\hat{\theta}^*_{offset}$ and $\hat{\theta}^*_{skew}$, together with new disturbances $\widehat{v_i}$.
4. Use these new disturbances $\widehat{v_i}$ to obtain a new estimation of $\hat{\rho}$. Neither Chatterjee and Venkateswaran's (2015) article nor Oxley and Roberts' (1982) give explicit details in this step. In this dissertation, four methods will be tested in the first part, and a best fifth option retained for the second part of the simulations.
5. Return to step 3 and repeat the steps until the values of $\hat{\theta}^*_{offset}$ or, alternatively, $\hat{\theta}^*_{skew}$ stabilize. In the second part of the simulations, the condition to end the iterative scheme will consist of having differences in two successive $\hat{\rho}$ below .0001.

These steps are summarized in the following table:

Table 3

Iterative Scheme to Obtain Clock Offset and Clock Skew

1. Initial $\hat{\rho}$ estimation.
2. Evaluate y_i^*, and $\left\{T_{1,i}^{(A)}\right\}$.
3. Use equation (30) to obtain $\hat{\theta}^*_{offset}$, $\hat{\theta}^*_{skew}$ and $\hat{v}_i$.
4. Use equation (24) to obtain a new $\hat{\rho}$.
5. Iterate from step 3.

The autoregressive structure presented in equation (28) is not the only residuals' scheme that was planned to be used at first. Cochrane and Orcutt (1949) proposed three additional schemes:

$$\omega_i = \nu_i \tag{32}$$

$$\omega_i = \nu_i - \nu_{i-1} \tag{33}$$

$$\omega_i = \omega_{i-1} + \rho(\omega_{i-1} - \omega_{i-2}) + \nu_i \tag{34}$$

Note however that neither equation (32) nor equation (33) constitute an autoregression model, since the values of the residuals at a given instant *i* $\{\omega_i\}$ do not depend on the values at any precedent instant *i-t*. In particular, the second case represents a normally distributed variable with zero mean and standard deviation $\sqrt{2}\tau$, i.e. $\mathcal{N}(0, \sqrt{2}\tau)$. Lastly, equation (34) does represent an autocorrelation model, but of a second instead of a first order. The behavior of the scheme presented in equation (**34**) was analyzed in the present dissertation. Another scheme analyzed was that corresponding to a second-order autoregression model with two autoregression parameters:

$$\omega_i = \rho_1\omega_{i-1} + \rho_2\omega_{i-2} \tag{35}$$

Initial Values

The problem of the initial values in an autocorrelated model such as that represented in equation (28) will be solved by using equations (17), (18) and (19). Note that, for these cases, we must have $|\rho| < 1$, that is, the ω_i values must be bounded, which guarantees a stationary distribution of ω_i (Prais & Winsten, 1954).

Transmission At Regular Intervals

An additional assumption is that node A transmits at regular intervals. In other words, the set $\{T_{1,i}^{A}\}$ is composed of timestamps represented by a chain of natural numbers $\{1, 2, 3, \dots\}$. This assumption is not difficult to accept, since even LTE synchronization schemes work with signals that repeat themselves at fixed intervals through time. It will be used to extrapolate the results obtained during active modes through idle ones.

The Model Values

The main values below will be used in the model proposed:

Table 4

Main Values for the Model

Parameter	Value
Number of first-level iterations (runs)	1,000
$\theta_{skew}^{(BP)}$ (true value, to be estimated)	0.97
$\theta_{offset}^{(BP)}$ (true value, to be estimated)	1.45 sec
Autocorrelation coefficient ρ (true value, to be estimated)	0.65
ω_1 (first residual)	$0.001 * \mathcal{N}(0,0.25)$
$\{\nu_i\}$ (disturbances)	$0.001 * \mathcal{N}(0,2)$

Note. A skew value of 10^{-6} is more typical in today's hardware (Rhee et al., 2009). Yiğitler et al. (2020) use a skew value one order of magnitude bigger, 10^{-5}. This dissertation uses the skew value proposed in Chatterjee and Venkateswaran (2015) of 0.97. The iterative nature of the

residuals' autoregression schemes such as those represented in equations (28) and (34) forces us to adopt a random value for the first term, ω_1.

The loops per duty cycle are divided into wake and sleep modes. Whereas the total number of loops is set to 400, the sleep/idle mode is composed of several loops of between 0 and 40, and the active mode is composed of between 100 and 60 loops. Thus, a duty cycle is composed of 100 cycles. After the idle cycle finishes, a new active cycle begins again, until the total number of loops (400) is completed:

Figure 6

Basic Cycle: Active and Idle Modes Distribution

With the information displayed in Table 4, the autoregressive model in equation (28) can be constructed to obtain the values $\{y_i\}$ through equation (29). This set of values $\{y_i\}$ correspond to real values in the model, as obtained by node B. Of course, node B does not know either $\theta_{skew}^{(BP)}$ or $\theta_{offset}^{(BP)}$, which are the parameters that our model estimates. In other words, what the timestamps reveal are the times at which each message left node A/P, but node B ignores what time was that "as seen by B". The result of the estimations, $\hat{\theta}_{skew}^{(BP)}$ and $\hat{\theta}_{offset}^{(BP)}$, will be compared to their true values, thus providing information of the accuracy of the model used, and the efficiency of the assumptions made in the model.

In every loop, node A broadcasts a new timestamped message $T_{1,i}^{A}$, which node B may or may not receive depending on whether it is in active or idle mode. If in active mode, node B incorporates the new message into a new estimation of $\hat{\theta}_{skew}^{(BP)}$ and $\hat{\theta}_{offset}^{(BP)}$. Conversely, if node B is in idle mode, it extrapolates the value of y_i to y_{i+1} using equation (29) and the estimations $\hat{\theta}_{skew}^{(BP)}$ and $\hat{\theta}_{offset}^{(BP)}$ obtained in the last loop of an active mode.

With the set of values $\{y_i\}$, whether received (in active mode) or extrapolated (in idle mode), and $\{x_i\}$, equation (29) can be used to obtain the outliers, which are the estimated set of residuals $\{\hat{\omega}_i\}$. The application of equation (28) will lead us to an estimation of the autoregressive coefficient $\hat{\rho}$. This coefficient will be used to obtain y_i^*, $T_{1,i}^{(A*)}$ and $T_{1,1}^{(A*)}$ and, regressing the first over the difference of the two last parameters, a new estimation can be obtained for the following values: $\hat{\theta}_{offset}^{*}$, $\hat{\theta}_{skew}^{*}$ and $\hat{v}_i$. This last estimation corresponds to the disturbances, which may be used in equation (28) to get a new estimation of $\hat{\rho}$, and start a new iteration.

Simulations Outline

As stated above, the simulations were performed in two parts: First, the recursive algorithm used was analyzed. Second, the effect of incorporating idle modes over duty cycles without them was assessed. The next subsections detail the content of each of them.

The Recursive Algorithm

This part was divided into five sections:

1. The first section analyzed the dependence of the number of timestamps and second-level iterations to perform correlation measurements on the clock offset and skew. For instance, it is not the same to obtain correlation values in equation (29) with two than with 50 observations. Also, the MSE in the $\{\hat{y}_i\}$ estimations – see equation (27)– for each combination of number of timestamps and second-level iterations was obtained. The dispersion of the results was compared with the CRB to check the MLE property of minimum variation exposed above. A "sufficient" number of observations was defined and determined through simulations. A one-sample t-test with p=.05 was used to compare clock offset and skew estimations with their corresponding true values. This parameter corresponds to the *i* index in equation (6).
2. The second section analyzed the dependence on the results of the variability of the noise generated during the transmission. This variability was considered under three aspects: The initial value of the residuals ω_1 and the disturbances terms $\{v_i\}$. In this section, it was assumed that the noise is normally distributed that is, $\{v_i\}$ was assumed to follow a normal distribution, $\mathcal{N}(0, \sigma_v)$. ω_1 was a random sample of $\mathcal{N}(0, \sigma_\omega)$ Both $\{v_i\}$ and ω_1 were scaled at -3dB. Initially, as Table 4 shows, σ_ω was set to 0.25 and σ_v was fixed at 2. This section left these two parameters to vary separately and analyzed their influence over the previous estimations. Also, the variation of the autocorrelation parameter $\rho\epsilon]0,1[$ was analyzed.
3. The third section analyzed the behavior of the three remaining statistics proposed above for the disturbance terms $\{v_i\}$ in the autocorrelation model, i.e.: exponential, gamma and Weibull, with corresponding parameters assuming values

within a given range. Exponential results were compared to Lee and Chin's (2016), whereas Weibull results were compared to Ahmad et al. (2010). Whereas these statistics may better model real delays, none of them can have zero mean, and hence the corresponding estimations using them will be worse than with gaussian distributions, as will be observed later.

4. The fourth section analyzed the behavior of two different autocorrelation models, found in Cochrane and Orcutt (1949). These models have been presented in equations (34) and (35). The results of this section helped determine which models had the best behavior, in the sense of which of them releases the best $\hat{\theta}_{skew}^{(BP)}$ and $\hat{\theta}_{offset}^{(BP)}$ estimates with a minimum number of iterations, and why.
5. The last section determined which is the best algorithm to update ω_i in the scheme's step 5. This part was considered key to determine the algorithm with the best behavior. Those analyzed here were:

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(1) - \hat{v}_i(q) \tag{36}$$

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(1) - [\hat{v}_i(q) - \hat{v}_i(1)] \tag{37}$$

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(q-1) - \hat{v}_i(q) \tag{38}$$

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(q-1) - [\hat{v}_i(q) - \hat{v}_i(q-1)] \tag{39}$$

In equations (36) to (39), the parameter q represents the iteration ordinal.

During all this synchronization analysis, the duty cycle was composed of active cycles solely, i.e. no idle cycles were included in the simulation scheme. Results served as a reference for the duty cycle analysis.

The Duty Cycle Analysis

This part was divided into two sections:

1. The first section analyzed the effect of introducing idle modes in the recursive scheme. During an idle or sleep mode, estimations of actual clock offset and skew are extrapolated from previously calculated results, but no new messages are processed. The active/idle modes were set from a proportion of 100/0 to 60/40. Also, MSE results on the $\{\hat{y}_i\}$ estimations –see equation (27)– for each combination of number of timestamps and second-level iterations was obtained. The case with no idle modes and the case of a recursive scheme with 400 observations were analyzed in further detail. An important parameter of the evaluations, the time invested in obtaining results against the number of timestamps, was also investigated. This time is important in the second part of the tests, where the constrain of performing a pre-determined number of iterations was lifted. Instead, the program was set to continue iterating until the difference between $\hat{\rho}$ values in two successive iterations was lower than a given number. This new approach resulted in less overflow errors than when the number of iterations was fixed, as in the first part.
2. Finally, a brief exploration of noise levels effects was investigated. Clock offset and skew estimations were presented against the noise level, which was left to vary between -3dB and 0dB.

The first part of the present chapter presented definitions of key concepts used in the ensuing discussion. It also developed the ROS model that was used during the simulations.

Finally, it outlined the simulations performed, whose results and ensuing discussion may be found in Chapter IV, for the recursive algorithm, and Chapter V, for the effect when introducing idle modes. A scheme of the Matlab® program used for the simulations is in the appendix.

IV. RECURSIVE ALGORITHM. SIMULATION RESULTS

This chapter presents the results of testing the recursive algorithm with different parameters values. It is divided into five sections –see the end of chapter III. The final section discusses some programming issues that the researcher found while developing it in Matlab®.

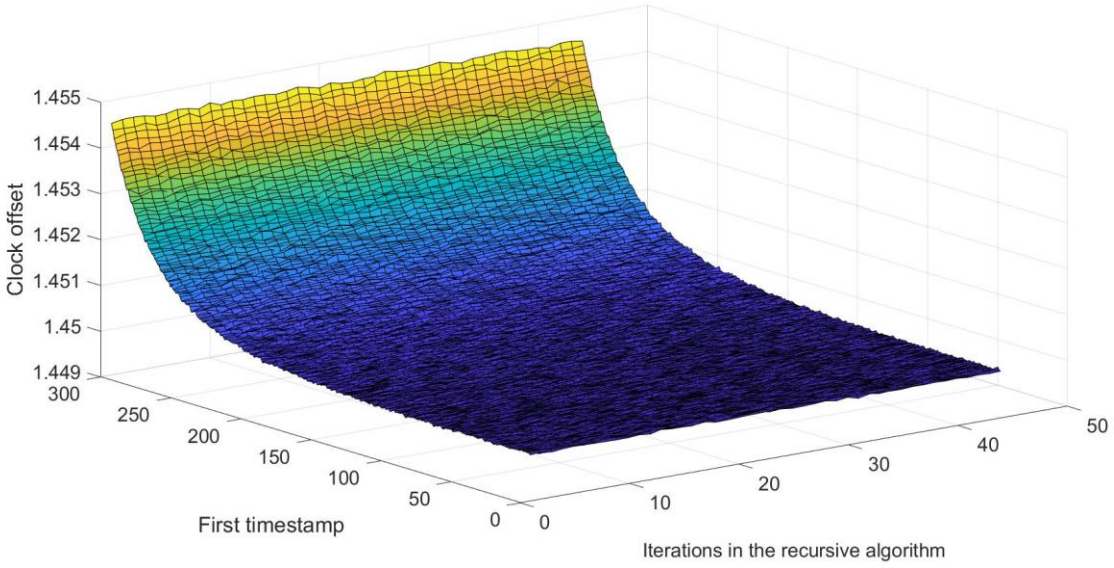
Influence of Timestamps and Iterations

The number of timestamps and the number of iterations in the recursive scheme are parameters that, when combined, help achieve a pre-determined accuracy level. This section analyzes both factors simultaneously.

The model developed used the parameter values indicated in Table 4. The first timestamp taken was initially left to vary between $T_{1,1}^{(A)} = 1$ and $T_{1,1}^{(A)} = 300$, whereas the last observation was fixed at $T_{1,N}^{(A)} = 400$. Therefore, the total number of timestamps $\left\{T_{1,i}^{(A)}\right\}$ took one of the 300 values within the [400,100] range. The number of iterations in the recursive scheme was initially left to vary between 3 and 50. The following figure represents the results of the clock offset estimations, $\hat{\theta}_{offset}^{(BP)}$.

Figure 7

Clock Offset Estimations



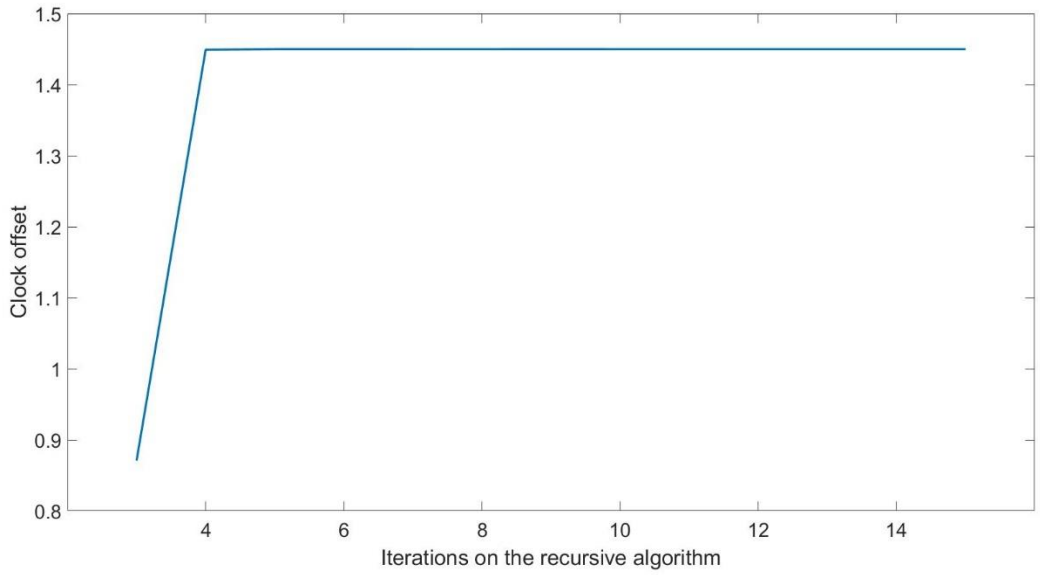
Note. The clock offset obtained was retrofitted to the equivalent of the clock offset at time $T_{1,1}^{(A)} = 1$. This was made to compare these results with the original value, $\theta_{offset} = 1.45$. First-observation messages strings above $T_{1,1}^{(A)} = 300$ appear cut off in the figure, since they resulted in very inaccurate results.

Figure 7 shows that estimations are quite stable if we assume a minimum number of timestamps, on the order of 250, i.e., until $T_{1,1}^{(A)} \sim 150$. However, as we reduce this number further, the clock offset estimations divert significantly from the true value of $\theta_{offset} = 1.45$. The estimations seem independent from the number of iterations in the recursive algorithm. To

check this statement, a plot was obtained with a fixed set of 400 timestamps $\left\{T_{1,1}^{(A)}, \ldots, T_{1,400}^{(A)}\right\}$, in order to better appreciate the evolution of clock offset estimations with the number of iterations:

Figure 8

Clock Offset Estimations with the First Timestamp at Time $T_{1,1}^{(A)}$ and the Last One at $T_{1,400}^{(A)}$.



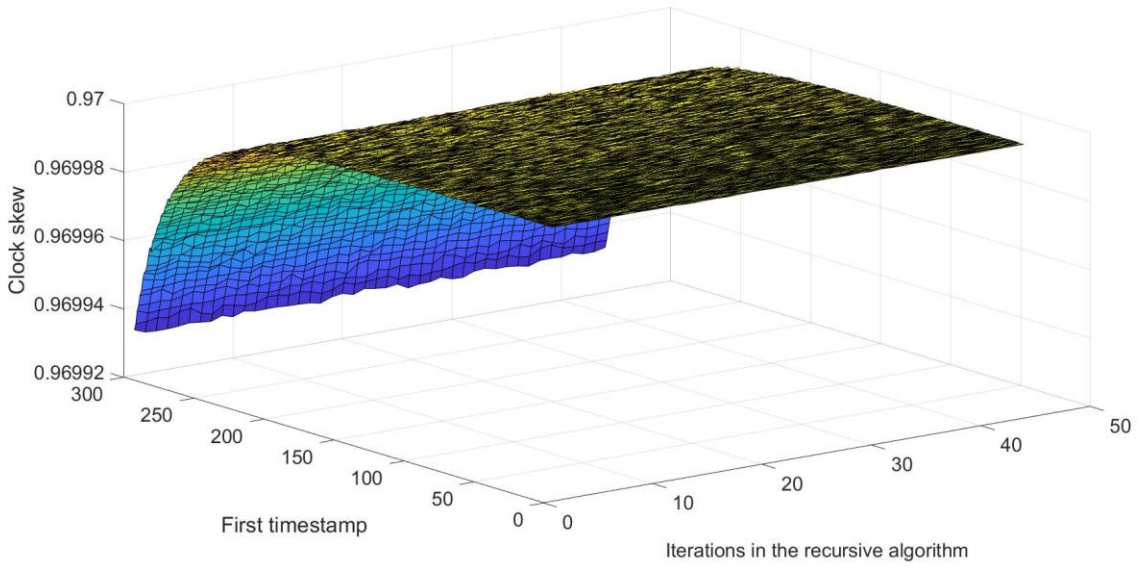
Note. The number of iterations was left to vary between 3 and 15.

Figure 8 shows that the offset estimation levels off by the fifth iteration. For a number of iterations greater than five, the estimation value obtained was exactly the true value 1.45 –with Matlab® precision.

An equivalent analysis was performed with the clock skew estimations $\hat{\theta}_{skew}^{(BP)}$. Figure 9 represents the results obtained.

Figure 9

Clock Skew Estimations



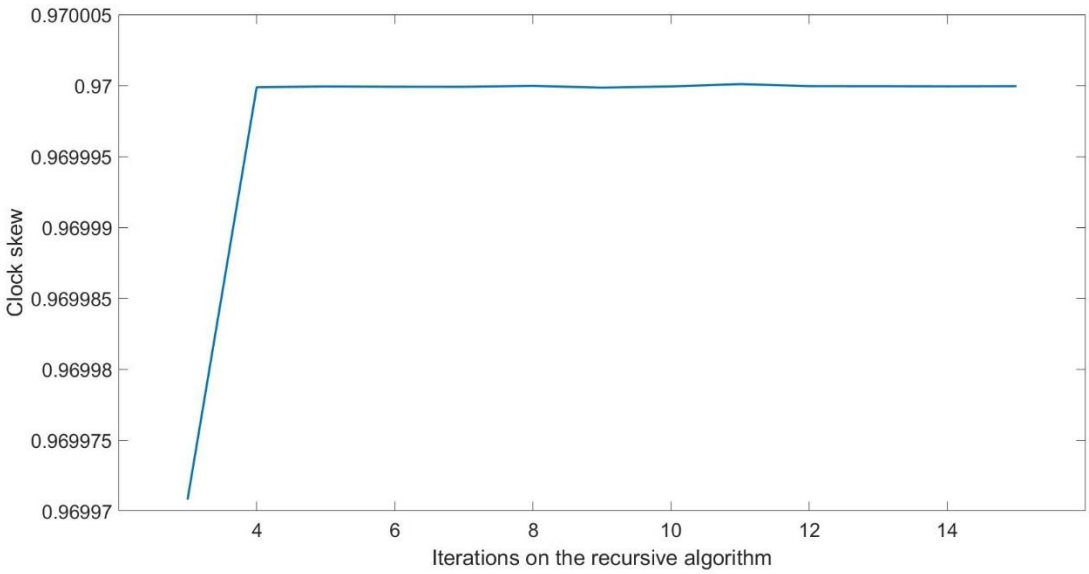
Note. As with the clock offset, the number of timestamps $\left\{T_{1,i}^{(BP)}\right\}$ varied between 400 and 100.

The clock skew estimations converge to the true value much faster than the clock offset. Thus, it was suggested that the algorithm be modified to only evaluate $\hat{\theta}_{skew}$, and then used the timestamps values $\left\{T_{1,i}^{(BP)}\right\}$ to evaluate $\hat{\theta}_{offset}$ –see, for instance, equation (27).

The following figure represents the evolution of clock skew with the first timestamp at $\mathrm{T}_{1,1}^{(\mathrm{A})} = 1$ and the last one at $\mathrm{T}_{1,400}^{(\mathrm{A})} = 400$. It is the same case as that represented in Figure 8, but with the clock skew instead.

Figure 10

Clock Skew Estimations with the First Timestamp at Time $T_{1,1}^{(A)}$ and the Last One at $T_{1,400}^{(A)}$.



Note. As in Figure 8, the number of iterations was left to vary between 3 and 15.

Figure 10 showed a similar pattern than that presented in Figure 8. The original value for clock skew was initially set at $\theta_{skew} = 0.97$. We could observe that by iteration 5 results are accurate to at least four decimal points, 99.9995%.

For each combination of iterations and number of timestamps, the program was run 1,000 times –a tenfold increase with respect to the tests performed by Chatterjee and Venkateswaran (2015). Hence, from each of the $390x48$ samples sets, we obtained 1,000 pair values of $\hat{\theta}_{offset}^{(BP)}$ and $\hat{\theta}_{skew}^{(BP)}$. With these values, estimations of $\hat{y}$ as defined in equation (27) were obtained, and the error in the MSE sense–see equation (1)– was obtained. The following figure presents these results:

Figure 11

MSE Results of $\hat{y}$ Estimations

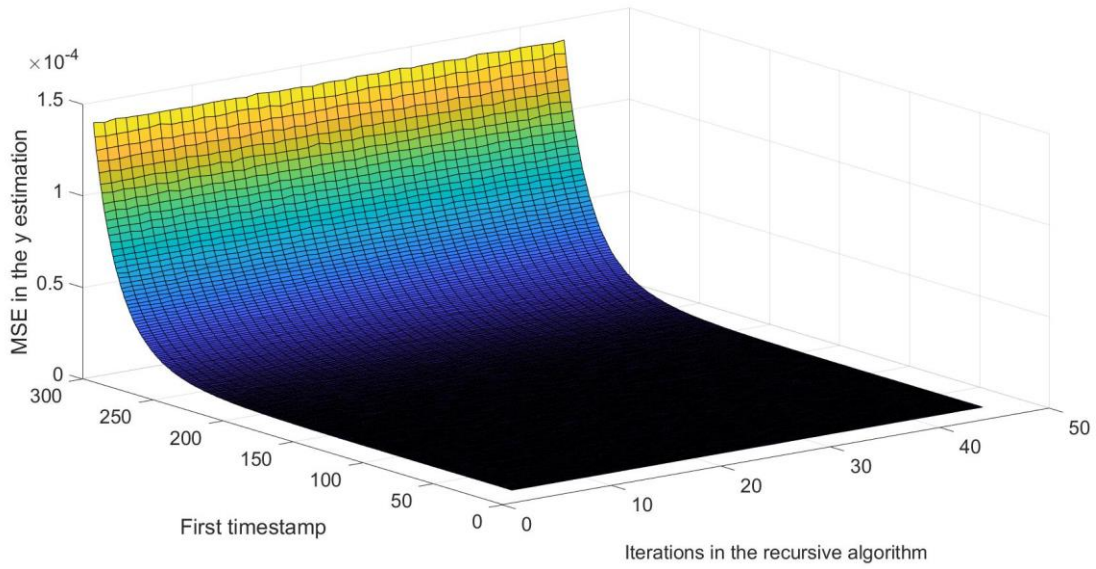


Figure 11 shows that MSE errors are in the order of 10^{-4} for a number of timestamps of at least 100, and its value goes below 10^{-5} as the number of timestamps approach 400. We compare these results with those of two previous works: Chatterjee and Venkateswaran (2015), Noh and Serpedin (2007):

- Chatterjee and Venkateswaran (2015) achieved MSE values of 10^{-3}. However, their paper does not indicate how many timestamps and iterations they were using, and hence corresponding results might not be equivalent. We also saw that, as in the previous cases, the number of iterations did not seem to influence results, at least at a significant level.
- Noh and Serpedin (2007) obtained MSE values of the order of up to 10^{-1} for the clock offset estimations and 10^{-8} for the clock skew's. They correlated up to 50

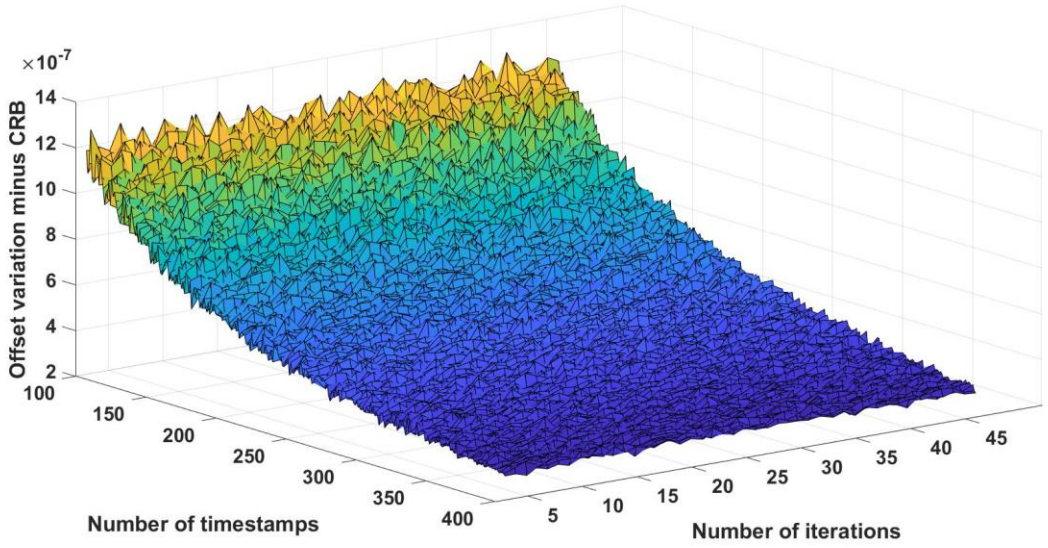
messages, compared to our 400 value. The present tests MSE values are 10^{-6} for the clock offset estimations and 10^{-9} for the clock skew, respectively. In addition to the different number of timestamps $\left\{T_{1,i}^{(BP)}\right\}$ considered, the difference in the results can also be because Noh and Serpedin (2007) assumed that the variable portion of delays is normally distributed, whereas the present dissertation assumes a one-lagged autocorrelation model. Moreover, Noh and Serpedin (2007) did not indicate how many samples they took to obtain the MSE values. The present dissertation took 1,000.

The accuracy of the results exposed above is not the only reference we should consider evaluating the "good behavior" of an algorithm. Indeed, if results are too much spread out, chances are that we eventually get estimations "too far off" the mean. To evaluate this effect, the present author compared the dispersion on the results obtained against the CRB value. However, we must keep in mind that this limit is valid only if the estimations are unbiased (i.e., their expected value $E[\hat{\theta}]$ is identical to the true value θ). The construction of the algorithm detailed in chapter 3 proves that the algorithm is, in fact, unbiased. However, for low values of timestamps and lower values of iterations we cannot ensure that $E\left[\hat{\theta}_{offset}\right] = \theta_{offset}$. So far, we have not checked which combinations of timestamps and iterations comply with this unbiased condition. However, before discriminating the samples obtained, we will check this difference in all the $390x48$ samples sets and observe its evolution across all possible combinations. Thus, the variation within each of these samples was compared with the CRB, as presented in equations (22) and (23) for the clock offset and the clock skew, respectively. The lowest differences

between the variation and the CRB indicate an efficient combination of $\hat{\theta}_{offset}^{(BP)}$ and $\hat{\theta}_{skew}^{(BP)}$. The results are represented in the following figures:

Figure 12

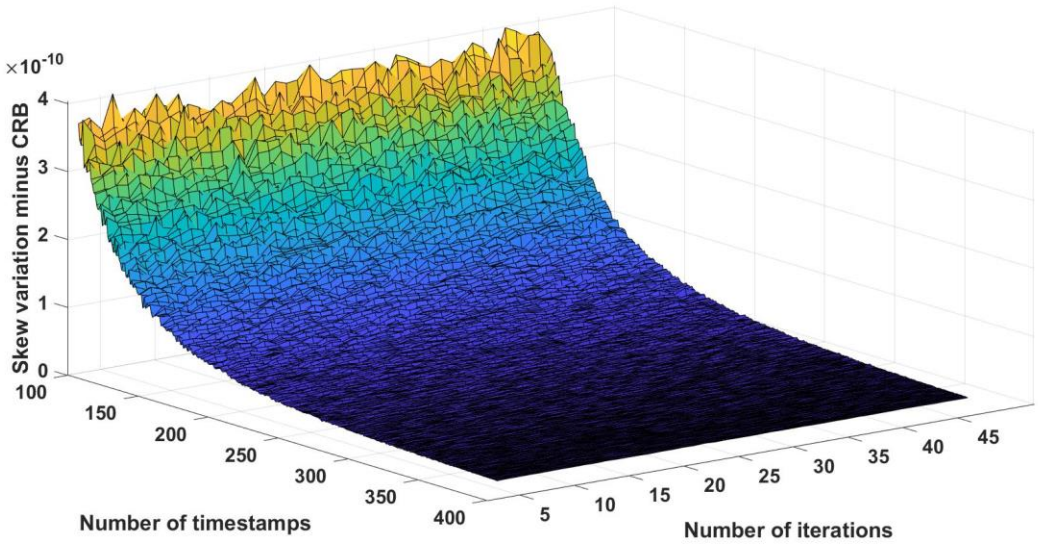
Variation in the Clock Offset Estimation Minus the Cramér Rao Bound (CRB)



Note. The first observation was left to vary within $\left[T_{1,1}^{(A)}, T_{1,300}^{(A)}\right]$, whereas the last observation was set at $T_{1,400}^{(A)}$. The algorithm was iterated between 5 and 50 times.

Figure 13

Variation in the Clock Skew Estimations Minus the Cramér Rao Bound (CRB)



Results in Figure 12 and Figure 13 show that the scheme proposed is very close to the maximum efficiency –differences are on the order of 10^{-7} and 10^{-10} in the cases of clock offset and clock skew, respectively–. It is reasonable to consider that, if we account for the noise introduced in the system through the first residual ω_i and disturbance terms $\{\nu_i\}$, the scheme actually meets the CRB bound with a minimum number of timestamps. In other words, these results demonstrates that the scheme is "efficient", as expected from a MLE.

To fully appreciate the increasing efficiency for a fixed number of iterations –a value of 50 was taken–, a plot was obtained that represents the difference between the dispersion in the data set of the clock offset estimations and the CRB.

Figure 14

Variation of Offset Estimations Minus CRB with 50 Iterations in the Recursive Algorithm

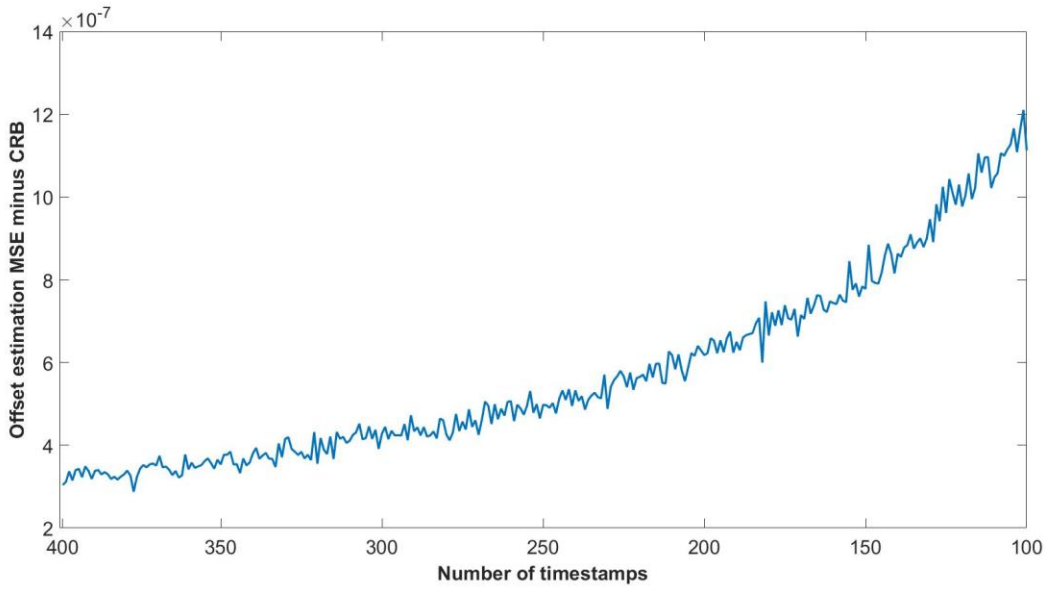


Figure 14 shows that the efficiency increases with the number of timestamps, and note that the chart must be read from right to left in increasing number of timestamps. Considering the noise in the system, it is reasonable to conclude that the algorithm could be considered as fully efficient, particularly when a "sufficiently high" number of timestamps is used. In the tests performed, this means using approximately more than 350 timestamps. An equivalent analysis could have been made with the skew estimations, but current results are considered sufficient to prove the method's efficiency.

The last part of this section consisted of evaluating the accuracy of the results obtained. We performed one-sample t-tests with each of the 390x48 sets of 1,000 samples each. The null hypothesis was H_0: "The clock offset and clock skew estimations come from normal distributions with unknown variance and means equal to θ_{offset} and θ_{skew}, respectively"

(p=.05). Whereas θ_{skew} was fixed at 0.97, the value of θ_{offset} had to be adjusted depending on the first timestamp taken. Thus, for $T_{1,1}^{(A)}$, $\theta_{offset}^{(1)}$=1.45, and the subsequent values were calculated using the following expression:

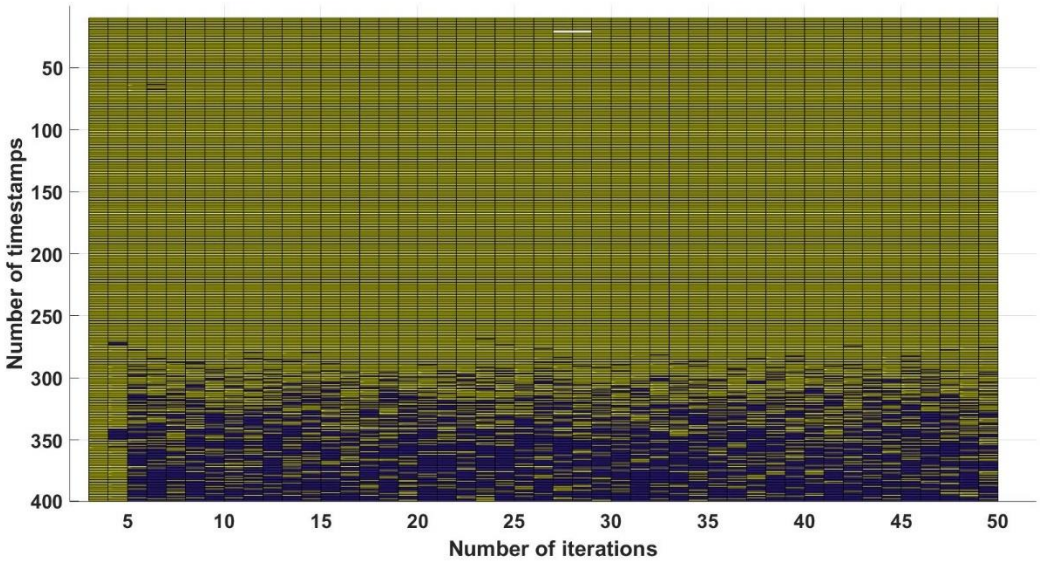
$$\theta_{offset}^{i} = \theta_{offset}^{1} + \theta_{skew}(T_{1,i}^{(A)} - T_{1,1}^{(A)}) \tag{40}$$

In equation (40), *i* took values within the range $[1, 390]$.

Results of the t-tests are represented in the following figure:

Figure 15

T-test Results to Test H_o that Clock Offset and Skew Come from a Normal Distribution with Unknown Variance and θ_{offset}, θ_{skew} Respective Means



Note. Yellow cells show where we reject H_o, whereas blue cells show where we fail to reject H_o at p=.05 level. The degrees of freedom were 999 in each individual sample.

Figure 15 shows two clear sections, separated with an imaginary line at about 250 timestamps, i.e. $T_{1,i}^{150} = 150$. This means that, if we took less than 250 timestamps messages, the application of the recursive scheme would not result in accurate estimations for the clock offset and clock skew. However, as we increase the number of timestamps, accurate results begin to appear, sporadically at first and more consistently afterwards.

These results suggested that some pair of iterations and number of timestamps combined efficiently to deliver results that were not "too far" from the expected values. Of all these possible combinations, the present author decided to retain the following ones:

Table 5

Values Retained After the Simulations

Number of iterations	20
Number of timestamps, N	400: $\left\{T_{1,1}^{(A)}, T_{1,400}^{(A)}\right\}$

These values were fixed for subsequent simulations in this part of the tests, as the behavior of other parameters were analyzed in turn. We now turn our attention to results obtained of noise effects on the estimations.

Influence of Noise

This section analyzed qualitatively the influence of noise on the results. As stated earlier, noise was introduced in the simulations through three parameters: The first value of the residuals ω_1, the disturbance terms $\{v_i\}$, and the correlation parameter ρ. Whereas the level of noise was fixed at -3dB with respect to the signal, these tests set dealt with the effect of the noise dispersion, or standard deviation. Of the three, ρ does not provide information of this dispersion,

and thus we considered that it only had an indirect influence on the results. However, ρ remains a very sensitive parameter, since with values of $\rho > 1$ the algorithm becomes unstable and the estimations fall completely off the true values. We analyzed the effect of these three parameters over $\hat{y}$ estimations in the MSE sense in turn.

Figure 16

Effect of the Disturbance's Standard Deviation σ_ν over $\hat{y}$ Accuracy

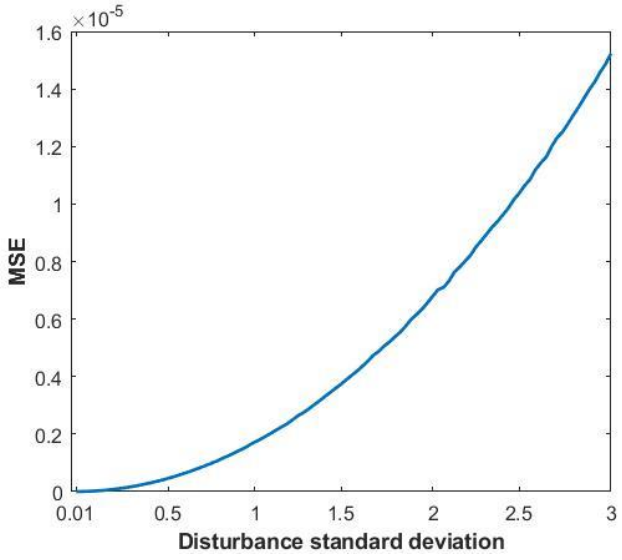
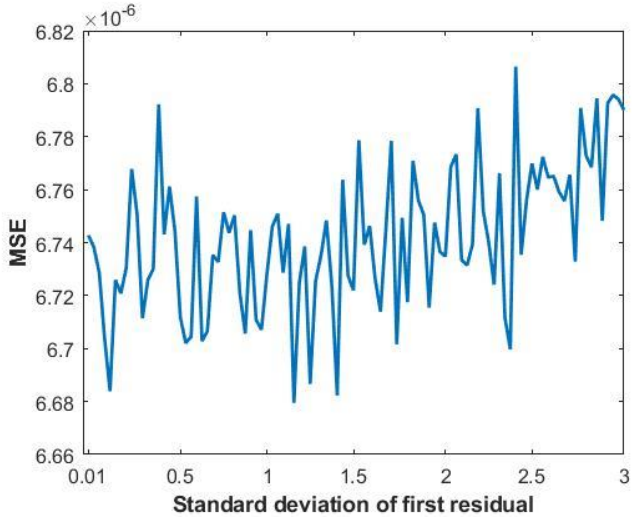


Figure 16 shows that MSE values increase consistently with the standard deviation of the disturbances $\{\nu_i\}$. By default, the test took a σ_ν value of 2.

Figure 17

Effect of the Standard Deviation σ_ω of the First Residual ω_1 over $\hat{y}$ Accuracy



A comparison between Figure 16 and Figure 17 shows that the effect of ω_1 standard deviation is not as clear as the effect of the disturbances' $\{\nu_i\}$ –the former is one order of magnitude lower, at least within the range considered. This can be explained by the fact that in the last case, σ_ν value extends over the whole set of the disturbances $\{\nu_i\}$ with $i = [1, N]$. On the contrary, in the case of the residuals the standard deviation only affects ω_1; the rest of the residuals' values $\{\omega_i\}$ within $i = [2, N]$ are only partially affected through ρ, particularly for values of ρ close to –but not greater than– 1. In other words, ρ cushions the propagating effects. To prove this statement, a third analysis was performed in which $\omega_1 = \mathcal{N}(0,0.25)$, $\sigma_\nu = 2$, and ρ was left to vary $\rho = [0.01,0.99]$:

Figure 18

Effect of the Autocorrelation Value ρ over $\hat{y}$ Estimation Accuracy

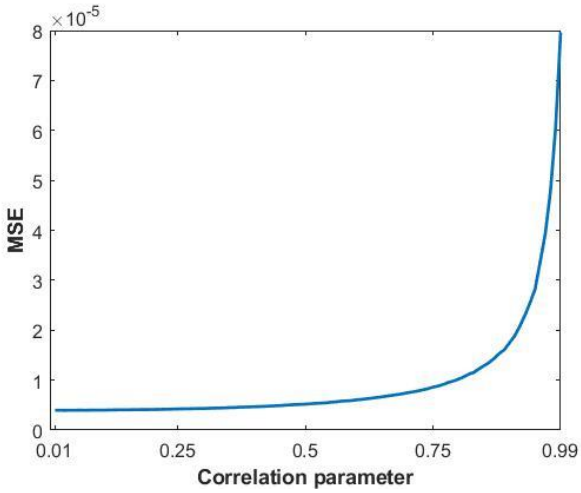


Figure 18 shows that ρ values below 0.5 do not propagate the errors due to residuals significantly. However, after this value the effects become very significant and may negatively impact the algorithm's efficiency.

The influence of σ_v over the results accuracy is thus proven. We also showed that, whereas the effect of σ_ω is limited, it can be boosted if ρ values were too close to unity. These facts must be considered if the model is to be applied in the real world. Once the dispersion of noise has been analyzed, we dedicate the next section to four noise models found in the field literature.

Influence of Delays' Models

As was stated in chapter 2 of the present dissertation, noise is a stochastic parameter that may be modeled with one of the following statistics: gaussian, exponential, gamma and Weibull. Whereas the gaussian distribution was the statistics used by default in the simulations performed

so far, the next set of simulations compared the estimated values of clock offset and skew that would have been obtained if any of the three remaining statistics had been used instead. The fact that the mean of these statistics is never zero, and hence the disturbance values can never be symmetrically distributed about the origin, will have a negative impact in the estimations –see Figure 19, Figure 20, Figure 23, Figure 24, Figure 25 and Figure 26. Chapter 2 in the present dissertation stated that a zero-mean distribution of the errors was a requirement for the maximum likelihood estimator (MLE). Hence, the following cases do not constitute MLEs, and the estimations are necessarily worse than with a gaussian assumption. As has been said before, however, certain noise sources match non-gaussian models better, hence the interest of checking their behavior under the following conditions:

Table 6

Fixed Parameters in the Noise Distribution Section of the Tests

Standard deviation of the disturbances, σ_v	2 (Gauss distribution)
Standard deviation of the first residual ω_1, σ_ω	0.25 (Gauss distribution)
Autocorrelation parameter, ρ	0.65
Noise level	-3dB

In the remaining tests, a new change in the iterative scheme was adopted. This change, which will also be used during the second part of the tests –see Chapter 5–, consisted on the following: Instead of fixing the iterations on the recursive scheme beforehand (to 20 as indicated

in Table 5), we used a *while* loop with the following two conditions: 1.- Repeat until the difference between two consecutive estimations of the autoregression parameter $\hat{\rho}$ is lower than 0.01, this value was also used in Chatterjee and Venkateswaran (2015); 2.- If the number of iterations goes beyond a certain value (which can be as low as 10, as shown in the exponential case), stop the loop and take the last value of $\hat{\rho}$ obtained as the best estimation possible.

Exponential Distribution

The exponential distribution is a one-parameter curve with the following probability density function:

$$y = f(x|\mu) = \frac{1}{\mu} e^{\frac{-x}{\mu}} \tag{41}$$

A simulation with a set of 1,000 samples was used to compare estimations of clock offset and clock skew between disturbances with a gauss and exponential distribution. 100 different means were taken within the range $\lambda \in [1,10]$. Other key values are indicated in Table 5 and Table 6. The true clock offset value was set at $\theta_{offset} = 1.45$, and the true clock skew value was set at $\theta_{skew} = 0.97$.

Figure 19

Comparison on Clock Offset Estimates when the Disturbances Come from a Gaussian and an Exponential Distribution

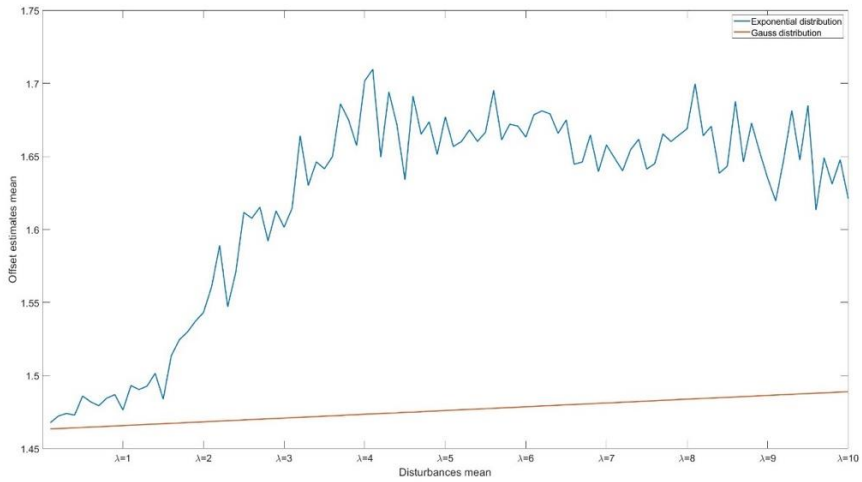


Figure 20

Comparison on Clock Skew Estimates when the Disturbances Come from a Gaussian and an Exponential Distribution

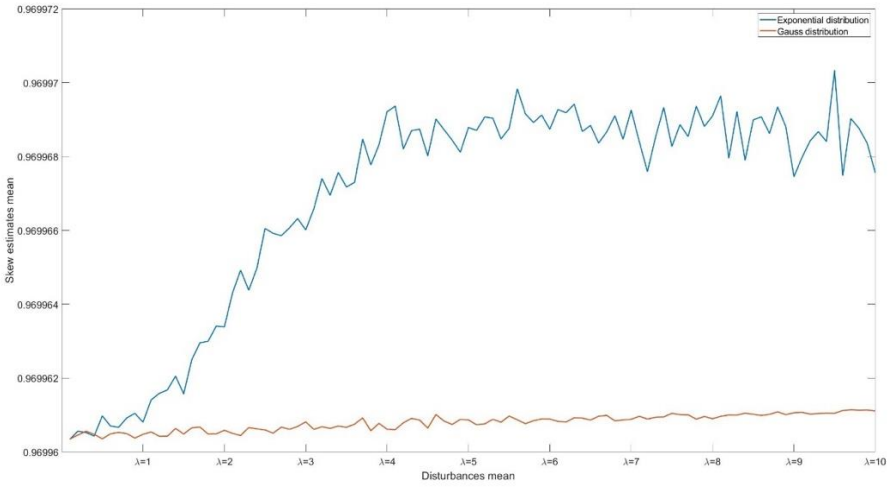


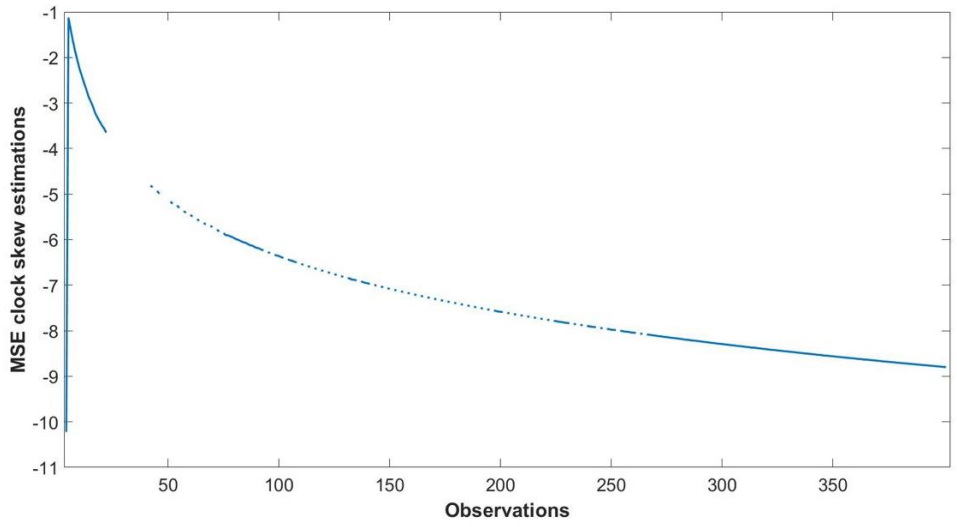
Figure 19 shows that the effect of increasing the exponential distribution mean has a significant effect on the offset estimates. Observe that this effect is not so important with a gauss distribution. However, remember that the noise effects were dampened at a -3dB in time, which is the factor by which the means represented in the figure should be multiplied to obtain the real mean values –x axis. If we remember that exponential distributions were good to model noise in non-line of sight (NLOS) operations, results suggest that the distribution of noise sources in these environments do not deliver good estimates.

Figure 20, on the contrary, shows that the effect of a non-zero mean is not so drastic in the skew estimation process. This could be explained as follows: Non-zero means of the disturbances are absorbed by the offset estimate –the intercept in our linear regression model– in every iterative step of the scheme, but the skew estimate –the slope in the model– remains less affected.

Lee and Chin (2016) obtained results using an exponential distribution for the random part of the delays, with a mean of $3.33\mu sec$. They used a receiver-receiver synchronization scheme, not a ROS, and the results of MSE clock skew estimations with 25 observations (timestamped messages) were of the order of 10^{-18} μsec^2, whereas the MSE clock offset estimations were of the order of 10^{-9} sec^2. We reproduced these simulations in our ROS scheme:

Figure 21

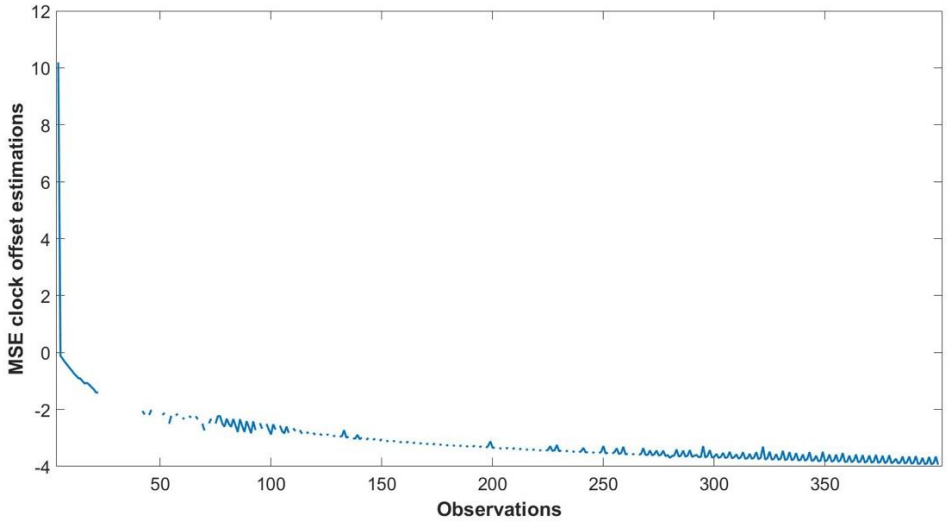
Skew Estimations. Results Comparison with Lee and Chin, 2016



Note. The *y* axis is presented in a logarithmic scale (base 10).

Figure 22

Offset Estimations. Results Comparison with Lee and Chin, 2016



Note. The *y* axis is presented in a logarithmic scale (base 10).

An explanation on the discontinuous graphs above: Figure 21 and Figure 22 were obtained with a program which dropped those results where convergence was not achieved after 10 iterations. This was the case in 74 out of 398 cases, or about 18% –iterations with less than three observations were not performed since the figures above show that results were completely inaccurate. Our results were far less accurate than Lee and Chin's (2016). Although we think that MSE results for skew estimations of the order of 10^{-7} are still good, the difference between offset estimations in our model and Lee and Chin's is important. Once again, this shows that our model with autocorrelated, non-zero mean noise is not appropriate, and another scheme such a receiver-receiver protocol should be used instead. In any case, good MSE values are only reached after around 200 observations are considered, again far worse than Lee and Chin's values of about 25. On a positive note, it is interesting to note that, when it happens, convergence is very fast: In 182

out of 398 of the cases, or 45%, less than five iterations were needed to reach a difference of two consecutive $\hat{\rho}$ values of less than 0.01.

The next two sections analyze two distributions, gamma and Weibull, that can only take positive values as well. These two distributions are two-parametric. Thus, the effect on the estimations was analyzed based on the values these parameters take in combination.

Gamma Distribution

The gamma distribution is a two-parameter curve with the following probability density function:

$$Gamma(\alpha, \beta) = f(x; \alpha, \beta) = \frac{\beta^{\alpha} x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \quad (42)$$

In equation (42), α and β are the shape and scale parameters, respectively. The gamma distribution may be good to represent noise generated in multipath environments (Chen et al., 2020). The mean of a gamma distribution is: $\alpha\beta$. Thus, we would expect that estimations worsen with relatively high combined values of α and β.

The following figures represent clock skew and clock offset estimations over 1,000-size data samples, with the following range of gamma parameters' values: $\alpha = [0.5,10]$ and $\beta = [0.5,1]$.

Figure 23

Clock Skew Estimations with Gamma Distribution of Disturbances

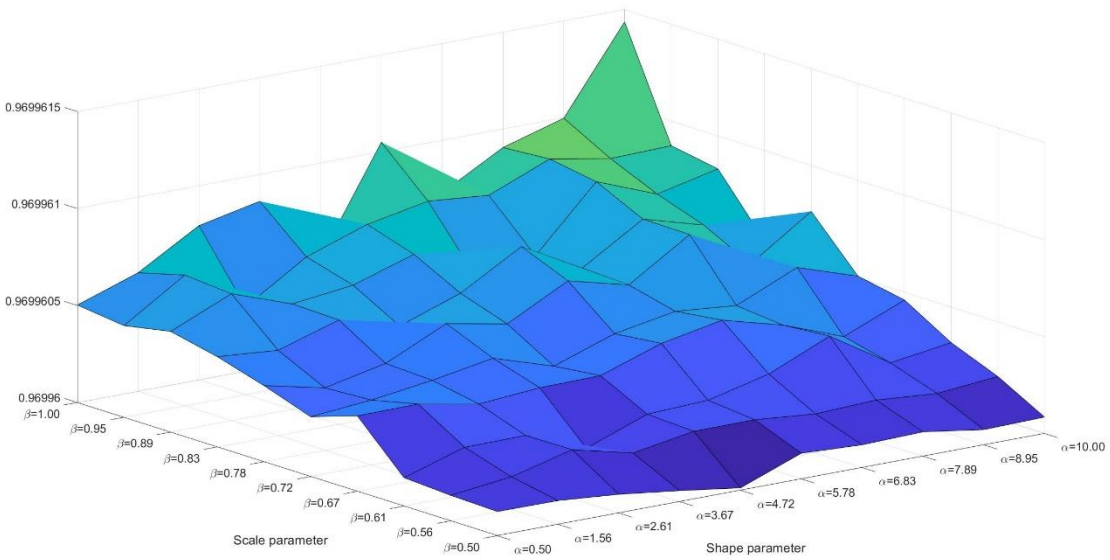


Figure 24

Clock Offset Estimations with Gamma Distribution of Disturbances

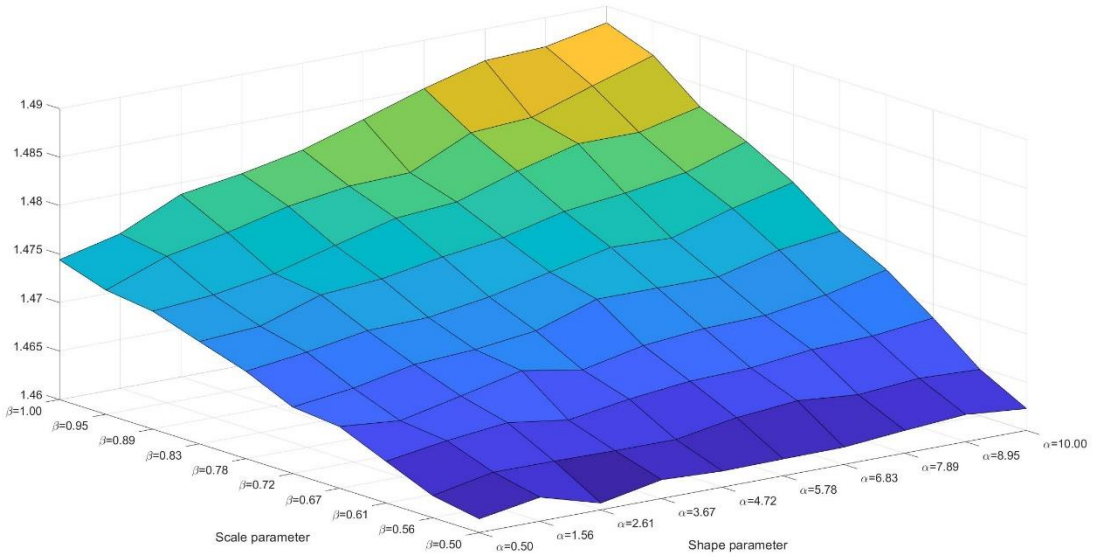


Figure 23 and Figure 24 show that the scale parameter $\boldsymbol{\beta}$ influence over the results but increases with the shape parameter $\boldsymbol{\alpha}$. Results suggest that, in general, gamma distribution results in similar clock skew estimations than the exponential distribution. The reason of the inaccuracies in offset estimations shown in Figure 23 and Figure 24 may be the same as explained in the exponential distributions, i.e., the scheme does not behave well when the mean noise level is different from 0. Considering the mean of the distribution $-\alpha\beta-$, it can be concluded that far-off reflective surfaces worsen the results more than if located closer to the receiver. Gamma distribution has a generalized version to which Weibull distributions can be reduced with an adequate choice of parameter values. The generalized gamma distribution was not analyzed in the present dissertation.

Weibull Distribution

The Weibull distribution is, like gamma, a two-parameter distribution. The probability density function is represented by:

$$Weibull(\alpha,\beta) = f(x;\alpha,\beta) = \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} e^{-(x/\alpha)^{\beta}} \quad (43)$$

In equation (43), α and β are the scale and shape parameter, respectively. The same analysis made with the gamma distribution is reproduced here. Values taken are: $\beta = [0.5,1]$ and $\alpha = [0.5,10]$.

Figure 25

Clock Skew Estimations with Weibull Distribution of Disturbances

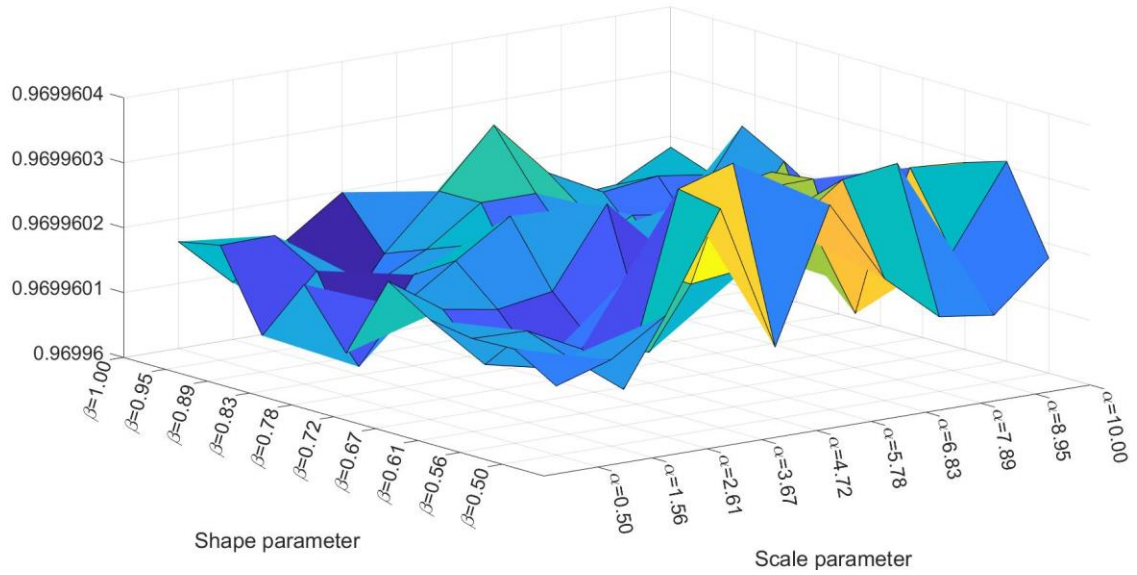


Figure 26

Clock Offset Estimations with Weibull Distribution of Disturbances

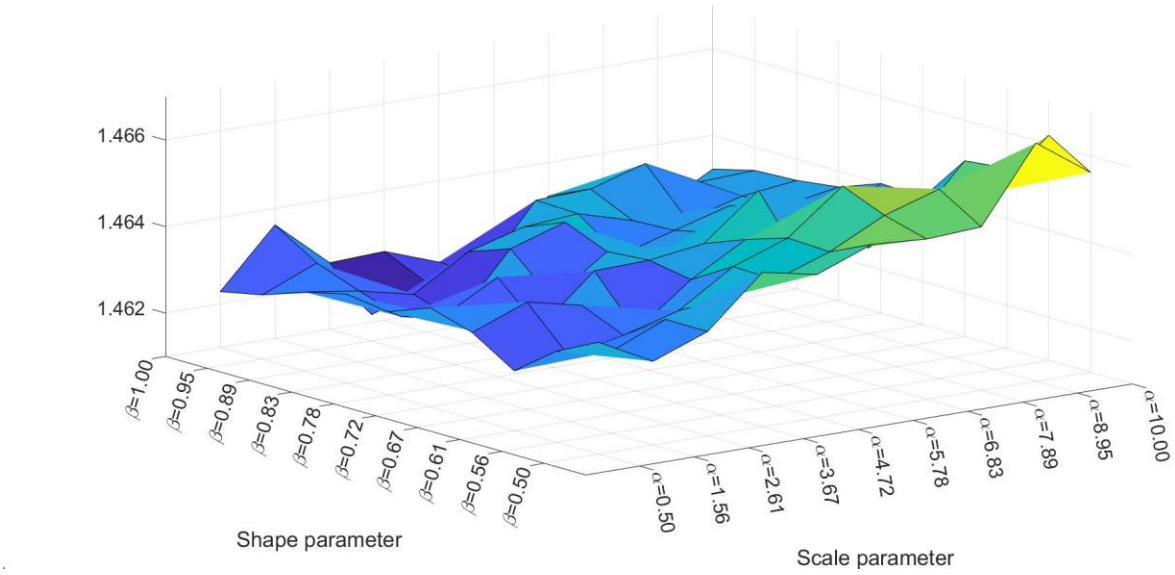


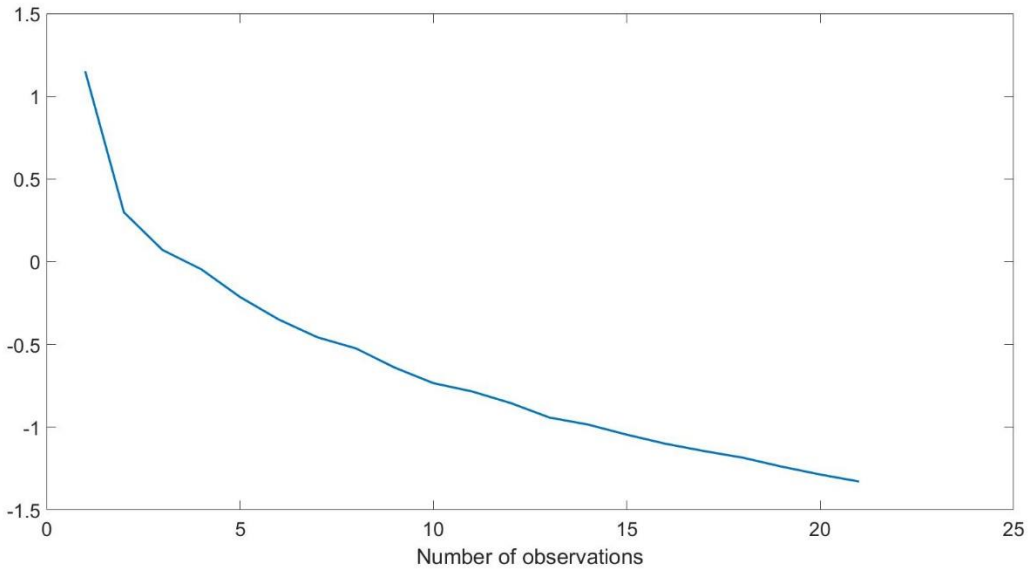
Figure 26 show that the clock offset estimations get worse with a combination of low shape β and high scale α parameters values. As in the precedent cases, the mean of a Weibull distribution is $\alpha\Gamma(1+1/\beta)$, which means that distributions of disturbances $\{v_i\}$ and first residual ω_1 with high means deliver worst results, since it produces more outliers for the receiving times. Also, a comparison between Figure 25 and Figure 26 show that clock skew is less affected by high noise means.

Ahmad et al. (2010) obtained results for MSE of clock offset estimations $\{\hat{\theta}^i_{offset} - \theta_{offset}\}_N$ using Weibull distributions with shape parameter β=2, scale parameter $\alpha = 5$ and with several observations ranging from 2 to 22. They analyzed two cases: one case where fixed delays (d) were known and the other where the fixed delays were unknown. They used a two-way synchronization protocol. The MSE offset decrease with the number of samples taken, up to a

value of about 10^{-2} for *d* unknown and 0.06 for *d* known. Their tests considered no clock skew. The present dissertation reproduced the tests with 22 timestamps messages and obtained MSE values of offset estimations over a sample of 1,000 elements for several observations ranging from 2 to 22, and $\theta_{offset} = 1.45$ and $\theta_{skew} = 0.97$, as usual. The results are shown below:

Figure 27

MSE of Clock Offset Estimations with $\beta = 2$, $\alpha = 5$



Note. Values are represented in a logarithmic scale. Number of samples: 1,000.

We observe that the scheme proposed in the present dissertation are of the same order that those of Ahmad et al. (2010) for the case of fixed delays known. However, they are worse than those of Ahmad et al. with unknown fixed delays. If the number of observations was 400, as is the case in the tests performed earlier, the MSE clock offset estimations would be of the order of 10^{-4}, two orders of magnitude below both cases (i.e., *d* known and *d* unknown). Once again,

we observe that the ROS-based scheme, although potentially more accurate, presents a lower convergence. As a reminder, the time synchronization proposed in the present dissertation assumes that *d* is unknown.

This section provided a qualitative overlook at some noise distribution models. Although the gaussian distribution is the most common model used in the literature by far, comparing results with different schemes highlighted interesting conclusions about similarities and differences between them. We can conclude that, in general, gaussian and exponential distributions lead to results which are closer to the true values, whereas gamma distribution is the less accurate. Weibull distributions lie somewhere in the middle. In any case, no distribution with mean different from zero will yield to a maximum likelihood estimator (MLE), and hence gaussian models, when duly justified, will always be more accurate in their outcomes.

This concludes our brief analysis on the noise effects. We now turn our attention to two additional aspects of our particular algorithm: the autocorrelation model and the residuals' updates.

Autocorrelation Models

Two models were analyzed for autocorrelation, which have been presented above –see equations (35) and (34). They are repeated here for convenience:

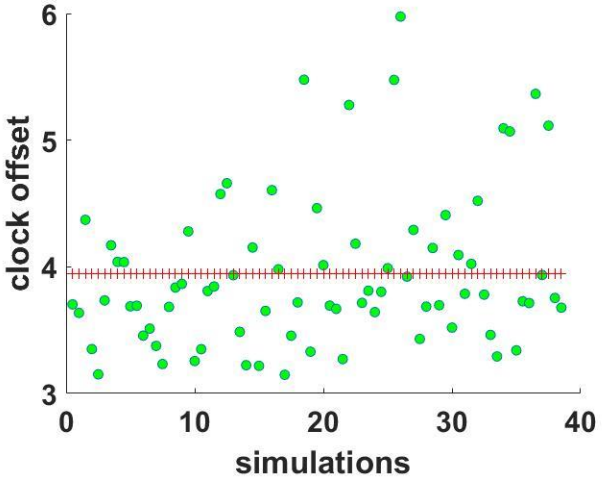
$$\omega_i = \rho_1 \omega_{i-1} + \rho_2 \omega_{i-2} \tag{44}$$

$$\omega_i = \omega_{i-1} + \rho(\omega_{i-1} - \omega_{i-2}) + \nu_i \tag{45}$$

The first case did not deliver accurate results, at least for clock offset estimations, as we can see in Figure 28. Values of the independent variables used were those specified in Table 5 and Table 6. The autocorrelation parameters were $\rho_1 = 0.85$ and $\rho_2 = 0.15$.

Figure 28

Clock Offset Estimations with Two Autocorrelation Parameters



Note. To obtain these results, the following main parameters were used: $\rho_1 = 0.85$, $\rho_2 = 0.15$, $\theta_{offset} = 1.45$. 23 outliers –those whose distance to the mean was more than three standard deviations– out of a total of 100 values were removed from the final plot. The average clock offset estimation was 3.9419.

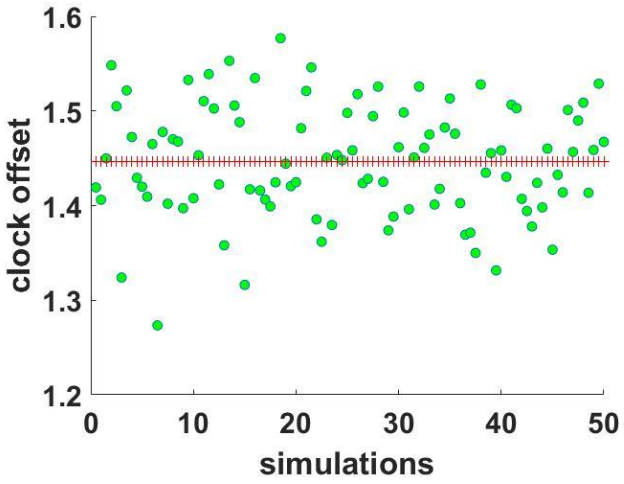
Figure 28 shows a clear bias of the results with respect to the true value of the clock offset, $\theta_{offset} = 1.45$. A plausible explanation might lie in the recursive scheme used: Any error in the first estimation of the autocorrelation parameters $\hat{\rho}_1$ and $\hat{\rho}_2$ is absorbed in the estimation of the disturbances $\{\hat{v}_i\}$, which are used in turn to estimate the residuals $\{\hat{\omega}_i\}$. These were used to evaluate $\hat{\rho}_1$ and $\hat{\rho}_2$ in the second iteration, and so on. This phenomenon is also present in the case

of the one-lagged autocorrelation model presented in equation (28), but the effect is boosted with two parameters as is the present case.

When the residuals model represented in equation (45) was used instead, results were much more accurate:

Figure 29

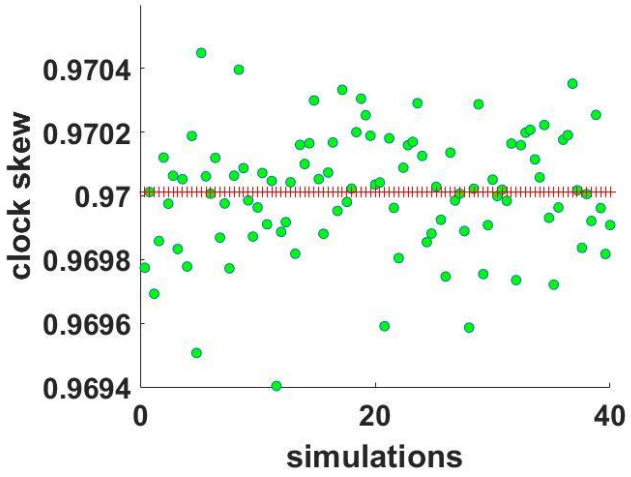
Clock Offset Estimations with Two-Lagged Autocorrelation Scheme



Note. A t-student test was performed to check whether the estimations differ significantly from a normal distribution with mean the true offset value, $\theta_{offset} = 1.45$. Results were not significant t(99)=.5676, p=.05.

Figure 30

Clock Skew Estimations with Two-lagged Autocorrelation Scheme



Note. A t-student test was performed to check whether the estimations differ significantly from a normal distribution with mean the true offset value, $\theta_{offset} = 1.45$. As with the clock offset, results were not significant t(99)=.5515, p=.05.

The application of the autocorrelation scheme represented by equation (45) also highlighted an interesting fact that could be applied to any of the schemes considered: Since the accumulation of errors detected in these schemes affected the estimation of the clock offset more than the clock skew, one solution would be to remove the clock offset estimation from the scheme altogether, as is was done in this case. With the obtention of the clock skew, we could apply equation (27) to get $\hat{\theta}_{offset}$ through averaging the expression over all the timestamps:

$$\hat{\theta}_{offset} = y_i - \hat{\theta}_{skew} * \left(T_{1,i}^{(A)} - T_{1,1}^{(A)}\right) - \hat{\omega}_i \quad (46)$$

This solution, although considered very efficient, would not be applied to subsequent simulations.

Residuals Estimation's Updates Strategies, $\widehat{\omega}_{next}^{i}$

In the last section of the first part of the tests, the accuracy in the results were compared with different schemes of residuals estimation's updates. Four of these schemes were considered, those represented by equations (36), (37), (38) and (39) which, for reasons of convenience, are repeated here:

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(1) - \hat{v}_i(q) \tag{47}$$

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(1) - [\hat{v}_i(q) - \hat{v}_i(1)] \tag{48}$$

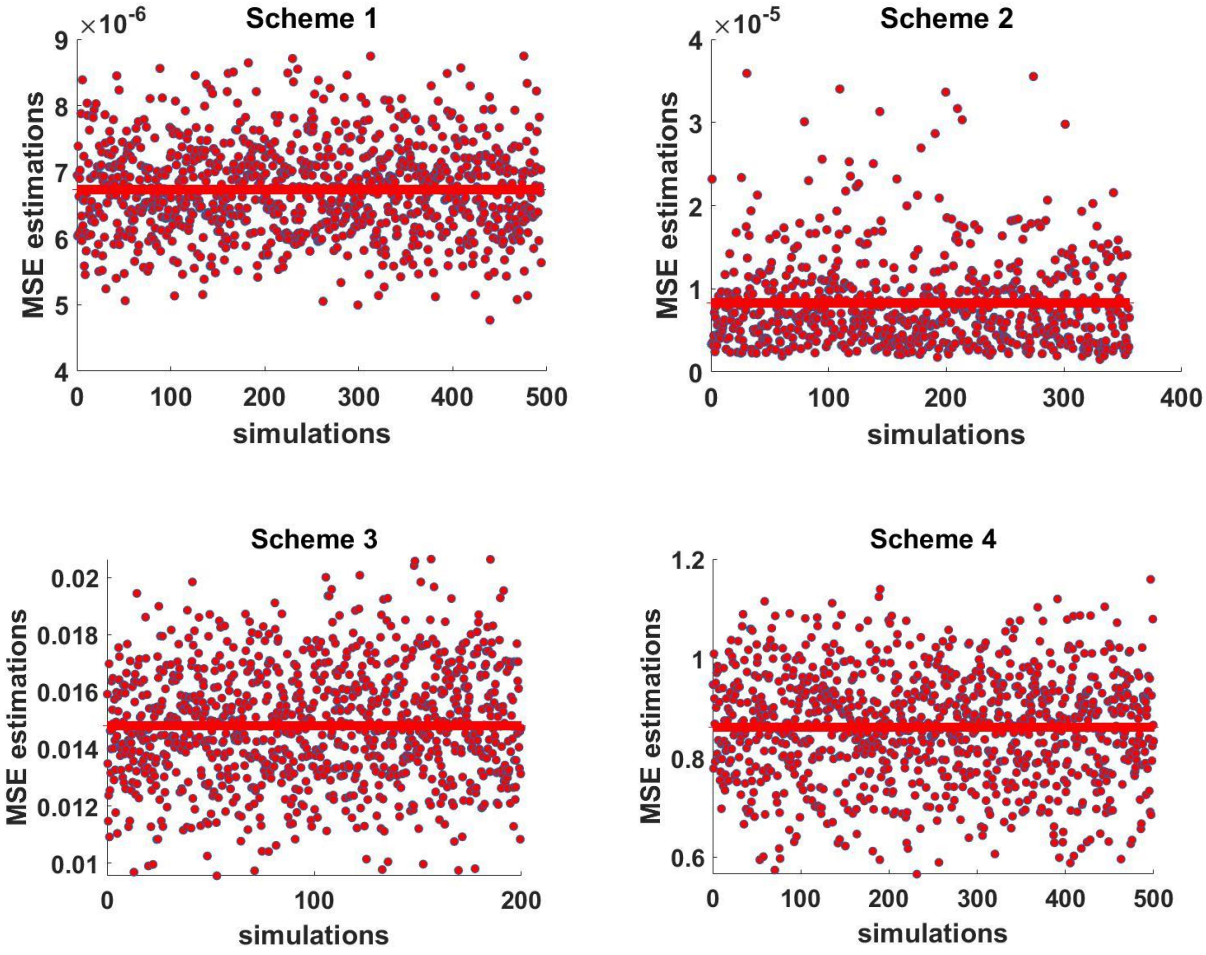
$$\widehat{\omega}_i(q) = \widehat{\omega}_i(q-1) - \hat{v}_i(q) \tag{49}$$

$$\widehat{\omega}_i(q) = \widehat{\omega}_i(q-1) - [\hat{v}_i(q) - \hat{v}_i(q-1)] \tag{50}$$

In equations (47) through (50), q represents the iteration order. Thus, for example, $\widehat{\omega}_i(q-1)$ corresponds to the estimation of the residuals in iteration *q-1*. All results are presented together in the following figure for better comparison:

Figure 30

Residuals Estimation's Update Schemes



Note. In each case, outliers farther than three standard deviations from the mean have been removed from the data. The initial samples were composed of 1,000 elements.

With an accuracy of one, three and five orders of magnitude better than schemes 2, 3 and 4 respectively, we conclude that the first scheme delivered more accurate results. This can be explained by the fact that, in schemes 3 and 4 –equations (49) and (50), respectively– the errors in the residuals estimations are carried on from one iteration to the next one, whereas in schemes 1 and 2 –equations (47) and (48)– the new estimations of disturbances are applied over the first

estimation of the residuals only. Scheme 1 was the scheme used to deliver the previous results of this part.

Additional Remarks

Autocorrelated Noise

The ROS scheme as presented in Chatterjee and Venkateswaran (2015) assumes a one-lag autocorrelated noise model. This needs some justification. When discussing the minimum squared error (MSE) as a maximum likelihood estimator (MLE) in chapter 2, a mention was made to the Gauss-Markov theorem, and the conditions it imposed errors which should be independent and identically distributed (i.i.d.). Structured residuals, that prevent the direct application of MSE to obtain the clock skew and clock offset and therefore force the use of the recursive scheme proposed by Chatterjee and Venkateswaran (2015), can happen in two circumstances (Thejll & Schmith, 2005): a) missing key predictors, and b) mixing key variables with different levels of serial correlation. The common effect in these two cases is the correlation of residuals, as assumed in the present dissertation. Some studies have detected a strong tendency to signals mixing in multipath environments (Sharma & Mathur, 2018), which could in turn result in the residuals being autocorrelated. Correlated noise has also been detected in analyses of signals from antennas with deformed reflectors (Tan et al., 1996). Last, it should be noted that closely spaced signals as in the present case have a higher probability of resulting in autocorrelated noise.

Be that as it may, the case remains worth the present effort even if no noise correlation was present. Although in this case the recursive scheme could be circumvented in favor of a

direct application of classical regression analysis, the present approach would still be valid, only with taking $\rho = 0$.

Issues Regarding the Matlab Functions Used

During the execution of the simulations, the Matlab® program presented some issues that had to be fixed ad hoc. Most of these problems could be reduced to two cases, $\hat{\rho}$ estimates bigger than 1 and disturbances estimates with expected values different from 0, $E[\{\hat{v}_i\}] \neq 0$.

In the event of an iteration delivering $\hat{\rho} > 1$, the problem of finding the first estimates in the star values x^* and y^*would deliver complex values; see equations (18)and (19). These values would corrupt the rest of the estimated values until calculations would overflow, bringing the loops to an unexpected stop. This issue was solved in different ways through the tests performed. Sometimes, a strong condition was imposed on the $\hat{\rho}$ estimations, whose value had to be within 0 and 1, $\hat{\rho} \in]0,1[$. However, this solution was applied as a last resort, since in principle the estimations should not deliver $\hat{\rho}$ values outside the specified range, particularly with true values far from the extremes. This brought the author's attention to the evaluation of linear models of the type $Ax = B$ to obtain x, which Matlab® can solve using different commands. Four of them were tried: *regress*, *polyfit*, *mldivide* and *fitlm*. The initial recursive algorithm using an autocorrelation model as expressed by equation (28) and the residuals' $\{\omega_i\}$ updates given by equation (47) delivered best results with *mldivide*. Moreover, no need to impose the range condition on $\hat{\rho}$ was necessary in this case. Nevertheless, problems emerged as these schemes grew in complexity. For instance, when taking more observations, or iterating a higher number of times. Mathworks (n.d.a) indicates that *mldivide* can yield to program errors if the matrix $[A]$ is nearly singular. In these cases, the strong condition on $\hat{\rho} \in]0,1[$ had to be imposed, and *fitlm*

appeared to have the best behavior. In this respect, Mathworks (n.d.b) informs that *fitlm* identifies and removes NaN values in the calculations, thus decreasing the probability of errors.

Noise Level

The level of noise also appeared to be critical in the program stability. This noise level affected both the first residual ω_1 and the disturbances $\{v_i\}$, for which a possible physical explanation of the phenomenon is ventured: In the case of weak signals, the information is lost beneath the noise levels, and the estimations worsen. For this reason, it was decided to give the noise a value of -3dB, or 10^{-3} below the signals of interest. The effect of noise on results is specifically analyzed in the second part of the simulations –see Chapter 5.

This chapter analyzed several options for the recursive algorithm used. In the next chapter, we introduce the concept of a duty cycle.

V. DUTY CYCLE. SIMULATIONS RESULTS

The development of a program to run the tests presented here came as a result of overcoming significant difficulties. The most important ones are presented in this section.

First, results presented in the previous chapter were assumed as proven. For instance, it was assumed that we need 400 timestamped messages $\left\{T_{1,i}^{(BP)}\right\}$ to achieve estimates with a sufficient level of accuracy. Thus, it was not considered necessary to perform test with messages strings larger than this number. Also, the noise level was initially set to -3dB, with a gaussian distribution. Other parameter values such as those presented in Table 4 and Table 6 were retained, except that the number of first level iterations was lowered from 1,000 to 100, in order to save computing time.

At first, several iterations in the recursive algorithm of 20 was used, but results were not accurate enough. Simulations were run for numerous timestamps between 4 and 400, to analyze how the scheme built up accurate outcomes. Therefore, the regression parameter of the residuals $\hat{\rho}$ was taken as the reference parameter, and the algorithm was iterated if the difference between two successive estimations of $\hat{\rho}$ was larger than .0001. Thus, the number of iterations depended on the number of timestamped messages taken –with the mentioned limit of 400. To guarantee $\hat{\rho}$ within the range of stability, i.e. $]0,1[$, $\hat{\rho}$ values outside this range were immediately replaced by .5 in the following iteration.

Another important factor was the scheme to update residuals estimations $\widehat{\omega}_i$ analyzed in Figure 30. It was observed that the program's overflow risk seriously compromised its stability. The cause of this behavior was thought to lie in the regression analysis performed by Matlab®.

Warning messages in the command window indicated that the values $\widehat{\omega}_i$ were too close together, thus making it difficult for the program to estimate the correlation parameter $\hat{\rho}$ adequately. It was therefore necessary to determine a new way to update the residuals estimations. A renew analysis of equation (28) suggested that this equation is valid for any *i* value. Knowing, for instance, the residuals in the ninth and tenth observation in the regression scheme, i.e. $\widehat{\omega}_9$ and $\widehat{\omega}_{10}$, as well as the disturbance estimations at the tenth observation $\hat{v}_{10}$, one would be capable to calculate $\hat{\rho}$ as $\hat{\rho} = (\widehat{\omega}_{10} - \hat{v}_{10})/\widehat{\omega}_9$. There seems not to be any privileged observation over which this formula should be applied, that is, any *i* value between *1* and *N* would work as well. Therefore, the present author decided to obtain the average over all *i*'s. These decisions do not contradict the approach taken previously for the update scheme –see equation (**36**). They also seem to fit Chatterjee and Venkateswaran (2015) approach.

Once some of the most significant changes to the program have been exposed, we will turn our attention to the duty cycle introduced in the second part of the simulations.

Buildup of the Duty Cycle

A duty cycle consists of active/wake and idle/sleep modes of the receiver, which for long strings could help minimize the energy consumed in the process. Macii et al. (2009) provided a rough estimation of power consumption during protocol synchronization in wake mode. Their results confirm that such a duty cycle could help reduce resources utilization, a schematic example of which is presented in Figure 6. It is reproduced below for reasons of interest:

Figure 31

Basic Cycle: Active and Idle Modes Distribution

Figure 31 shows that the basic cycle is divided into active and idle modes. During the active modes, the sensors receive the timestamped messages, and therefore can build up the sequences $\{y_i\}$ and $\left\{T_{1,i}^{(A)}\right\}$. Regressing the last over the former sequence –that is, applying equation (27) – $\hat{\theta}_{offset}$, $\hat{\theta}_{skew}$ and $\hat{\omega}_i$ can be obtained as the intercept, slope and residuals respectively of the linear model. As it has been stated previously, the scheme is iterated following the same steps found in Chatterjee and Venkateswaran (2015) and summarized in Table 3 above.

During the idle modes, it is first necessary to obtain the $\{\hat{y}_i\}$ estimations somehow. After several trials, it was determined that extrapolating previous $\{y_i\}$ results over the points falling within the idle mode was the best option. A different alternative considered before was to extrapolate $\hat{\theta}_{offset}$ and $\hat{\theta}_{skew}$ directly. However, it was found out that inaccurate estimates of the clock offset and clock skew obtained previously corrupted subsequent values. Hence, this option was discarded.

The simulations were run over a pattern that repeated itself every 100 cycles, that is, four times, bringing the total number of cycles to 400. 41 different schemes were tried, with sets of

active cycles ranging from 60 to 100, or equivalently idle cycles ranging from 40 to 0, to make a grand total of wake and sleep cycles to 100.

Computation Time

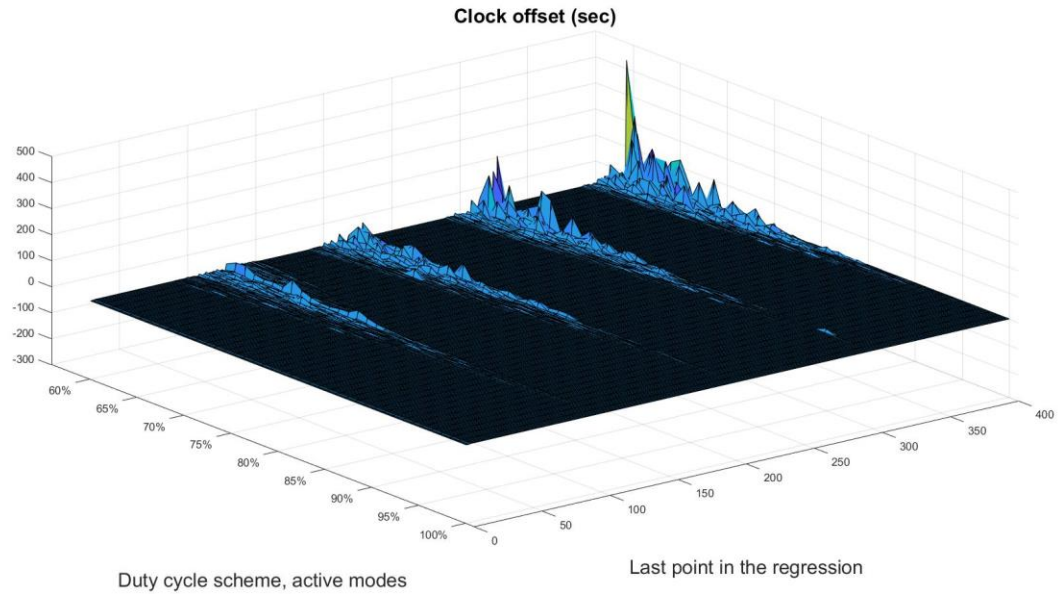
A new parameter was analyzed: the amount of time the computer spent in the evaluations. It is noted here that the program was run on a computer with a CPU of 1.80 GHz. Different speeds might be achieved with processors even with this same value, depending on their configuration. Again, the results analysis of computation times will only be qualitative.

Effects of Introducing Idle Modes

Results for the clock offset with two independent parameters, namely the length of the active period and the number of observations –points in the regression scheme– taken, are represented in the following figure:

Figure 32

Clock Offset with the Duty Cycle Effect



Note. The figure has been obtained averaging over 100 iterations.

Figure 32 shows two areas of time offset estimations: The flat areas and the peaks, in the form of mountain ranges, at given values of the last observation in the regression scheme. The flat areas conform quite well with the true value of 1.45. On the contrary, in the "mountain ranges" there are very significant excursions of up to more than 400 sec. To understand why these ranges are created, we must first look at how the algorithm was constructed:

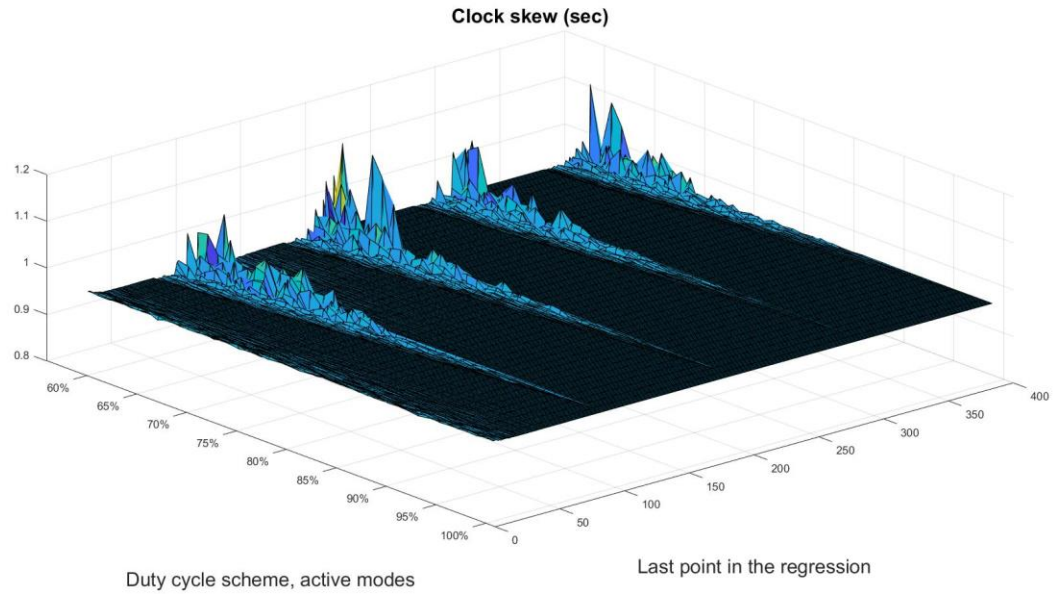
Since regression consists of a linearization over a set of observations, it would not work with a single value. For stability purposes, it was decided that a value of 4 was a minimum to start iterating. Observe that for a number of active cycles close to 100%, values are quite flat – and close to the true value of 1.45, as it has been remarked earlier. However, as the idle cycles size grows, some peaks begin to appear, in the form of outliers. This effect is all the more

remarkable as the size of idle cycles increase, up to a maximum value of 40% taken in the simulations. Also, note that as the idle cycles grow in number, the "mountain range" areas breadth also increases, i.e., more values in the x axis are affected. Remember that the x axis represents the last observation in the regression. This effect can be explained by the way the algorithm was constructed. Take, for instance, the case of a duty cycle made of 75% of active modes and 25% of idle modes. In this particular case, the first 75 values are taken in the active mode, thus, y_i and $\left\{T_{1,i}^{(A)}\right\}$ for *i*=4…75 are obtained, not estimated –see equation (27)or (29). By the time when the 76th value arrives, the receiver is shut off. Hence, y_{76} must be extrapolated from previous results. The same happens with the $T_{1,76}^{(A)}$ value. Indeed, the program needs to extrapolate the 25 values between 76 and 100, before a second active mode starts again. As the number in the idle cycle grow, so grows the number of extrapolations to obtain their corresponding y_i and $\left\{T_{1,i}^{(A)}\right\}$ values. Since the extrapolations are less accurate than the values obtained directly, the estimations of clock offset and clock skew worsen.

A similar rationale can be used to explain a similar pattern for the clock skew, shown in Figure 33:

Figure 33

Clock Skew with the Duty Cycle Effect



Note. The first 10 values of the last point in the regression –x axis– were outliers and have been removed. The figure has been obtained averaging over 100 iterations.

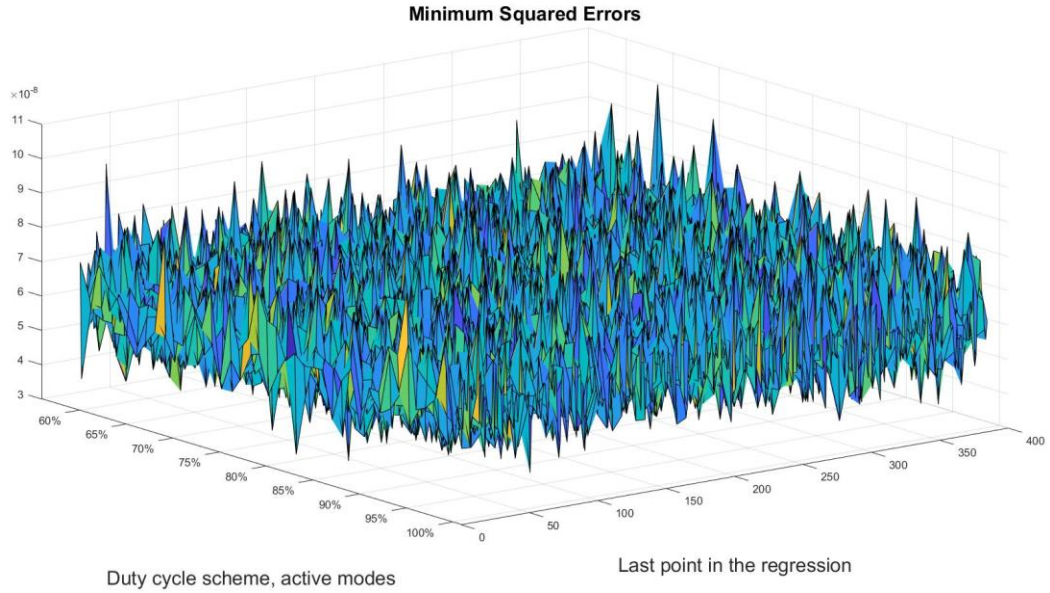
A qualitative comparison between the range of the outliers in the clock skew (Figure 33) with the outliers in the clock offset (Figure 32) suggests that the clock skew has a lower standard deviation than the clock offset. This fact should not come as a surprise if we analyze the nature of equation (29) in detail. In this equation, $\theta_{skew}^{(BP)}$ represents the slope of a straight line, whereas the clock offset $\theta_{offset}^{(BP)}$ is represented by the intercept. However, the intercept is also influenced by the set of residuals $\{\hat{\omega}_i\}$, whose real values need also be estimated through the regression parameter $\hat{\rho}$; see equation (28). Moreover, equation (28) shows that the residuals are not i.i.d. variables of zero mean, since they are autocorrelated with one-lag delay. This feature makes the

simultaneous estimations of $\hat{\theta}_{offset}^{(BP)}$ and the set of residuals $\{\hat{\omega}_i\}$ to be very sensitive to each other's estimation. As it was stated at the beginning of the chapter, the recursive scheme was iterated until the difference in two successive estimations of $\hat{\rho}$ was less than .0001. Unfortunately, this does not mean that the estimation was unbiased. Here lies the main weakness in the approach to solve a time synchronization using autocorrelated noise values –represented by the residuals $\{\hat{\omega}_i\}$. The author thinks that this problem cannot be solved in a better manner, at least with the scheme adopted here from Chatterjee and Venkateswaran (2015), and first proposed by Cochrane and Orcutt (1949).

A further proof that clock offset and residuals influence each other can be observed in the following figure:

Figure 34

Minimum Squared Error of the $\hat{y}_i$ Estimations



Note. The first 10 values of the last point in the regression were outliers and have been removed. The figure has been obtained averaging over 100 iterations.

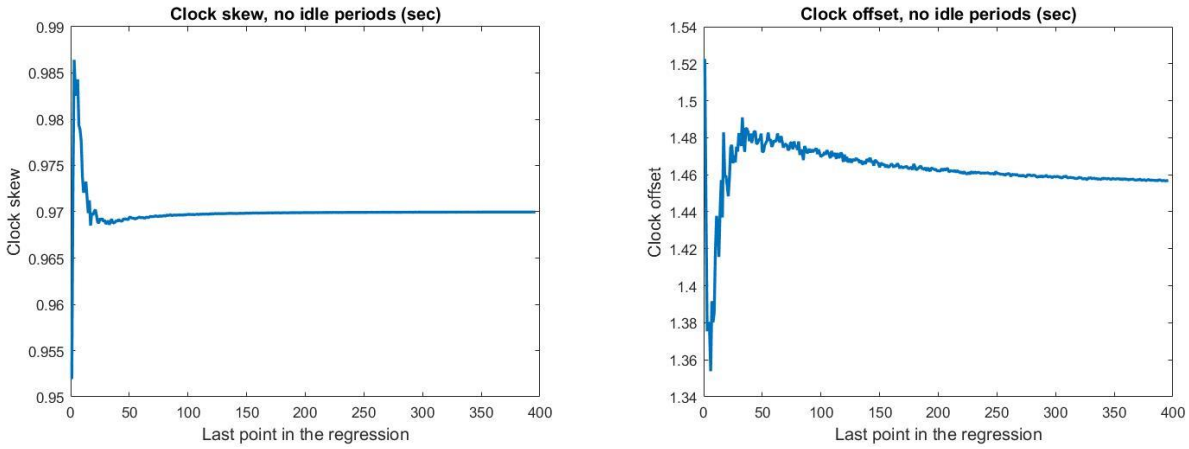
The low MSE values of $\{y_i\}$ estimations, of the order of 10^{-8}, across any combination of duty cycles and last points in Figure 34 would not be understood if only Figure 32 and particularly Figure 33 were considered. Indeed, the "mountain ranges" of Figure 32 and Figure 33 completely disappear in Figure 34. Again, the nature of equation (29) may help explain the results: The tradeoff between $\hat{\theta}_{offset}^{(BP)}$ and $\{\hat{\omega}_i\}$ balance out the inaccuracies in both terms, and the estimates of $\{\hat{y}_i\}$ are quite close to the real values $\{y_i\}$, as can be observed in Figure 34. The errors in Figure 34 are of the order of 10^{-8}, whereas the same errors in Figure 11 are of the order of 10^{-4}. Results are coherent, with only remembering that in the former case, the recursive scheme was iterated until a difference of .0001 of two successive values of $\hat{\rho}$ was achieved.

However, in the last case, the number of iterations was previously fixed, and the loops stopped no matter what that difference was.

The same difference may help explain the results of clock offset and clock skew where no idle modes were introduced. In this respect, compare Figure 8 and Figure 10 with Figure 35:

Figure 35

Clock Skew and Clock Offset with no Idle Modes

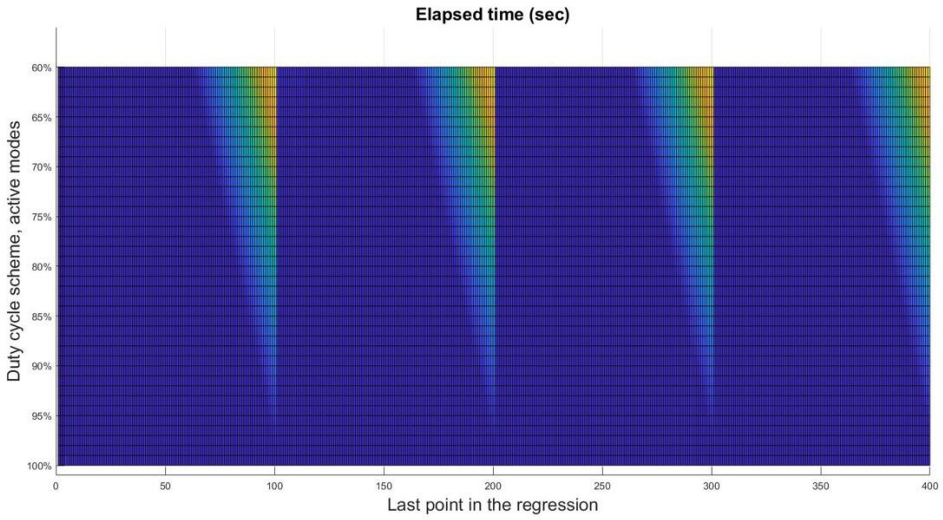


Note. The first 10 points, which are outliers, have been removed.

It would be interesting to analyze the pattern of the time spent by the program to converge to the estimated $\hat{\rho}$ value. This is represented in Figure 36 below:

Figure 36

Time Elapsed During Estimations



Note. Blue hues represent low values, whereas yellowish values represent high values of the elapsed time. In the simulations performed, any combination of the duty cycle scheme and the last point in the regression scheme did not have an elapsed time greater than 0.27 sec. The figure has been obtained averaging over 100 iterations.

The first thing to note when observing Figure 36 is that the plains/peaks range pattern is repeated here. As in Figure 32 and Figure 33, these “ranges” broaden as we increase the size of the concatenated idle periods, from 0% to 40% --or, equivalently, as we reduce the active periods from 100% to 60%. On the contrary, there does not seem to be a difference in the addition of regression points.

If we looked at the evolution of clock offset and clock skew with a fixed number of points of 400 –Figure 38 and Figure 39–, we would obtain the following graphics for the clock offset and clock skew:

Figure 37

Clock Skew vs. Number of Idle Modes in the Duty Cycle Scheme

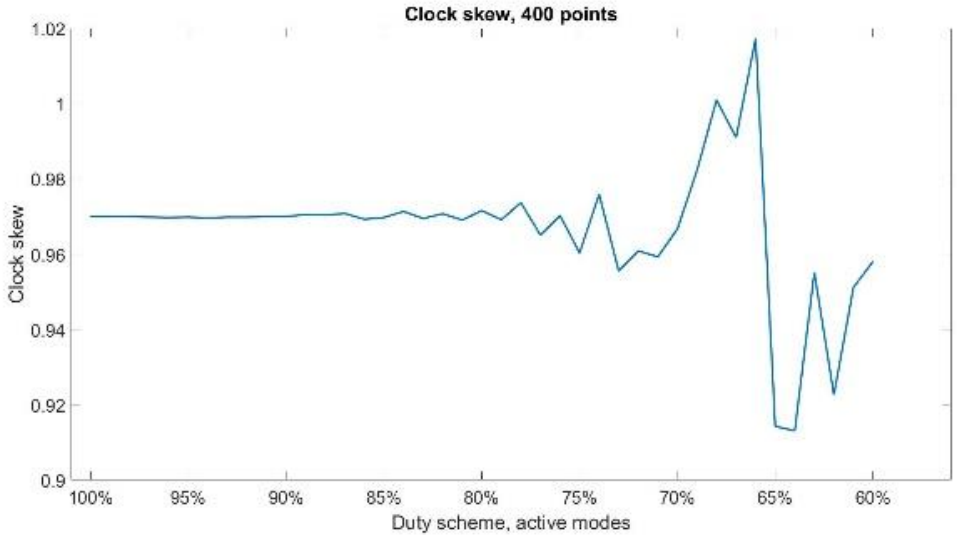
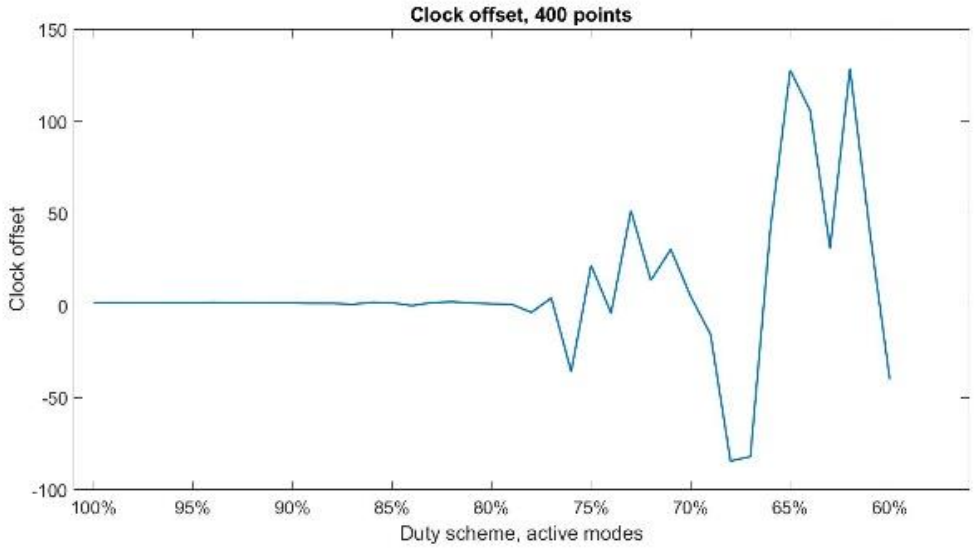


Figure 38

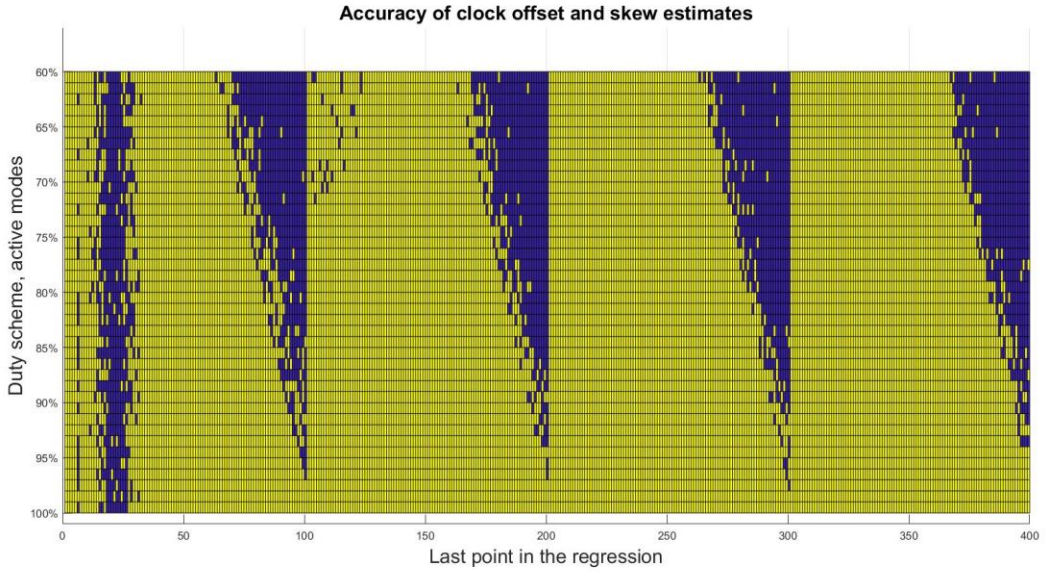
Clock Offset vs. Number of Idle Modes in the Duty Cycle Scheme



As we can observe in Figure 38 and Figure 39, estimations get worse as idle modes are increasingly introduced into the duty cycle. A t-test was run to obtain the duty cycle with the highest number of idle modes which could deliver estimates of clock offset and clock skew reasonably close to the true values, $\theta_{offset} = 1.45$ and $\theta_{skew} = 0.97$, respectively. The null hypothesis H_0 was: "The clock offset and clock skew data are normally distributed with averages of 1.45 and 0.97, respectively". The significance level was 5%. Results are presented in Figure 40.

Figure 39

Accuracy of Clock Offset and Skew Estimates



Note. T-test results at .05 significance level to determine whether clock offset and clock skew estimates are reasonably close to 1.45 and 0.97, respectively. H_o would confirm this hypothesis. Yellow cells represent combinations of duty cycle and observations which confirm H_o. Blue cells, on the contrary, reject H_o at .05 level of significance.

Figure 40 shows that the estimations differ from the true values significantly in the "mountain range" areas. Also, it is possible that as more observations are added to the regression, the scheme is more robust against idle modes. However, this could only be proven with regression schemes beyond the 400 observations, which this dissertation did not investigate. At this limit of 400 observations, Figure 40 shows that good approximations could be made with as much as 8% of idle modes introduced in the duty cycle. Compare this result with Figure 37 and Figure 39: Although these figures suggest that a duty cycle of up to about 20% of idle modes might still deliver good estimates, more accurate investigations involving t-tests show that the number of idle modes must be much lower.

Effects of Noise Levels

In this section, the limit of 8% of idle modes will be retained with a 400-observations scheme; i.e., $\left\{T_{1,i}^{(A)}\right\}_{1}^{400}$. The objective was to analyze the impact of varying the noise level of the residuals $\{\omega_i\}$ and the disturbances $\{\nu_i\}$ on the initial results. To this end, noise levels were made to vary between -3dB and 0dB with respect to the signal. The other parameters stayed as presented in Table 6. The distributions of residuals and disturbances were normal with 0 mean.

The following figures presents the clock offset and clock skew estimations with different noise levels:

Figure 40

Clock Offset Estimations with Different Noise Levels

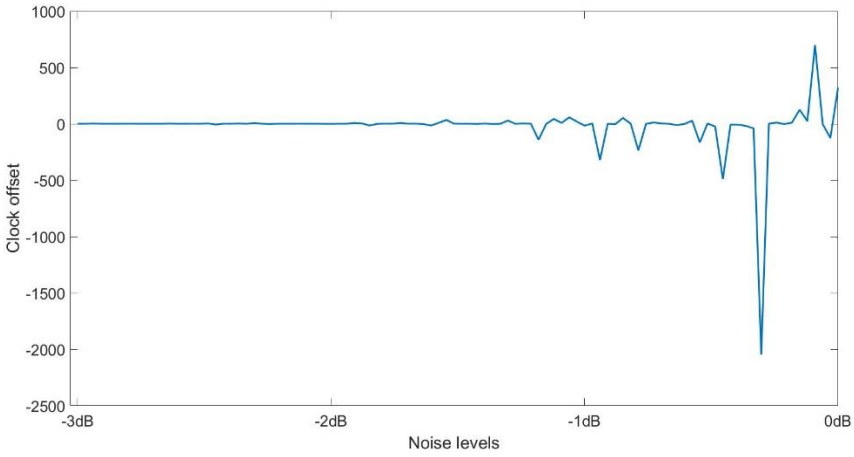
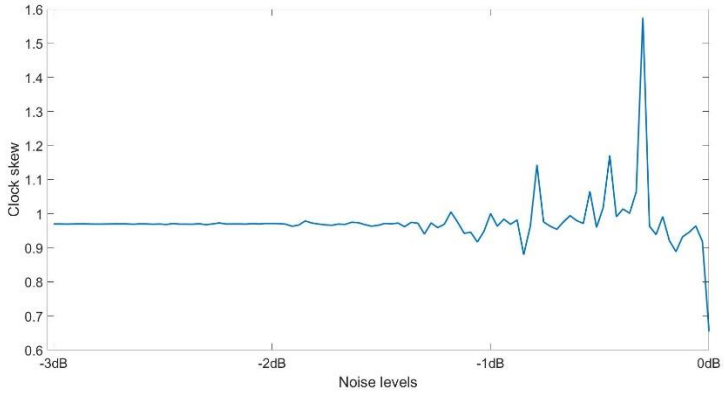


Figure 41

Clock Skew Estimations with Different Noise Levels



It was observed that the estimations below a -2dB noise level were quite acceptable.

VI. CONCLUSIONS AND FUTURE WORK

This last chapter summarizes the main conclusions obtained through the simulations performed above. These conclusions will help respond to the research questions introduced in the first chapter. The outline of three proposed future tests checking the key assumptions of this model and some thoughts on the use of 5G in aviation bring the present dissertation to its closure.

Results Summary and Research Questions Addressed

The main important results from the tests, from which the answers to the research questions at the end of the first chapter will be elicited, are presented below:

- When considering zero-mean distribution of disturbances (i.e., the gaussian model used in the first part of the tests) the model shows convergence with several iterations as low as 4 (see Figure 8 and Figure 10). Further iterations did not make any significant difference on the results. However, it also shows that the number of observations –of about 250, see Figure 7 and Figure 9– is much higher than other models. Compare this last number with Lee and Chin's (2016) results, using an exponential distribution of errors with a mean which is very close to zero ($3.33\ \mu\text{sec}$). In their case, MSE for clock offset and clock skew was achieved at the respective order of 10^{-6} and 10^{-15} with less than 25 observations.
- The variation in the estimations results is sufficiently close to the CRB limit to safely conclude that, despite the high number of observations needed, the

estimation is fully efficient (see Figure 12 and Figure 13). Moreover, the variation in the prediction decreases as the number of observations increase (see Figure 14).

- As the noise becomes highly correlated, i.e., as the correlation parameter $\rho \to 1$, the MSE grows drastically (see Figure 18). This result suggests that the scheme might not be appropriate in multipath environments, such as urban areas, where reflecting surfaces creates strong signal echoes.
- The case of non-zero mean distributions for the noise is interesting: Results that use exponential-, gamma- and Weibull-distributed noise models are all biased, but the MSE decreases with the number of observations. This is in line with Chaudhari et al. (2010), who showed this effect in the exponential case, for which they concluded that the MSE of the clock offset estimator is inversely proportional to the square of the number of observations. We could conclude that more observations added to the scheme result in more accurate estimations. Whether the distribution is or is not zero-mean does not seem to be relevant: Compare the same trend in Figure 11 for a gaussian distribution (which is also zero-mean, with Figure 22) which shows an exponential distribution.
- The tests performed showed that the scheme is inaccurate with non-zero mean disturbances. If we review the assumptions in the model, and equation (30) in particular, we see that the means of the disturbances may be absorbed in the clock offset term, thus increasing the difference between the estimations $\hat{\theta}_{offset}$ and the true value θ_{offset}. This is not the case for the clock skew, the slope term in the model. We observed this effect by comparing the results between clock offset and clock skew in the three cases (i.e., exponential, gamma and Weibull). We thereby

confirmed that estimations are still good for the clock skew or, in other words, they are not as much affected by the disturbances' mean as clock offset estimations are.

- Different autocorrelation schemes from the one-lagged presented in equation (9) have been tested. Whereas biparametrical autocorrelation worsens the initial results (see Figure 28), two-lagged autocorrelation schemes showed to be more efficient (as seen in Figure 29). This could be explained by the fact that, in the first case, the estimations of both autocorrelation parameters were done simultaneously, thus increasing the error in their estimation. It is not clear to which real scenarios these schemes might correspond, but it might be the case that two-lagged autocorrelated noise manifests in urban environments, where multipath effects are present.
- Four schemes to update the residuals' estimations were tested (see equations (47) to (50), represented in Figure 31). The first of them delivered best results, probably because this is the scheme with less accumulated errors after iterating: Observe equation (**47**), which shows that the disturbances' estimations are subtracted from the first residuals' estimation in each iteration and are therefore not influenced from previous steps.

In the second part of the tests a duty cycle was added to the initial scheme. A summary of the results is provided below:

- Estimations were acceptable far-off from the idle modes. Figure 32 and Figure 33 showed a typical mountain range-shape around the idle modes, which got coarser with the length of the idle period. These results suggest that idle periods, although

useful to save resources, should not be abused: As Figure 32 and Figure 33 show, their impact on the estimations is important.

- Figure 37 showed that the elapsed computation time to obtain the estimations grew significantly during idle periods, which means that convergence to the real values was slower. The effects begin to manifest when more than 5% of the total number of periods are idle –see results of a t-test with $p = .05$ in Figure 40–, but they become increasingly predominant when this percentage reaches 25% of the total duty cycle.
- An additional result from the tests is related to the noise level. Remember that, in this dissertation, this level does not refer to power, but to a number which multiplies the residuals, thus making them smaller –low variability– or larger – high variability of the results. As the noise level increases, i.e. echoes of precedent signals are present with an effect of delaying the signal reception, estimations become less accurate. Figure 40 and Figure 42 show that the scheme could have a good behavior up to a "noise level" of -2dB.

The following three research questions were stated at the end of chapter 1, to which an answer is presented using these research findings:

1. Which parameters characterize a recursive algorithm that estimates the clock offset and clock skew over a ROS synchronization protocol using a duty cycle? The number of observations is a key parameter in the model proposed, and low variability of results, with respect to the CRB value, may be achieved with a considerable number of these (above 250, approximately). On the contrary, the

scheme proposed converges very fast, within as few as 4 iterations, and we can safely state that their influence is not critical (see Figure 7 and Figure 9).

Two-lagged autocorrelation schemes seem to be as efficient as in the one-lagged case. Also, estimates updates in the form showed by equation (**47**) are most accurate, compared to the rest of the schemes; equations (**48**) through (**50**).

2. Which of these parameters' values provide an optimum result in terms of accuracy and resources utilization?

   Whereas accuracy discussions are covered in the previous question, our focus of attention now should be on the wake/sleep scheme adopted to save resources. Results showed that such schemes grow increasingly inaccurate with the number of idle periods, and effects become manifest with as low as a 95/5 proportion (see Figure 32 and Figure 33).

3. How does the choice of the noise model affect the efficiency of the selected scheme?

   Non-zero mean models used in the present analysis (i.e., exponential, gamma and Weibull distributions) are not suitable for the scheme proposed. Whereas skew estimations are not as much affected by the noise model proposed, offset estimations are. Although these estimations grow increasingly accurate with the number of observations, results are never as satisfactory as in the gaussian model case. Existing literature suggest that gamma and Weibull noise distributions are well correlated in multipath environments, whereas exponential distributions might be good to simulate communications beyond the visual line of sight. Hence, the scheme proposed might not be good for any of these environments.

In order to confirm the results obtained in the present dissertation, some real tests could be performed in the future. The next section describes the rationale for three of them.

Proposal for Future Research: Three Real Tests

This section provides an outline of three tests that could be performed to further analyze the behavior of the synchronization protocol scheme analyzed in this dissertation.

The Baseline Test

The baseline should be established within line of sight, in an open, rural space, so that the chances of multipath effects are minimized. A variation of the scheme with a two-lagged noise distribution could also be examined here –see equation (45)– since good theoretical results were obtained above (see Figure 29). It is possible that, to achieve an adequate noise correlation, the ground antenna should broadcast signals at a high rate, the limit being the drone's processor speed.

BVLOS Communications

As indicated above, existing literature suggests that exponential noise distributions might provide a good correlation in BVLOS environments. Despite having a non-zero mean, exponential distributions could be used to analyze the efficiency of the scheme proposed. Figure 21 and Figure 22 show that skew estimates might still provide good results but offset estimates might become increasingly differentiated from the true value.

Wake/Sleep Schemes

Finally, wake/sleep schemes could be introduced to prove the theoretical results presented in the second part of this dissertation. These tests could further analyze two aspects related to the use of such schemes: the accuracy of the estimates when different schemes are used, that might help prove the 95/5 limit encountered in the numerical analysis of this dissertation; and the effect on the processing time. Indeed, one of the surprising results was related to the time needed for convergence during idle periods. It would be interesting to analyze this phenomenon and, if the tests confirm the results, propose new ways to reduce such times, thereby saving energy resources.

As it was stated in the first chapter to this dissertation, 5G standards provide an opportunity to increase the speed of communications and hence the amount of information payload that can be transmitted per unit time. It would also increase the accuracy of any measurement made, and the prospects of reducing communications latency is very promising. However, some airspace users have presently raised some concerns on the use of 5G near airports and in areas that might affect air navigation and surveillance systems. Because an accurate time synchronization protocol could reveal itself as a very efficient means to keep pace with the speed expected in 5G, some initial analyses of these concerns are of necessity here. They are provided as concluding reflections to the present chapter and to the dissertation as a whole.

Some Final Thoughts on the Use of 5G in Aviation

Clocks slay time... time is dead as long as it is being clicked off by little wheels; only when the clock stops does time come to life. --William Faulkner, *The Sound and the Fury*

Starting on September 9, 2020, the Federal Communications Commission (FCC) auctioned off the operation in the 3.7 GHz band (Federal Communications Commission, n.d.), which is foreseen to be used for the implementation of 5G communication standards. This band is made up of four blocks, namely A, B and C, plus a guard band which did not enter the auction process. With a length of 80 megahertz spanning from 3,980-4,000 MHz, the C block was purchased by ATT and Verizon.

Meanwhile, the Federal Aviation Agency, indicated that the use of these frequencies near airports might interfere with certain aircraft safety equipment (Federal Aviation Administration, 2021). They came up with two solutions aimed at mitigating the impact: First, either retrofitting or replacing radar altimeters that turned out to be not robust enough against 5G interference. Indeed, the recent clearing of two radio altimeter models that accounted for most of Airbus and Boeing aircraft, allowed to minimize the impact of such a measure (Clark, 2022). Second, the telecommunication companies agreed to create buffer zones in airports where close transmitters could be a source of interference for a period of six months and delay their deployment until January 19. With the creation of these buffer zones, nearly 45% of U.S. commercial fleet would be allowed to perform low-visibility landings, whenever the use of radio-altimeters is needed.

However, the same FAA sources show that less restrictive requirements have been applied in other countries such as France. This country has made communication companies to adopt certain measures to mitigate interference with third parties, among which:

1) The power levels of 5G antennas in France are lower than in the U.S.
2) 5G frequencies are located farther away from the frequency bands used by radar altimeters.
3) French antennas are tilted down, thereby providing less harmful interference to their surroundings.

At the time of this writing, some of the major U.S. airlines have asked the administration for immediate intervention on the issue (Muntean & Wallace, 2022), and the communication companies have already agreed to delay the 5G rollout nearby airports. In any case, promising results elsewhere suggest that a careful handling of the issue might guarantee that the aviation community is protected against any negative impact.

If we consider the advantages of 5G standard (namely, as this dissertation highlights, the speed and lower latency in communications), we can conclude that the research of systems or techniques to minimize harmful effects is worth the effort. It is the author's opinion that the aviation system may greatly benefit from the 5G revolution, which may have a sweeping impact in drones' usage by reducing resources consumption and achieving a higher-quality payload delivery. Certainly, all this could not be achieved without major improvements in the way communications are established, and an enhanced synchronization protocol may play a key role here. In proposing such a system, it was this author's intention to contribute to the discussion at the crossroads of software engineering, communications and avionics.

In retrospective, all this effort might not have been necessary if clock offsets were not an issue. Unfortunately, the reality shows that quartz crystals are far from perfect, and the problem of synchronizing two systems set apart at a given distance with no physical means between them

will always persist. Lest, as Faulkner's quote at the beginning of this section suggests, ticking time was made with more perfect, although less fascinating, devices. In such an ideal world, the search for a more efficient time synchronization protocol would not have been worth the effort, and the present author could well have picked up a different topic for his dissertation. Fortunately, this has not been the case, as the learnings have far outpaced the effort to reach this end.

Appendix

This section presents a scheme of the Matlab® program used in the tests. The script was slightly modified to obtain specific results in every part. The main script calls six different functions, which were built for the tests from scratch:

- *initial_values*: The initial values of the model, such as the total –first level– number of iterations or the recursive algorithm's –second level– number of iterations were specified.
- *timestamps*: The timestamps of the model $\left\{T_{1,i}^{(BP)}\right\}$ were generated, and the values $\{y_i\}$, $\{v_i\}$ and $\{\omega_i\}$ obtained –see equation (29).
- *firstwrhoestimate*: The first estimates of $\{\widehat{w}_i\}$ and $\hat{\rho}$ in the recursive algorithm were obtained, e.g. using *mldivide*.
- *starvalues*: The values of $\{\hat{v}_i\}$, $\hat{\theta}_{offset}^{*}$ and $\hat{\theta}_{skew}^{*}$ in the recursive algorithm were obtained –see equation (30).
- *nextwrhoestimate*: Successive values of $\{\widehat{w}_i\}$ and $\hat{\rho}$ were obtained. They were in turn used in a new loop over the *starvalues* function.
- *evaluation*: Once two successive estimations of $\hat{\rho}$ were close enough, or the number of second level iterations was reached, the loop was terminated and the program used the values of $\hat{\theta}_{offset}^{*}$, $\hat{\theta}_{skew}^{*}$ and $\{\widehat{\omega}_i\}$ estimates and obtained $\hat{\theta}_{offset}$ –see equation (**12**)–, $\hat{\theta}_{skew} = \hat{\theta}_{skew}^{*}$ and $(y_i - \hat{y}_i)$ in the MSE sense, see equation (1).

References

Abedi, O., & Pourhasani, A. (2021). Prioritized multi-channel MAC protocol in ad hoc networks using a TDMA/CSMA approach. *Wireless Networks, 27*, 2629-2640. https://doi.org/10.1007/s11276-021-02581-7

Ahmad, A., Noor, A., Serpedin, E., Nounou, H. & Nounou, M. (2010). On clock offset estimation in wireless sensor networks with Weibull distributed network delays. *2010 20th International Conference on Pattern Recognition*, 2322-2325. https://doi.org/10.1109/ICPR.2010.568

Akila, V., Sheela, T., & Macriga, G. A. (2017). Efficient packet scheduling technique for data merging in wireless sensor networks. *China Communications*, *14*(4), 35–46. https://doi.org /10.1109/CC.2017.7927575

Balsi, M., Prem, S., Williame, K., Teboul, D., Délétraz, L., & Capdeville, P. I. H. (2019). Establishing new foundations for the use of remotely-piloted aircraft systems for civilian applications. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42*(2), 197-201. https://doi.org/ 10.5194/isprs-archives-XLII-2-W13-197-2019

Bartsch, R. I., Coyne J., & Gray, K. (2017). *Drones in society: Exploring the strange new world of unmanned aircraft*. Routledge.

Bottom, V. E. (1982). *Introduction to quartz crystal unit design*. Van Nostrand Reinhold Company.

Buck, J. (1988). Synchronous rhythmic flashing of fireflies. II. *The Quarterly Review of Biology, 63*(3), 265-289. https://doi/org/10.1086/415929

Caraballo, T., El Fatini, M., El Khalifi, M., Gerlach, R., & Pettersson, R. (2020). Analysis of a stochastic distributed delay epidemic model with relapse and Gamma distribution kernel. *Chaos, solitons and fractals, 133*, 1-8. https://doi.org/10.1016/j.chaos.2020.109643

Chatterjee, A. & Venkateswaran, P. (2015). An efficient statistical approach for time synchronization in wireless sensor networks. *International journal of communication systems, 29*, 722-733. https://doi.org/10.1002/dac.2944

Chaudhari, Q. M., Serpedin, E., & Kim, J-S. (2010). Energy-efficient estimation of clock offset for inactive nodes in wireless sensor networks. *IEEE Transactions on Information Theory, 56*(1), 582-596. https://doi.org/10.1109/TIT.2009.2034817

Chelouah, L., Semchedine, F., & Bouallouche-Medjkoune, L. (2018). Localization protocols for mobile wireless sensor networks: A survey. *Computers and electrical engineering, 71*. 733-751. https://dx.doi.org/10.1016/j.compeleceng.2017.03.024

Chen, R., & Cassandras, C. G. (2018). Stochastic flow models with delays and applications to multi-intersection traffic light control. *IFAC papers online, 51*(7), 39-44. https://doi.org/10.1016/j.ifacol.2018.06.276

Chen, X., Morton, Y. J., & Yu, W. (2020). GPS L1CA/BDS B1I multipath channel measurements and modeling for dynamic land vehicle in Shangai dense urban area. *IEEE Transactions on Vehicular Technology, 69*(12), 14247-14263. https://doi.org/10.1109/TVT.2020.3038646

Clark, M. (2022, January 20). *FAA estimates 78 percent of US planes can now land at airports with 5G C-band*. The Verge. https://www.theverge.com/2022/1/20/22893597/faa-5g-c-band-rollout-airports-cleared-altimeters-regional-jets

Cochrane, D. & Orcutt, G. H. (1949). Application of least squares regression to relationships containing auto-correlated error terms. *Journal of the American Statistical Association, 44*(245), 32-61. https://www.jstor.org/stable/2280349

Cramér, H. (1999). *Mathematical methods of statistics*. Princeton University Press.

Deng, M. (2019). Stability of a stochastic delay commensalism model with Lévy jumps. *Physica A, 527*, 1-10. https://doi.org/10.1016/j.physa.2019.121061

Dielman, T. E. (1985). Regression forecasts when disturbances are autocorrelated. *Journal of forecasting, 4*(3), 263-271. https://doi.org/10.1002/for.3980040303

Durbin, J., & Watson, G. S. (1950). *Testing for Serial Correlation in Least Squares Regression: I*. 21. *Biometrika, 37*(3/4), 409-428. https://www.jstor.org/stable/2332391

Durbin, J., & Watson, G. S. (1951). *Testing for Serial Correlation in Least Squares Regression. II*. 20. *Biometrika, 38*(1/2), 159-177. https://www.jstor.org/stable/2332325

Eier, D. & Sharples, M. (2019). Method for GPS and GNSS independent MLAT system synchronization. *2019 Integrated Communications, Navigation and Surveillance (ICNS) Conference*, 1-27. https://doi.org/10.1109/ICNSURV.2019.8735366

Elson, J., Girod, L. & Estrin, D. (2002). *Fine-grained network time synchronization using reference broadcasts*. Proceedings of the 5th Symposium on Operating System Design and Implementation. (pp. 147-163). USENIX Association. https://doi.org/10.1145/844128.844143

Ergen, S. C. & Varaiya, P. (2009). TDMA scheduling algorithms for wireless sensor networks. *Wireless networks, 16*. 985-997. https://doi.org/10.1007/s11276-009-0183-0

Faizulkhakov, A. (2007). Time synchronization methods for wireless sensor networks: a survey. *Programming and computer software, 33*(4), 214-226. https://doi.org/10.1134/S0361768807040044

Federal Aviation Administration Modernization and Reform Act 2012, § 331-332 (2011). https://www.congress.gov/bill/112th-congress/house-bill/658/text

Federal Aviation Administration Order No. 8040.6, C.F.R. 14 (2019). https://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.information/documentID/1036752

Federal Aviation Administration. (2021, May 4). *UAS by the numbers*. https://www.faa.gov/uas/resources/by_the_numbers/

Federal Aviation Administration (2021, November 2). *Special Airworthiness Information Bulletin: Risk of Potential Adverse Effects on Radio Altimeters*. Federal Aviation Administration. https://rgl.faa.gov/Regulatory_and_Guidance_Library/rgSAIB.nsf/dc7bd4f27e5f107486257221005f069d/27ffcbb45e6157e9862587810044ad19/%24FILE/AIR-21-18.pdf

Federal Aviation Administration (n.d.) Satellite Navigation- Global Positioning System (GPS). Retrieved January 27, 2022, from https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps#:~:text=The%20Global%20Positioning%20System%20(GPS,used%20for%20monitoring%20and%20control.&text=GPS%20is%20operated%20and%20maintained,Department%20of%20Defense%20(DoD)6

Federal Communications Commission (2017, April 14). *Open for business: FCC's new experimental licensing system accepting new applications*. https://www.fcc.gov/news-events/blog/2017/04/14/open-business-fccs-new-experimental-licensing-system-accepting-new

Federal Communications Commission. (July 15, 2021a). *Innovation zones. Public notice, ET Docket no. 19-257*. https://docs.fcc.gov/public/attachments/DOC-374118A1.pdf

Federal Communications Commission. (January 14, 2021b). *Order DA 21-58*. https://docs.fcc.gov/public/attachments/FCC-21-58A1.pdf

Federal Communications Commission. (August 25, 2021c). Report No. 3181. Retrieved from https://ecfsapi.fcc.gov/file/082591916582/DOC-375207A1.pdf

Federal Communications Commission (August 20, 2021d). *Wireless telecommunications bureau seeks to refresh the record on unmanned aircraft systems use of the 5 GHz band*. Retrieved from https://www.jdsupra.com/legalnews/alert-fcc-seeks-comment-on-radio-9876830/

Federal Communications Commission (n.d.) *Auction 107: 3.7 GHz Service*. https://fcc.gov/auction/107/factsheet

Fletcher, B. (June 11, 2019). *AT&T, Uber collaborate on 4G and 5G connectivity for aerial ridesharing, cargo drones*. 5G Technology World. Retrieved November 4, 2021, from https://www.5gtechnologyworld.com/att-uber-collaborate-on-4g-and-5g-connectivity-for-aerial-ridesharing-cargo-drones/

Freris, N. M., Graham, S. R., Kumar, P. R. (2011). Fundamental limits on synchronizing clocks over networks. *IEEE transactions on automatic control, 56*(6), 1352-1364. https://doi.org/10.1109/TAC.2010.20892210

Frerking, M. E. (1978). *Crystal oscillator design and temperature compensation*. Van Nostrand Reinhold Company.

Global Positioning System. (2019). *Timing*. Retrieved June 7, 2021, from https://www.gps.gov/applications/timing

Harris, M. (2021). Lost in airspace. Military tests that jam and spoof GPS signals are an accident waiting to happen. *Spectrum IEEE.org, 58*(2), 22-27. https://doi.org/10.1109/MSPEC.2021.9340116

Hildreth, C. & Lu, J. Y. (1969). *A Monte Carlo study of the regression model with autocorrelated disturbances, RM-5728-PR*. The RAND Corporation.

Huerta v. Pirker, Docket CP-217. (2014). https://www.ntsb.gov/legal/alj/Documents /5730.pdf

Institute of Electrical and Electronics Engineers. (2019). *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems* (IEEE Std 1588™-2019). https://ieeexplore-ieee-org.ezproxy.library.und.edu/stamp/stamp.jsp?tp=&arnumber=9120376

Iordache, A. V., & Marghescu, I. (2012, November 15-16). *LTE network time synchronization.* 10th International Symposium on Electronics and Telecommunications, 159-162. https://doi.org/ 10.1109/ISETC.2012.6408099

Ivancic, W. D., Kercewski, R. J., Murawski, R. W., Matheou, K., & Downey, A. D. (2019). *Flying drones beyond visual line of sight using 4G LTE: Issues and concerns*. 2019 Integrated Communications Navigation and Surveillance (ICNS) Conference, 1-26. https://doi.org/10.1109/ICNSURV.2019.8735278

Kim, J-S, Lee, J., Serpedin, E., Qaraqe, K. (2009). A robust estimation scheme for clock phase offsets in wireless sensor networks in the presence of non-gaussian random delays. *Signal processing, 89*, 1151-1161. https://doi.org/10.1016/j.sigpro.2008.12.021

Kumar, A., Namboothiri, P. G., Deshpande, S., Vidhyadharan, S., Sivalingam, K. M., & Murty, S.A.V. S. (2012). Testbed based throughput analysis in a wireless sensor network. *2012 National Conference on Communications (NCC)*, 1-5. https://doi.org/101109/NCC .2012.6176804

Lamonaca, F., Carnì, D. L., Riccio, M., Grimaldi, D. (2017). Preserving synchronization accuracy from the plug-in of nonsynchronized nodes in a wireless sensor network. *IEEE transactions on instrumentation and measurement, 66*(5), 1058-1066. https://doi.org/10.1109 /TIM.2017.2664422

Lee, Y-R., & Chin, W-L. (2016). Low-complexity time synchronization for energy-constrained wireless sensor networks: Dual-clock delayed-message approach. *Peer-to-peer networking & applications, 10*(4), 887-896. https://doi.org/10.1007/s12083-016-0437-4

Luo, B., Cheng, L. & Wu, Y-C. (2016). Fully distributed clock synchronization in wireless sensor networks under Exponential delays. *Signal processing, 125*, 261-273. https:// dx.doi.org/10.1016/j.sigpro.2016.02.007

Lyu, L., Chen, C., Hua, C., Zhu, S., & Guan, X. (2017). Co-design of stabilisation and transmission scheduling for wireless control systems. *IET Control Theory & Applications*, *11*(11), 1767–1778. https://doi.org/10.1049/iet-cta.2016.0871

Macii, D., Ageev, A., & Somov, A. (2009, 5-7 May). *Power consumption in wireless sensor networks through optimal synchronization* [Conference session]. 12MTC 2009-International Instrumentation and Measurement Technology Conference, Singapore. https://doi.org/10.1109/IMTC.2009.5168665

Marinelli, M., Caggiani, L., Ottomanelli, M., & Dell'Orco, M. (2018). En route truck-drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems, 12*(4), 253-261. https://doi.org/10.1049/iet-its.2017.0227

Mathworks (n.d.a) *mldivide* https://www.mathworks.com/help/matlab/ref/mldivide.html?|searchHighlight=mldivide&s_tid=srchtitle_mldivide_1

Mathworks (n.d.b) *fitlm* https://www.mathworks.com/help/stats/fitlm.html?searchHighlight=fitlm&s_tid=srchtitle_fitlm_1

Meera, G.S., Gupta, V, Sekhar, S. P., Sreejith, V., & Anupama, K. M. (2017). An efficient mobile sink routing in wireless sensor network using dynamic Steiner tree. *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 1-6. https://doi.org/101109/ANTS.2016.7947869

Mills, D. L. (1991). Internet time synchronization: the Network Time Protocol. IEEE Transactions on communications, 39(10), 1482-1493. https://doi.org/10.1109/26.103043

Moody, J. (2021, June 24). *Skyward and Federal Aviation Administration to test cellular-connected drones*. News Center. https://www.verizon.com/about/news/skyward-federal-aviation-administration-test-drones

Mostafa, M. Z., Khater, H. A. K., Rizk, M. R., & Bahasan, A. M. (2019). A Novel GPS/RAVO/MEMS-INS Smartphone-Sensor-Integrated Method to Enhance USV Navigation Systems during GPS Outages. *Measurement Science and Technology, 30*(9), 1-16. https://doi.org/10.1088/1361-6501/ab161c

Muntean, P. & Wallace, G. (2022, January 18). *Airlines ask Biden administration for 'immediate intervention' on 5G*. CNN. https://www.cnn.com/2022/01/17/tech/faa-5g-flight-restrictions/index.html

Nikolić, G. Stojčev, M., Stamenković, Z., Panić, G. & Petrović, B. (2014). Wireless sensor node with low-power sensing. *Electronics and energetics, 27*(3), 435-353. https://doi.org/10.2298/FUEE1403435N

Noh, K-L & Serpedin, E. (2007). *Pairwise broadcast clock synchronization for wireless sensor networks*. 2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 1-6. https://doi.org/10.1109/WOWMOM.2007.4351793

Nur, F. N., Sharmin, S., Razzaque, A., Islam, S., & Hassan, M. M. (2017). A low duty cycle MAC protocol for directional Wireless Sensor Networks. Wireless Personal Communications, 96, 5035-5059. https://doi.org/10.1007/s11277-016-3728-4

Otto, A., Agatz, N., Campbell, J., Golden, B., & Pesch, E. (2018). Optimization approaches for unmanned aerial vehicles (UAV) or aerial drones: A survey. *Networks, 72*(4), 411-458. https://doi.org/10.1002/net.21818

Oxley, L. T. & Roberts, C. J. (1982). Pitfalls in the application of the Cochrane-Orcutt technique. *Oxford bulletin of economics and statistics, 44*(3), 227-240

Pajic, M., & Mangharam, R. (2010). Spatio-temporal techniques for anti-jamming in embedded wireless networks. *EURASIP journal on wireless communications and networking, 2010*, 1-16. https://doi.org/10.1155/2010/819318

Panfilo, G. & Arias, F. (2019). The Coordinated Universal Time (UTC). *Metrologia, 56*(4), 1-26. https://doi.org/10.1088/1681-7575/ab1e68

Peniel, W. & Granshaw, S. I. (2018). RPV, UAV, UAS, RPAS… or just drone? *The photogrammetric record, 33*(162), 162-170. https://doi.org/10.1111/phor.12244

Penttinen, J. T. J. (2016). *Wireless communications security. Solutions for the internet of things*. John Wiley & Sons, Ltd.

Pindyck, R. S., & Rubinfeld, D. L. (1998). *Econometric models and economic forecasts*. Irwin/McGraw-Hill.

Pinto, R. L. & Almeida, L. (2018). A robust approach to TDMA synchronization in aerial networks. *Sensors, 18*(4497), 1-18. https://doi.org/10.3390/s18124497

Prais, S. J. & Winsten, C. B. (1954). *Trend estimators and serial correlation*. Unpublished Cowles Commission, Discussion Paper, Chicago

Qualcomm Technologies, Inc. (2017). LTE Unmanned Aircraft Systems. https://www.qualcomm.com/documents/lte-unmanned-aircraft-systems-trial-report

Radio technical Commission for Aeronautics. (2020). *Assessment of C-Band mobile communications interference impact on low range radar altimeter operations*. https://www.rtca.org/wp-content/uploads/2020/10/SC-239-5G-Interference-Assessment-Report_274-20-PMC-2073_accepted_changes.pdf

Rao, C. R. (1945). Information and the accuracy attainable in the estimation of statistical parameters. *Breakthroughs in Statistics. Springer Series in Statistics (Perspectives in Statistics)*. (pp. 235-247). Springer. https://doi.org/10.1007/978-1-4612-0919-5_16

Rhee, I-K., Lee, J., Kim, J., Serpedin, E. & Wu, Y-C. (2009). Clock synchronization in wireless sensor networks: an overview. *Sensors, 9*, 56-85. https://doi.org/10.3390/s90100056

Rose, J. A. R., Watson, R. J., Allain, D. J., & Mitchell, C. N. (2014). Ionospheric corrections for GPS time transfer. *Radio science, 49*(3), 196-206. https://doi.org /10.1002/2013RS005212

Scott, M. (September 28, 2021). *The era of the cellular-connected drone has arrived*. Verizon newscenter. Retrieved November 4, 2021 from https://www.verizon.com/about/news /era-cellular-connected-drone-has-arrived

Senior, K. L., Ray, J. R., & Beard, R. L. (2008). Characterization of periodic variations in the GPS satellite clocks. *GPS solutions, 12*, 211-225. https://doi.org/10.1007/s10291-008-0089-9.

Serpedin, E. & Chaudhari, Q. M. (2009). *Synchronization in wireless sensor networks*. Cambridge University Press.

Sharma, A., & Mathur, S. (2018). Comparative analysis of ML-PSO DOA estimation with conventional techniques in varied multipath channel environment. *Wireless Personal Communications, 100*, 803-817. https://doi.org/10.1007/s11277-018-5350-0

Small Unmanned Aircraft Systems, 14 C.F.R. § 107 (2016). https://www.ecfr.gov/ current/title-14/chapter-I/subchapter-F/part-107

Snedecor, G.W. & Cochran, W. G. (1967). *Statistical methods*. The Iowa State University Press.

Solomon, P. D., Wang, J., Rizos, C. (2011). Latency determination and compensation in real-time GNSS/INS integrated navigation systems. *International archives of the photogrammetry, remote sensing and spatial information sciences, 38*(1), 303-307.

Son, H., & Lee, S. (2007). Multi-cell communications for OFDM-based asynchronous networks over multi-cell environments. *Wireless networks, 15*, 917-930. https://doi.org/10.1007/s11276-007-0084-z

Spires, J. (2021, April 9). *AT&T shows off its 5G innovation studio, featuring drones*. DroneDJ. Retrieved November 4, 2021 from https://dronedj.com/2021/04/09/att-shows-off-its-5g-innovation-studio-featuring-drones/

Sun, W. & Amin, M. G. (2005). A Self-Coherence Anti-Jamming GPS Receiver. *IEEE Transactions on Signal Processing, 53*(10), 3910-3915. https://doi.org/https://doi.org/10.1109/TSP.2005.855428

Sundaraman, B., Buy, U., & Kshemkalyani, A. D. (2005). Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks, 3*, 281-323. https://doi.org/10.1016/j.adhoc.2005.01.002

Swain, A. R., & Hansdah, R. C. (2015). A model for the classification and survey of clock synchronization protocols in WSNs. *Ad hoc networks, 27*, 219-241. http://dx.doi.org/10.1016/j.adhoc.2014.11.021

Tan, H. H., Liang, R., & Suen, P.-H. (1996). Optimum combining of residual carrier array signals in correlated noises (TDA PR 42-124). National Aeronautics and Space Adminsitration. https://ntrs.nasa.gov/citations/19960022221

Thejll, P., & Schmith, T. (2005). Limitations on regression analysis due to serially correlated residuals: Application to climate reconstruction from proxies. *Journal of geophysical research, 110*, 1-6. https://doi.org/10.1029/2005JD005895

Third Generation Partnership Project. (2021a). *Technical Specification group services and system aspects; Unmanned Aerial Systems (UAS) support in 3GPP; stage 1* (Standard TS 22.125 v17.3.0 [2021-03]). Retrieved from https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3545

Third Generation Partnership Project. (2021b). *Technical specification group services and system aspects; Service requirements for the 5G system; stage 1 (release 18)*. (Standard TS 22.261 v18.3.0 [2021-06]). Retrieved from https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107

Third Generation Partnership Project. (2021c). *Technical specification group services and system aspects; service requirements for cyber-physical control applications in vertical domains; Stage 1 (release 18)*. (Standard TS 22.104 v18.2.0 [2021-09]). Retrieved from https://portal3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3528

Wang, H., Zeng, H., Li, M., Wang, B. & Wang. P. (2017). Maximum likelihood estimation of clock skew in wireless sensor networks with periodical clock correction under Exponential delays. IEEE *Transactions on signal processing, 65*(10), 2714-2724. https://doi.org/10.1109/TSP.2017.2675863

Warner, R. M. (2013). Applied statistics. From bivariate through multivariate techniques. SAGE Publications, Inc.

Wu, Q., Xu, J., Zeng, Y., Ng, D. W. K., Al-Dhahir, N., Schober, R., & Swindlehurst, A. (2021). A comprehensive overview on 5G-and-beyond networks with UAVs: from communications to sensing and intelligence. *IEEE Journal on Selected Areas in Communications, 39*(10), 2912-2945. https://doi.org/10.1109/JSAC.2021.3088681

Yacoub, M. D. (2007). The $\alpha - \mu$ distribution: A physical fading model for the Stacy distribution. *IEEE transactions on vehicular technology, 56*(1), 27-34. https://doi.org/10.1109/TVT.2006.883753

Yi, J., Miraz, D., Kastner, R., Schurgers, C., Roberts, P., & Jaffe, J. (2015). ToA-TS: Time of arrival based joint time synchronization and tracking for mobile underwater systems. *Ad hoc networks, 34*, 211-223. https://dx.doi.org/10.1016/j.adhoc.2014.10.010

Yiğitler, H., Badihi, B., & Jäntti, R. (2020). Overview of time synchronization for IoT deployments: clock discipline algorithms and protocols. *Sensors, 20*, 1-59. https://doi/org/10.3390/s20205928

Zhang, R., Wang, B., & Yi, D. (2020). Skew for saving energy in wireless sensor networks. *IEEE open access journals and conferences, 8*, 107814-107822. https://doi .org/10.1109/ACCESS.2020.3001302

Zhang, J., Zhang, K., Grenfell, R., & Deakin, R. (2006). Short note: on the relativistic Doppler effect for precise velocity determination using GPS. *Journal of geodesy, 80*(2), 104-110. https:/doi.org/10.1007/s00190-006-0038-8

Zimmermann, H. (1980). OSI reference model- The ISO model of architecture for Open System Interconnection. *IEEE transactions on communications, 28*(4), 425-432.