Delft University of Technology

Master Thesis

Time Synchronization for Anchorless Satellite Networks

prepared by

Felix Abel

to obtain the degree of Master of Science

Project duration:	January 2021 – August 2021
Student number:	5014336
Supervisor:	Dr.ir. Prem Sundaramoorthy
Supervisor:	Dr. Raj Thilak Rajan
Committee Chair:	Prof.dr.ir. Alle-Jan van der Veen

Abstract

In this thesis, we propose a new class of pairwise frequency, multi-domain time synchronization and ranging algorithms for anchorless mobile networks of asynchronous nodes. We apply these techniques and study network and mission level aspects of time synchronization to Orbiting Low Frequency Array for Radio astronomy (OLFAR), a proposed distributed radio interferometer.

In the first step, the Frequency Pairwise Least Squares (FPLS) that estimates clock skew and relative velocity in a pairwise setup using only frequency measurements is formulated. In the second step, we extend this method to a motion model with constant acceleration. Since frequency domain methods do not estimate clock offset and pairwise range, relying purely on frequency domain estimates is not feasible for various applications.

To harness the potential of frequency domain synchronization and ranging, the Combined Pairwise Least Squares (CPLS) has been proposed. The combined method reduces the number of minimum required messages from 4 to 3 compared to current methods and decreases the computational complexity. Using a generic simulation with nodes in pairwise non-linear motion, we show that frequency domain methods can outperform time domain methods in clock skew and relative velocity estimation and that the proposed multi-domain method delivers better clock offset and pairwise range estimation in low to medium SNR conditions.

In the second part of our work, we apply the proposed methods to OLFAR —— a space-borne large aperture radio interferometric array platform. We address network level and mission level aspects, proposing network path planning for pairwise synchronization algorithms and determining the required resynchronization period.

Preface

We are just an advanced breed of monkeys
on a minor planet of a very average star.
But we can understand the Universe.
That makes us something very special.

— Stephen Hawking

This thesis project is the final building block of my Masters education at TU Delft. I would like to express my gratitude to my teachers that have helped me adjust to the demanding environment of an international research university and to navigate the challenges of online education during a pandemic. You have inspired me and equipped me with the skills and tools to tackle this thesis.

Completing this thesis would not have been possible without many people who accompanied me along the way. I would like to thank my daily supervisor, Prem, for taking me on as a student and guiding me through this thesis project. Your enthusiasm about space exploration, your prior work on OLFAR and your expertise in orbital mechanics and patience when explaining them to me were invaluable. I would also like to thank my supervisor Raj. Building on top of your own work and relying on your expertise in time synchronization has been a great asset. Without your help to quickly reduce complex problems and helping me limit the scope of this project, I would not have been able to complete it on time and within scope. Working with both of you has truly been a pleasure and honor.

Furthermore, I would like to express my gratitude to the Circuits and Systems group and all its members, particularly the group of Master students in which we met weekly for topical exchange and to Professor van der Veen for serving as Chairman of my thesis committee.

I want to thank my Dutch and International friends and my housemates for the exchange, the fun times and the interesting conversations. Thank you for making the past two years an amazing experience despite these adverse circumstances for student life.

Finally, I would like to express my deep gratitude to my family. Without your unconditional love and support none of this would have been possible. Danke für alles!

Felix Abel
Delft, August 2021

List of Figures

List of Tables

Contents

1 | Introduction

1.1 Motivation

The aim of this thesis is to improve time synchronization methods for anchorless mobile networks of asynchronous nodes and to contribute a system level characterization by applying the developed methods to a space interferometry mission — OLFAR.

Asynchronous networks are characterized by the lack of an absolute time reference. However, time synchronization among the nodes is often required in wireless sensor networks to fulfil their tasks for example for interferometry or localization. Such networks commonly have limited available energy resources. It is thus desirable to decrease communication and processing needed to achieve synchronization among nodes using pairwise synchronization algorithms. By improving and augmenting prevalent algorithms [1], [2], this thesis aims to contribute to making future wireless sensor networks more energy efficient and more widely applicable.

While applications are numerous, the particular motivation for this thesis is the application of anchorless satellite networks, and, more specifically the OLFAR mission — a space-based radio interferometer introduced in section 1.4. The aforementioned synchronization techniques will be applied to and tailored for this network of satellites. The goal is to optimize the time synchronization parameters for overall system performance taking into account not only the time synchronization algorithm itself but also system and mission aspects.

1.2 Time synchronization

Anchorless mobile networks of asynchronous nodes do not have an external absolute time reference and the local time references produced by an erroneous clock at each node are initially dissimilar. Furthermore, the nodes are in relative motion to one another, meaning their distances and hence message propagation delays change within the period required to synchronize the nodes.

A clock model is employed to relate the erroneous local time provided by a local clock to the true time. The purpose of time synchronization is to find the clock model parameters of all nodes such that the calibrated time at all nodes in the network is the same — the nodes are synchronized. Synchronization is achieved by exchanging time-stamped messages between the

nodes.

Among network time synchronization methods one can distinguish between pairwise methods and global methods. Pairwise methods synchronize a network by synchronizing pairs of nodes with one another whereas global methods compute a solution for the whole network in a centralized algorithm [3, ch. 4]. This work will limit its scope to pairwise methods because of their linear complexity with number of nodes in the network, their lower computational requirements and their inherently distributed nature.

Current algorithms for time synchronization in anchorless networks of mobile asynchronous nodes rely solely on measuring time stamps of exchanged messages. The main contribution this thesis aims to make in the field of time synchronization algorithms is the use of frequency domain methods to augment existing time domain methods. Frequency measurements could cater to skew estimation and doppler velocity estimation and improve overall parameter estimation performance.

1.3 Distributed space systems

In the past decades, the miniaturization in electronics led to ever smaller spacecraft. The resulting decrease in absolute launch cost facilitated a rising number of distributed space systems with increasing number of nodes in such systems. Distributed space systems generally refer to multiple spacecraft being operated to achieve the goal of the space mission. There are different forms of distributed space systems, such as constellations, formation flying, swarms and fractionated spacecraft. The remainder of this section will follow [4] in presenting definitions for these distributed space systems.

*A **constellation** is a set of satellites distributed over space intended to work together to achieve common objectives* [5]. Typical examples of such constellations are Global Navigation Satellite Systems (GNSS) like GPS, GLONASS, Galileo and BeiDou that provide global coverage with navigation signals, or communications constellations like Iridium, Starlink and Oneweb.

The NASA Goddard Space Flight Center gives an engineering definition of **formation flying** as *the tracking or maintenance of a desired relative separation, orientation or position between or among spacecraft* [6]. Compared to other configurations of distributed space systems, the requirements regarding the relative positioning are unique. Such formations can be used to observe fast changing phenomena on Earth or in Earth's atmosphere e.g. fires or hurricanes or they can measure the changes e.g in Earth's gravitational field through its action on the satellite formation.

*A **fractionated spacecraft** is a new architectural model whereby a satellite is decomposed into a set of similar or dissimilar component modules which interact wirelessly while in cluster orbits* [7].

*A **Swarm** is defined as a set of agents which are liable to communicate directly or indirectly with each other and which collectively carry out a distributed problem solving* [8]. Other definitions like [9] apply more specifically to satellites: *A satellite swarm consists of a large number of physically identical elemental satellites.* In contrast to constellations, satellite swarms often operate in close proximity or let the swarm elements drift freely in a limited boundary.

Spacecraft deployed beyond Earth orbit, for example in deep space or around the Moon, typically lack GNSS-like clock and position references and are hence anchorless networks. The increased interest in distributed space systems — particularly fractionated spacecraft and swarms — beyond Earth orbit makes anchorless satellite networks a prime candidate for applying the time synchronization methods introduced in the previous section. Examples for such missions include missions to Mars or Moon [10], or radio astronomy missions like OLFAR.

1.4 OLFAR

OLFAR stands for Orbiting Low Frequency Array for Radio astronomy [11]. Its concept has been developed mainly in the last decade by a group of universities and research institutions in the Netherlands. They aim to augment the capabilities of the ground-based Low-Frequency Array (LOFAR) operated by the Netherlands Institute for Radio astronomy (ASTRON) [12]. The use case for OLFAR is to provide a space-borne large aperture radio interferometric array platform in a frequency range from 0.3 MHz to 30 MHz [9], [13]. By stationing OLFAR in space far away from Earth, in lunar orbit or Earth-Moon L2, interference from Earth can be mitigated [9]. In this thesis, lunar orbit as stationing location will be examined. OLFAR is proposed to consist of a swarm of $\geqslant 10$ satellites [13], however, the exact number of elements varies across the different publications. The performance of OLFAR as distributed radio telescope is strongly dependent on achieving time synchronization among its nodes. While prior work has been carried out on potentially suitable synchronization algorithms, a system level time synchronization study for OLFAR has not been conducted — a gap that this thesis aims to close.

1.5 Scope and problem statement

The problem statement of this thesis is *pairwise time synchronization of anchorless networks of mobile asynchronous nodes in a fully connected configuration applied to satellite swarms, namely the OLFAR interferometry mission.*
To clearly outline the scope of this work, the aspects of time synchronization can be attributed to three different hierarchy levels — pairwise, network and mission — shown in figure 1.1.

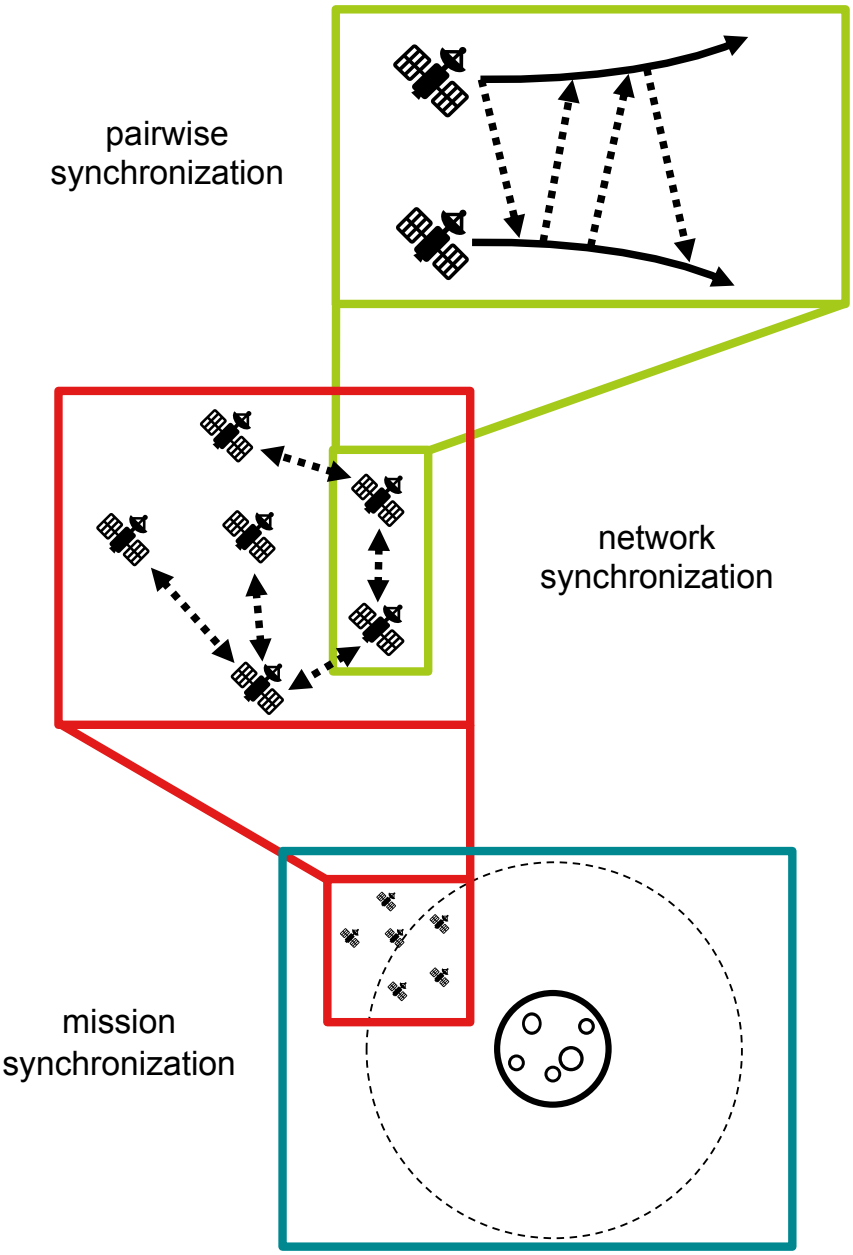


Figure 1.1: Synchronization on pairwise, network and mission level (microscopic to macroscopic)

On the microscopic level, pairwise time synchronization algorithms synchronize the clock of one node to another reference node through pairwise message exchange. On the next level — network synchronization — the pairwise synchronization techniques are employed to synchro-

nize a network of N asynchronous nodes. Finally, on mission level, network synchronization must be performed repeatedly and with an accuracy good enough to fulfill the mission requirements, in this case, those of the OLFAR space interferometry mission. This thesis aims to contribute to all three levels, firstly by conducting a literature study into existing pairwise synchronization algorithms in chapter 2. Then in chapters 3 and 4 a class of novel frequency domain and multi domain pairwise synchronization algorithms are proposed and validated. In the second part of this thesis, namely chapters 5 and 6, various aspects of network and mission level synchronization are addressed on the example of the OLFAR space mission, resulting in proposing a time synchronization protocol for the OLFAR mission.

The general problem statement can be broken down into three research questions:

R1 **Frequency information:** Can frequency information be used to improve pairwise time synchronization performance? If yes, how? (chapters 3 and 4)

R2 **Network synchronization:** How can pairwise algorithms be used on a fully connected network? Which synchronization path should be chosen? (chapters 5 and 6)

R3 **Mission synchronization:** In the OLFAR mission, how can the network synchronization be employed to achieve the mission level synchronization requirements? How frequently does resynchronization need to be performed? (chapters 5 and 6)

2 | Time synchronization methods

2.1 Introduction

In this chapter, the topic of time synchronization will be introduced followed by a section on clocks and then a literature study into existing time synchronization methods. Time synchronization of inherently asynchronous nodes is a key challenge in wireless sensor networks. Where available, external time references such as a Global Navigation Satellite System (GNSS) can be used to achieve time synchronization, however, there are applications with limited or no access to an absolute time and position reference, e.g. in space beyond Earth orbit, dense urban environments, underwater or indoor applications. In such anchorless networks, where the lack of an external reference prevents absolute synchronization, synchronization among the nodes can be achieved using different algorithms. This chapter will thus address the research question 1: What state-of-the art pairwise time synchronization algorithms can be used for clock parameter estimation of anchorless networks of mobile asynchronous nodes?

For time synchronization, one can categorize the methods according to the clock model used, the position model used and furthermore distinguish between pairwise and global methods. In section 2.2, the clock model used for the scope of this work will be presented together with existing performance requirements for clock errors. The *Allan deviation*, a prevalent measure for clock error will be introduced. The position model represents the nature of the relative positions of the nodes in the network. After introducing the general concept of two way ranging in section 2.3, section 2.4 will explain the origin of the different delay components and how they relate to the position model. As the propagation delay — typically a dominant portion of the total delay — of a message sent between two nodes depends on the position model, its choice is paramount to successful time synchronization. In the last section of this chapter, three time domain methods with different position models will be described. First, a stationary scenario where the pairwise propagation delays are known as described by [14] will be examined in subsection 2.5.1. Second, a stationary scenario where delays are unknown will be considered. For this case, a method for joint synchronization and ranging that estimates clock parameters and pairwise distances by [1], [15] will be presented in subsection 2.5.2. Third, an extension to mobile anchorless networks will be addressed in subsection 2.5.3 . The class of mobile methods

presented in [2] and [3, ch. 4] model the propagation delay as L-th order Taylor approximation, thus the motion model can be adapted to the use case. As the scope of this thesis are anchorless networks of mobile asynchronous nodes, this class of algorithms seems most promising to be applicable to the scenario of interest.

Note, that besides the prevalent methods specifically for anchorless mobile networks, a plethora of time synchronization algorithms exist for WSNs in general as synchronization and ranging is a field of active research. For example in [16], Kazaz et al. use phase difference of arrival (PDoA) measurements of narrowband signals for synchronization in stationary IoT networks incorporating offset and skew effects. Wang et al. propose a clock skew estimator based on passive listening in [17]. In [18], Zhu et al. propose an innovative synchronization method based on reaching distributed consensus between the nodes of the network. However, none of these three methods operate on mobile networks. For mobile networks, recent advances have been made by Gu et al. who built on top of the MGLS algorithm from [2]. They proposed a Synchronous Two-Way Ranging based on pseudo range measurements thereby reducing computational complexity [19].

When it comes to network synchronization one can distinguish between pairwise methods and global methods. Pairwise methods synchronize two nodes at a time, whereas global ones collect information on the whole network to then synchronize it. This thesis will limit its scope to pairwise methods, however, section 2.6 will briefly address a global method by [2] and [3, ch. 4] in differentiation to the pairwise methods.

2.2 Clocks

In this section, the topic of clock modeling is addressed. The goal of this section is to find a model to relate true time to the perceived time of a local oscillator. The reasons for clock errors — the difference between true and perceived local time — are the result of clock imperfections and initial lack of synchronization. As it would reach beyond the scope of this thesis, it is not the goal of this section to discuss clock hardware or go deep into derivation of clock models.

Oscillators produce a phase error leading to a timing error. Let the time $\bar{\mathcal{C}}(t)$ at the local oscillator expressed as a polynomial function of t

$$\begin{aligned} \bar{\mathcal{C}}(t) &= t + \delta\bar{\mathcal{C}}(t), \\ \bar{\mathcal{C}}(t) &= \phi + \omega t + 0.5\ddot{\phi}t^2 + ... + \eta(t) \end{aligned} \tag{2.1}$$

where ϕ is the phase error or clock offset, ω is the frequency offset or skew and $\ddot{\phi}$ to $\eta(t)$ represent higher order polynomial terms and noise [3, ch. 4]. For an ideal clock, $\left[\phi, \omega, \ddot{\phi}, ..., \eta\right] = \left[0, 1, 0, \mathbf{0}^T, 0\right]$ and thus $\bar{\mathcal{C}}(t) = t$. To alleviate the errors, the clock parameters $\phi, \omega, \ddot{\phi}, ...$ could be estimated. However, one needs to choose the model order such that it is high enough for the local time to be sufficiently stable for the application at hand but also low enough to avoid

overfitting problems. Only estimating offsets would lead to high errors for the application and higher order parameters typically do not remain stable. Therefore, for the scope of this work, a commonly chosen model — the affine clock model — where offset and skew are estimated and higher order parameters are viewed as noise is chosen. The remaining non linear components are combined into the *Allan deviation* $\sigma_\zeta(\tau_c)$, which is defined as the expected value of the root mean square of the aforementioned non linear components. τ_c is the coherence time between two measurements. The Allan deviation is the figure of merit of the stability of an oscillator and can be obtained from manufacturers of clock hardware.

In prior work on radio astronomy, a requirement for clock accuracy has been defined as the RMS phase error of the clock remaining less than 1 radian [20], [3, ch. 3], hence

$$2\pi\nu_0\sigma_\zeta(\tau_c)\tau_c \lessapprox 1 \tag{2.2}$$

where ν_0 is the observation frequency.

2.3 Two Way Ranging

This section introduces the affine clock model and the concept of two way ranging (TWR) using time measurements as described and applied in [1]–[3], [14], [15]. To give an overview over the variables used in this chapter table 2.1 has been compiled.

Table 2.1: Variables time measurement following [3, ch. 4]

variable	unit	description
t	s	true time
i, j		node indices
k		index of pairwise communication
t_i	s	local time at node i
ω_i		clock slew of clock i
ϕ_i	s	clock offset of clock i
α_i		skew calibration parameter of clock i
β_i	s	offset calibration parameter of clock i
β_i	s	offset calibration parameter of clock i
$T_{ij,k}$	s	timestamp for the k-th transmission recorded at node i
$T_{ji,k}$	s	timestamp for the k-th transmission recorded at node j

The affine clock model represents the local time at node i

$$t_i = \omega_i t + \phi_i \qquad \Leftrightarrow \qquad \mathcal{C}_i\left(t_i\right) \triangleq t = \alpha_i t_i + \beta_i \tag{2.3}$$

where the calibration parameters are uniquely related to the clock parameters as $[\alpha_i, \beta_i] \triangleq \left[\omega_i^{-1}, -\phi_i\omega_i^{-1}\right]$ and $\mathcal{C}_i\left(t_i\right)$ expresses true time as a function of local time [3, ch. 4]. For the scope of this thesis, clock parameters are assumed to be constant over the maximum coherence

time $\tau_{c,max}$ and the synchronization period $t_r \ll \tau_{c,max}$ that the network takes to estimate clock parameters to be small compared to the maximum coherence time.

To synchronize their clocks, let two nodes i and j exchange time-stamped messages. As transmission is not instantaneous, there is a delay between a message being time-stamped by the transmitter i and the receiver j. This delay commonly consists of deterministic and stochastic terms and modeling them correctly is critical for good performance of the time synchronization algorithm. This aspect will be addressed in section 2.4. Following the notation of [3, ch. 4] and expressing the delay in terms of local times measured during message exchange we get for the k-th transmission,

$$\tau_k = \mathcal{C}_j\left(T_{ji,k}\right) - \mathcal{C}_i\left(T_{ij,k}\right) \qquad \text{for } i \rightarrow j \tag{2.4a}$$

$$\tau_k = \mathcal{C}_i\left(T_{ij,k}\right) - \mathcal{C}_j\left(T_{ji,k}\right) \qquad \text{for } i \leftarrow j \tag{2.4b}$$

where $T_{ij,k}$ and $T_{ji,k}$ are the timestamps collected at node i and j respectively and τ_k is the delay at the k-th transmission. The arrow denotes the direction of transmission. More generally, for bi-directional transmission one can write,

$$E_{ij,k}\tau_k = \mathcal{C}_j\left(T_{ji,k}\right) - \mathcal{C}_i\left(T_{ij,k}\right) \qquad \text{for } i \leftrightarrow j \tag{2.5}$$

where $E_{ji,k}$ indicates the direction of transmission as

$$E_{ij,k} = \begin{cases} +1 & i \rightarrow j \\ -1 & i \leftarrow j \end{cases}. \tag{2.6}$$

Expressing equation (2.6) in terms of calibration parameters,

$$E_{ij,k}\tau_k = \alpha_j T_{ji,k} + \beta_j - \alpha_i T_{ij,k} - \beta_i, \tag{2.7}$$

$E_{ij,k}$, $T_{ij,k}$ and $T_{ji,k}$ are known from measurements, α and β are to be estimated, the challenge is to model τ_k. A vectorial notation for K transmissions can then be adopted

$$\begin{bmatrix} -\mathbf{t}_{ij} & \mathbf{t}_{ji} & -\mathbf{1}_K & \mathbf{1}_K \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \\ \beta_i \\ \beta_j \end{bmatrix} = \mathbf{e}_{ij} \odot \boldsymbol{\tau}_{ij}, \tag{2.8}$$

where

$$\mathbf{t}_{ij} = \left[T_{ij,1}, T_{ij,2}, ..., T_{ij,K}\right]^T \in \mathbb{R}^{K\times 1} \tag{2.9a}$$

$$\mathbf{e}_{ij} = \left[E_{ij,1}, E_{ij,2}, ..., E_{ij,K}\right]^T \in \mathbb{R}^{K\times 1} \tag{2.9b}$$

$$\boldsymbol{\tau}_{ij} = \left[\tau_{ij,1}, \tau_{ij,2}, ..., \tau_{ij,K}\right]^T \in \mathbb{R}^{K\times 1}. \tag{2.9c}$$

The concept of all pairwise methods is to assume one node as the reference and synchronize the other node's clock parameters to the reference node. Let i be the reference node with $[\alpha_i, \beta_i] \triangleq [1, 0]$, which leads to the following equation:

$$[\mathbf{t}_{ji} \quad \mathbf{1}_K] \begin{bmatrix} \alpha_j \\ \beta_j \end{bmatrix} = \alpha_i \mathbf{t}_{ij} + \mathbf{e}_{ij} \odot \boldsymbol{\tau}_{ij} \tag{2.10}$$

The problem of determining $\boldsymbol{\tau}_{ij}$ will be addressed in the next section.

2.4 Delay

The true delay of the k-th transmission, broken into its components as follows [15].

$$\tau = \tau_{send} + \tau_{access} + \tau_{transmission} + \tau_{propagation} + \tau_{reception} + \tau_{receive} \tag{2.11}$$

Table 2.2 gives an overview of the different delay components that contribute to the total delay τ between the true time of the timestamp at transmitter and receiver as described in (2.11).

Table 2.2: Delay components as defined by [15], [21]

delay component	layer	description
τ_{send}	application	Send Time: The time spent in building the message and processing the send request.
τ_{access}	MAC	The waiting time for accessing the channel which is highly variable and depending on the specific MAC protocol.
$\tau_{transmission}$	PHY	Transmission Time: The time for transmitting a message.
$\tau_{propagation}$	PHY	Propagation Time: The actual time for a message to be transmitted from the sender to the receiver in a wireless channel.
$\tau_{reception}$	PHY	Reception Time: The time required for receiving, which is assumed to be the same as the transmission time.
$\tau_{receive}$	application	Receive Time: Time to construct and send the received message to the application layer at the receiver.

The components are introduced on different layers, namely on the application, medium access control (MAC) and physical (PHY) layers. When doing time stamping on application layer, all these components have to be taken into account. Especially the highly variable channel access time can make delay estimation difficult. For the scope of this thesis, PHY layer time stamping is considered, which simplifies the equation for the delay to

$$\tau = \tau_{transmission} + \tau_{propagation} + \tau_{reception} \tag{2.12}$$

This total delay can be represented by a deterministic term, which can be constant (stationary nodes, LCLS) or a Taylor series expansion (mobile nodes, MPLS and MGLS) and an additive stochastic term. The reason for the stochastic term is noise on timestamps due to quantization errors in ADCs, clock jitter, noise on distance and the physical medium [3, ch. 4]. For the scope of this thesis, it is assumed to be Gaussian [22], [3, ch. 4]. Noise on time and frequency measurements will be addressed further in section 4.2.

2.5 Pairwise time domain methods

2.5.1 Known delay

The optimal joint estimation of clock offset and skew under known delays τ_k was demonstrated by Noh et al. in [14]. Noh et al. assumed a stationary scenario. They further asserted that the delays τ_k were composed of a known delay component τ (constant over the synchronization period) and a stochastic component η_k such that

$$\tau_k = \tau + \eta_k \quad \forall\, k, \tag{2.13}$$

allowing to reformulate equation (2.10) as

$$\begin{bmatrix}\mathbf{t}_{ji} & \mathbf{1}_K\end{bmatrix}\begin{bmatrix}\alpha_j \\ \beta_j\end{bmatrix} = \mathbf{t}_{ij} + \tau\mathbf{e}_{ij} + \mathbf{e}_{ij} \odot \boldsymbol{\eta}_{ij}, \tag{2.14}$$

where

$$\boldsymbol{\eta}_{ij} = \left[\eta_{ij,1}, \eta_{ij,2}, ..., \eta_{ij,K}\right]^T \in \mathbb{R}^{K\times 1}. \tag{2.15}$$

For the stochastic delay component, zero mean Gaussian and exponential distributions were considered in [14]. For the Gaussian case, the maximum likelihood estimator (MLE) was derived by Noh et al. for $K = 2$. More generally for $K \geqslant 2$ one can write

$$\begin{bmatrix}\hat{\alpha}_j \\ \hat{\beta}_j\end{bmatrix} = \begin{bmatrix}\mathbf{t}_{ji} & \mathbf{1}_K\end{bmatrix}^{\dagger}\left[\mathbf{t}_{ij} + \tau\mathbf{e}_{ij}\right]. \tag{2.16}$$

While being the basis for further pairwise synchronization methods, the constraint that the pairwise distances must be known beforehand makes this method unusable for the type of anchorless networks targeted in this thesis.

2.5.2 Low Complexity Least Squares

In a next step, Wu et al. considered a scenario with constant but unknown propagation delay between the nodes for the duration of all K communications [1], [15]. As the unknown delay is required for clock parameter estimation, it is inevitable to estimate the propagation delay.

Hence, this algorithm is not merely a time synchronization method but rather a joint synchronization and ranging method. This is true for the further methods presented in this section. To jointly estimate the constant delay and clock parameters equation (2.14) can be rewritten as

$$\begin{bmatrix} \mathbf{t}_{ji} & \mathbf{1}_K & -\mathbf{e}_{ij} \end{bmatrix} \begin{bmatrix} \alpha_j \\ \beta_j \\ \tau \end{bmatrix} = \mathbf{t}_{ij} + \mathbf{e}_{ij} \odot \boldsymbol{\eta}_{ij}. \tag{2.17}$$

Again under the assumption of i.i.d. zero mean Gaussian distribution for the stochastic delay component $\boldsymbol{\eta}_{ij}$, the clock parameters and delays can be estimated as

$$\begin{bmatrix} \hat{\alpha}_j \\ \hat{\beta}_j \\ \hat{\tau} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{ji} & \mathbf{1}_K & -\mathbf{e}_{ij} \end{bmatrix}^{\dagger} \mathbf{t}_{ij}, \tag{2.18}$$

which leads to the Low Complexity Least Squares (LCLS) algorithm formulated by [1], [15]. This algorithm can be used for pairwise clock synchronization and ranging. Like all pairwise methods it can be extended to a network of nodes if the network forms a connected graph meaning each node has at least one link to the network. While the scope of this thesis are mobile nodes where the distance between the nodes varies over the course of the synchronization period, the solution to the static model serves as a baseline for solutions of the mobile scenarios under examination in the next subsection.

2.5.3 Mobile Pairwise Least Squares

In an effort to improve synchronization and ranging performance for networks of mobile nodes, Rajan and van der Veen have presented a class of more advanced synchronization techniques by extending the LCLS to a motion model that assumes constant velocity [23] or even constant acceleration [24]. A universal formulation employing an L-th order motion model was presented by the same authors in [3, ch. 4], [2], namely Mobile Pairwise Least Squares (MPLS) and Mobile Global Least Squares (MGLS). In this section an overview is provided over prior work on the MPLS method. Higher-order motion models significantly improved the estimation performance for mobile anchorless networks and are applicable even for non-linear pairwise motion. For this class of algorithms, the varying delay is modeled as

$$\tau_k = \gamma_{ij}^{(0)} + \gamma_{ij}^{(1)} T_{ij,k} + \gamma_{ij}^{(2)} T_{ij,k}^2 + ... + \gamma_{ij}^{(L-1)} T_{ij,k}^{L-1} + \eta_{ij,k} \tag{2.19}$$

where L is the order of the algorithm and $\gamma_{ij}^{(\cdot)}$ are the translated range parameters of the Taylor series expansion. The translated range parameters $\gamma_{ij}^{(\cdot)}$ are uniquely related to the range parameters (distance, velocity, acceleration, ...). It should be noted that the authors are implicitly assuming that transmission times are so short that the approximation of choosing one of the two timestamps of transmission or reception as variable for the Taylor approximation

representing the time of message exchange is sufficient. $\eta_{ij,k}$ again is an i.i.d. zero mean Gaussian random variable representing the stochastic delay component.

Equation (2.7) can then be extended to

$$\alpha_i T_{ij,k} + \beta_i - \alpha_j T_{ji,k} - \beta_j + E_{ij,k}\left(\gamma_{ij}^{(0)} + \gamma_{ij}^{(1)} T_{ij,k} + ... + \gamma_{ij}^{(L-1)} T_{ij,k}^{L-1}\right) + \eta_{ij,k} = 0, \tag{2.20}$$

or, in vectorial notation

$$\begin{bmatrix} \mathbf{t}_{ij} & -\mathbf{t}_{ji} & \mathbf{1}_K & -\mathbf{1}_K & \operatorname{diag}(\mathbf{e}_{ij})\,\mathbf{V}_{ij} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \\ \beta_i \\ \beta_j \\ \gamma_{ij}^{(0)} \\ \gamma_{ij}^{(1)} \\ \gamma_{ij}^{(2)} \\ \vdots \\ \gamma_{ij}^{(L-1)} \end{bmatrix} = \boldsymbol{\eta}_{ij}, \tag{2.21}$$

where $\mathbf{V}_{ij}$ is the Vandermonde matrix

$$\mathbf{V}_{ij} = \begin{bmatrix} \mathbf{t}_{ij}^{\odot 0} & \mathbf{t}_{ij}^{\odot 1} & \mathbf{t}_{ij}^{\odot 2} & ... & \mathbf{t}_{ij}^{\odot L-1} \end{bmatrix}. \tag{2.22}$$

Following other pairwise models, the calibration parameters of node i are set to 1 and 0 respectively to then solve in a least squares way, similar to the LCLS case [3, p. 74]:

$$\begin{bmatrix} \hat{\alpha}_j \\ \hat{\beta}_j \\ \hat{\gamma}_{ij}^{(0)} \\ \hat{\gamma}_{ij}^{(1)} \\ \hat{\gamma}_{ij}^{(2)} \\ \vdots \\ \hat{\gamma}_{ij}^{(L-1)} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{ji} & \mathbf{1}_K & -\operatorname{diag}(\mathbf{e}_{ij}) \end{bmatrix}^{\dagger} \mathbf{t}_{ij} \tag{2.23}$$

For the problem to be solvable, $K \geqslant 2 + L$ must hold for the number of communications and communication must be bi-directional, e.g. at least one message in each direction.

The mobile pairwise method from Rajan and van der Veen that was described in this subsection is out of the existing time domain methods the most suitable for the application of anchorless satellite networks, as these are generally in relative motion.

In anticipation of the frequency domain algorithms proposed in the next chapter, the approach of extending 0-th order model to an L-th order model can be implemented in frequency

domain in a similar fashion. Additionally, equation (2.21) can be modified to be used in combined algorithms operating in both time and frequency domain.

2.6 Global vs. pairwise methods

Besides the pairwise method, Rajan and van der Veen proposed global methods for time synchronization. More specifically, a stationary global method was presented in [25] and a mobile global formulation in [2],[3, ch. 4]. Global formulations like the MGLS algorithm solve the time synchronization and ranging problem for the whole network of nodes rather than in a pairwise manner. The formulation leads to a block-incidence-matrix-like structure in a system of equations that is initially under-determined for lack of a clock reference. In [3, ch. 4], several equality constrained least squares formulations have been proposed to solve this optimization problem. The possibility of choosing a virtual average clock as reference via the constraints (sum constraint, see [3, p. 85]) rather than a reference node as in pairwise methods can lead to improvements in clock parameter estimation. Furthermore, global methods can exploit information gathered on all $\frac{N(N-1)}{2}$ pairwise links which further contributes to the superior performance of global algorithms in fully connected networks. In contrast, existing pairwise methods are by design only able to make use of $N-1$ links. Nonetheless, pairwise methods are computationally less demanding as they scale linearly with network size requiring $N-1$ pairwise synchronizations where each pairwise operation is evidently independent of network size. The computational complexity of global methods for large networks is given as $\mathcal{O}(KN^2L^2)$ [3, p. 81]. Global algorithms are centralized, meaning they lend themselves well to Master/Slave configurations where a centralized node collects all data, computes the result to then distribute it to the nodes. MGLS is distributable [3, p. 81], however the distributed version will require additional communication between nodes and further increase implementation complexity. Pairwise methods are inherently distributed requiring no further communication in the network.

The anchorless satellite networks that this work is targeting are typically swarms of identical objects. Besides, it is a common requirement for distributed systems in general and distributed space systems in particular to have no single point of failure (SPOF), which disqualifies centralized algorithms for these applications. The distributed version of MGLS would circumvent any SPOF concerns, however, there are three strong points to be made against using global methods: Firstly, satellite swarms such as OLFAR often follow the New-Space approach of low-cost, simple design swarm elements which stands in stark contrast to the complex implementation of distributed MGLS. Secondly, the additional number of communications required to form a fully connected network (prerequisite for the major part of overperformance of MGLS over MPLS and required for the distributed solving) comes at a time and power consumption penalty. Thirdly and lastly, while OLFAR with its $\geqslant 10$ elements is a rather small satellite

swarm, the number of elements in satellite swarms tends to grow in the last decades as projects like Starlink ($\geqslant$ 10000 elements expected) show. This illustrates a big disadvantage of global methods as their computational complexity grows with the number of elements squared making these methods essentially unusable for large swarms. Frequency domain global methods are expected to behave similarly with respect to distributed implementation and computational complexity.

Consequently, this work will exclusively focus on pairwise synchronization methods in time and frequency domain as these are deemed more suitable for anchorless satellite systems in general and the OLFAR project in particular.

2.7 Summary

In summary, this chapter provided an introduction to time synchronization and the concept of two way ranging (TWR). It briefly addressed clock modeling and introduced the Allan deviation as clock performance measure. This will serve as basis for system level trade offs in later chapters. It determined that an affine clock model with clock offset and clock skew as parameters will be used in this work. The delay during message exchange was broken down into its components. It was decided to assume physical layer time stamping for the scope of this work, and, to assume the stochastic component of the delay to be zero mean Gaussian which was found to be the prevalent assumption in literature. Three pairwise time synchronization methods, one with known delay, LCLS with unknown delay in a stationary scenario and MPLS with unknown delay in a mobile scenario. The two latter methods are applicable to anchorless satellite networks (LCLS is expected to suffer from the insufficient dynamic model), and are thus to be used as baseline performance comparison for novel algorithms in frequency domain derived in the following chapter. It was furthermore determined that — due to their computational complexity and inherent centralized formulation — global methods will not be further considered in this thesis. Instead, the focus will remain on pairwise algorithms.

3 | Frequency domain synchronization

3.1 Introduction

In this chapter, the possible use of frequency measurements to improve time synchronization performance is explored to then propose frequency domain and multi domain synchronization methods. Proposing these novel methods constitutes one of the key contributions of this thesis. The research question to be addressed is **R1**:
Can frequency information be used to improve pairwise time synchronization performance? If yes, how?

It will be shown in this chapter, that a frequency domain method can directly estimate the doppler frequency and thus relative velocity of the nodes and estimate the clock skew independently from the offset. Finally, combining a novel method based on frequency measurements with existing time domain methods into a combined method has the potential to increase the overall estimation performance of the clock parameters.

In the next section, the need to investigate frequency domain methods for time synchronization is briefly motivated. Then, in section 3.3, the doppler effect and some basic relations are presented. In section 3.4, a two way ranging framework in frequency domain is derived, relating measured frequencies at the nodes with clock skew and relative velocity. Subsequently, three frequency domain algorithms are proposed allowing to estimate the aforementioned quantities in a pairwise scenario. First, an estimator based on uni-directional transmission is proposed. Second, the uni-directional formulation is extended to bi-directional communication leading to the Frequency Pairwise Least Squares (FPLS) algorithm. Third, the algorithm is extended to a motion model of constant acceleration proposing the Higher Order Frequency Pairwise Least Squares (HFPLS). Frequency domain methods cannot estimate clock offset and range. Using frequency domain information to augment time domain information, the Combined Pairwise Least Squares (CPLS) is proposed in section 3.6. Finally, an overview over existing time domain and novel frequency and multi domain synchronization techniques is presented before summarizing this chapter.

3.2 Motivation

The concept of augmenting time domain information with frequency domain information is frequently used in wireless sensing applications, for example in doppler radars [26] or doppler tracking of spacecraft by ground stations [27], [28]. Even in timing and ranging applications one can find examples of how frequency information is exploited, for example, in frequency difference of arrival (FDOA) ranging applications [29]. However, such applications require prior clock synchronization when used in a network of nodes. Another example is given in [30], where frequency information is used in the context of clock synchronization of stationary wireless sensor networks, however, frequency information is used in here as a binary indicator for the synchronization status of the clocks in the network rather than for clock parameter estimation. Among the prior work in this area, [31] comes closest to the application of interest in this thesis. In their work, Roehr et al. exploit frequency domain information for clock synchronization in a pairwise scenario. However, their approach is limited to stationary nodes and requires a coarse presynchronization of node clocks and is based on using frequency modulated continuous wave (FMCW) signals.

In the problem posed in this thesis — time synchronization in an anchorless network of mobile asynchronous nodes — the doppler shift on the communication between the nodes due to their relative motion and the clock skews ω_i of the nodes are directly related to the frequencies measured at transmitter and receiver. State-of-the-art time synchronization methods for such a scenario operate only in time domain as presented in the previous chapter. Frequency information is not exploited in current estimators.

Generally, it is desirable to achieve time synchronization with minimal number of message exchanges K to keep communication and computation effort low. Exploiting frequency information can potentially reduce the number of messages K, or, at constant number of messages increase the accuracy of the estimation. Assuming Gaussian noise on measurements, the aim of this chapter is to first derive least squares estimators for the clock skew of an asynchronous node in a pairwise scenario. Different estimators are derived for a scenario with constant pairwise velocity for uni-directional messaging, with constant pairwise velocity for bi-directional messaging and a scenario with a higher order motion model. These estimators by themselves can be used in applications where the offset is not of interest and only clock offset and skew is required. However, for applications in the scope of this thesis both clock offset and skew estimation are required. Therefore, secondly, the proposed frequency domain estimations can be used to augment time domain estimators with the goal to formulate a combined method that outperforms prevalent methods with respect to communication and computation cost and estimation accuracy.

3.3 Doppler effect

In this section, doppler effect basics and related equations will be introduced. Let an electromagnetic wave be sent from transmitter to a receiver that is moving w.r.t the transmitter. The doppler frequency — the frequency shift due to relative motion — can be expressed as

$$f_d = f_r - f_t \tag{3.1}$$

where f_t and f_r are the transmitted frequency and received frequency respectively.For small doppler velocities $|v_d| \ll c$, the doppler velocity — for the scope of this work typically $0 < |v_d| < 10^3 \frac{\mathrm{m}}{\mathrm{s}}$ in magnitude — can be expressed as

$$v_d = \frac{f_d c}{f} \qquad \Leftrightarrow \qquad f_d = \frac{v_d f}{c} \tag{3.2}$$

where c is the propagation speed of the wave — for the scope of this work speed of light — and f is the frequency of the wave without doppler shift. For an application like OLFAR, a possible frequency f could be S-band at 3GHz which would lead to doppler frequencies of $0 < |f_d| < 10\mathrm{kHz}$.

3.4 Frequency domain two way ranging

In this section, a framework for frequency domain two way ranging is proposed by relating the transmitted and received frequencies of two nodes exchanging messages to their clock parameters and relative motion. For use in this chapter, further variables are introduced in table 3.1.

Table 3.1: Additional variables for frequency domain methods

variable	unit	description
f	Hz	true frequency
f_i	Hz	local frequency at node i
f_d	Hz	doppler frequency
f_t	Hz	transmitted frequency
f_r	Hz	received frequency
v_d	$\frac{\mathrm{m}}{\mathrm{s}}$	doppler velocity
c	$\frac{\mathrm{m}}{\mathrm{s}}$	wave propagation speed
δf_d	Hz	doppler frequency accuracy
Δf_d	Hz	doppler frequency resolution
f_s	Hz	sampling frequency
f_{max}	Hz	maximum frequency
T_{meas}	s	measurement period
$F_{ij,k}$	Hz	frequency stamp for the k-th transmission recorded at node i
$F_{ji,k}$	Hz	frequency stamp for the k-th transmission recorded at node j

A relation between the clock parameters of an arbitrary node i and the frequency f_i it generates is established. First, the frequency of an electromagnetic wave is related to its period, Δt_i, as

$$f = \frac{1}{\Delta t}, \qquad f_i = \frac{1}{\Delta t_i}. \tag{3.3}$$

Secondly, let us express this time delay as a function of the local time delay using equation (2.3) for the affine clock model:

$$\Delta t = \mathcal{C}_i\left(t_i + \Delta t_i\right) - \mathcal{C}_i\left(t_i\right) = \alpha_i(t_i + \Delta t_i) + \beta_i - \alpha_i t_i - \beta_i = \alpha_i \Delta t_i \tag{3.4}$$

As expected, when trying to generate the wave with period Δt_i, the true period will be Δt dependent on the clock skew, not the offset. Thirdly, we can express this relation in terms of frequencies and clock skew ω_i as

$$f = \omega_i f_i \qquad \Leftrightarrow \qquad f_i = \omega_i^{-1} f. \tag{3.5}$$

Equation (3.5) shows that node i set to generate the frequency f_i, the clock skew/frequency offset ω_i at the node affects the true frequency f generated at the node. As expected, the clock offset does not influence the resulting frequency. After deriving the relation of frequency and skew for a single node, the relation is used in a pairwise communications scenario.

To illustrate which frequencies can be measured in a pairwise communication scenario, figure 3.1 describes this process in seven steps for a message sent from node i to j. First, a local nominal frequency is selected at the transmitting node i, where subscript k indicates the index of the message. The physical frequency is then affected by the clock skew ω_i as described in (3.5). As mobile nodes are considered, the transmission adds a doppler shift between the nodes, leading to the received frequency of transmitted and doppler frequencies as per equation (3.1). Sensing the frequency of the received signal at receiving node j is again subject to the local skew. This leads to the perceived frequency at the receiving node being a function of both nodes' skews, the doppler shift and the local transmitter frequency.

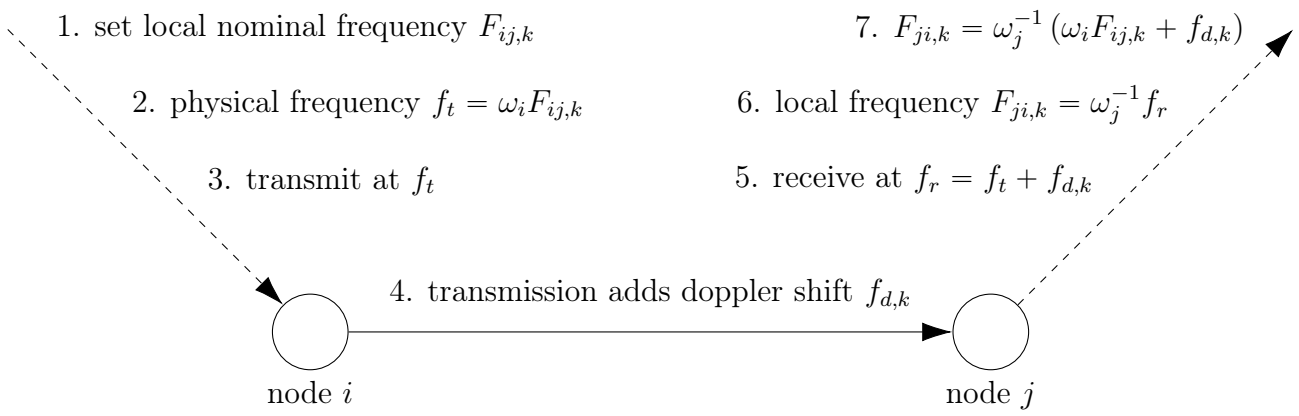


Figure 3.1: Doppler effect on transmission from node i to j

Rewriting the equation in (7.) for bi-directional communication, the doppler frequency can be expressed as a function of the local frequency measurements and clock skews

$$E_{ij,k} f_{d,k} = \omega_j F_{ji,k} - \omega_i F_{ij,k} \qquad \text{for } i \leftrightarrow j, \tag{3.6}$$

where $E_{ij,k} = +1$ for transmission from i to j and $E_{ij,k} = -1$ for transmission from j to i. In real world applications, the measured frequencies are not only dependent on skews and doppler shift, but there is also noise due to higher order motion, quantization errors, etc. The noise is assumed to be zero mean Gaussian, a choice that is motivated further in chapter 4. Thus equation (3.7) can be extended by the noise term

$$\boxed{E_{ij,k} f_{d,k} + \omega_i F_{ij,k} - \omega_j F_{ji,k} = \eta_k \qquad \text{for } i \leftrightarrow j}, \tag{3.7}$$

where

$$\eta_k \sim \mathcal{N}(0, \sigma^2). \tag{3.8}$$

In preparation of formulating clock skew estimators in the following section, the scalar frequency measurements of K transmissions can be stacked into a measurement vector as

$$\mathbf{f}_{ij} = [F_{ij,1}, F_{ij,2}, ..., F_{ij,K}]^T \in \mathbb{R}^{K \times 1}. \tag{3.9}$$

3.5 Frequency domain methods

3.5.1 Uni-directional Frequency Least Squares

In this subsection, an estimator for uni-directional skew and doppler least squares estimation is proposed.

Given equation (3.2), one can rewrite equation (3.7) in terms of the doppler velocity. The resulting equation only holds for communication in one direction, because the relation of doppler velocity and frequency is dependent on the transmitted frequency — here $\omega_i F_{ij,k}$. It will be extended to bi-directional transmission in the next subsection.

$$\frac{v_d}{c}\omega_i F_{ij,k} + \omega_i F_{ij,k} - \omega_j F_{ji,k} = \eta_k \qquad \text{for } i \to j \tag{3.10}$$

For this estimator we furthermore assert that the relative speed between the two nodes stays constant during all K transmissions as

$$v_d = v_{d,k} \; \forall \; k. \tag{3.11}$$

Then, equation (3.10) can be rewritten in vectorial notation for K communications as

$$[-\mathbf{f}_{ji} \quad \mathbf{f}_{ij}] \begin{bmatrix} \omega_j \\ \omega_i(1 + c^{-1}v_d) \end{bmatrix} = \boldsymbol{\eta}_{ij} \qquad \text{for } i \to j. \tag{3.12}$$

Let then node i be the clock reference, hence $\omega_i^{-1} \triangleq 1$. The equation simplifies to

$$\begin{bmatrix} -\mathbf{f}_{ji} & c^{-1}\mathbf{f}_{ij} \end{bmatrix} \begin{bmatrix} \omega_j \\ v_d \end{bmatrix} = \boldsymbol{\eta}_{ij} - \mathbf{f}_{ij} \qquad \text{for } i \to j \tag{3.13}$$

$$\boxed{\begin{bmatrix} \hat{\omega}_j \\ \hat{v}_d \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ji} & -c^{-1}\mathbf{f}_{ij} \end{bmatrix}^{\dagger} \mathbf{f}_{ij} \qquad \text{for } i \to j} \tag{3.14}$$

and can be solved for the clock skew of node j and the relative velocity if these requirements are met:

1. Messages for frequency measurements are sent in one direction only. For bi-directional transmission, the equations do not hold as the doppler frequency is dependent on the transmission frequency.

2. The number of messages fulfills $K \geqslant 2$, for lower number of messages the system of linear equations would be under-determined.

3. The transmit frequencies stacked in $\mathbf{f}_{ij}$ must be different from one another, more precisely at least $K \geqslant 2$ different frequencies need to be used. Otherwise, multiple rows would be equal and the system of linear equations would be under-determined.

It was shown in this subsection that for pairwise communication, with a different frequency for each transmission, uni-directional transmission, and assuming the transmitter as reference clock, the clock skew of the receiving node and the radial velocity between the two nodes can be estimated.

This method — like all frequency domain methods presented in this section — decouples clock skew and radial velocity from offset and distance. When clock offset and skew are unknown, time domain methods cannot decouple both and need to jointly estimate both clock parameters which requires bi-directional messaging as shown in equation (2.18). While this uni-directional frequency domain method cannot estimate clock offset and distance, it can estimate skew and velocity with offset and distance being unknown, an ability that extends beyond that of prevalent time domain methods. The uni-directional approach might make it unsuitable for use in anchorless satellite networks as prevalent methods relay on bi-directional communication.

However, in scenarios with broadcast communication the transmitter could with only two messages allow all receivers to synchronize their clock skews to the transmitter and measure their relative velocities w.r.t the transmitter. A possible application could be water current measurement with a wireless sensor network, where the approximate sensor locations are known and the relative velocities to the base station are of interest. In addition, any broadcast time synchronization application where the offset is either known a priori or not of interest can make use of this method.

The uni-directional frequency pairwise least squares is a special case of frequency domain least squares estimators. As its limitation to uni-directional messaging can be disadvantageous for time synchronization in anchorless satellite networks, this method will not be used any further in this work. In pursuit of a more general solution, the extension to bi-directional messaging will be explored in the next section.

3.5.2 Frequency Pairwise Least Squares

In this subsection the previous solution is extended to bi-directional messaging between the nodes, proposing the Frequency Pairwise Least Squares (FPLS) algorithm. It is the frequency domain equivalent to the time-domain Low Complexity Least Squares in the sense that it operates on the lowest order quantity in frequency domain.

In the previous subsection, equation (3.10) described the relation of frequency measurements, clock skews and radial velocity for communication in direction $i \to j$, which in opposite direction is expressed as

$$-\frac{v_d}{c}\omega_i F_{ji,k} + \omega_i F_{ij,k} - \omega_j F_{ji,k} = \eta_k \qquad \text{for } i \leftarrow j. \tag{3.15}$$

To describe an algorithm based on bi-directional messaging, equations (3.10, 3.15) are combined to a formulation universal w.r.t transmission direction

$$\frac{v_d}{c}\left(G_{ij,k}\omega_i F_{ij,k} - G_{ji,k}\omega_j F_{ji,k}\right) + \omega_i F_{ij,k} - \omega_j F_{ji,k} = \eta_k \qquad \text{for } i \leftrightarrow j, \tag{3.16}$$

where the binary variable

$$G_{ij,k} = \begin{cases} 1 & i \to j \\ 0 & i \leftarrow j \end{cases}, \qquad G_{ji,k} = \begin{cases} 0 & i \to j \\ 1 & i \leftarrow j \end{cases}, \tag{3.17a}$$

$$\mathbf{g}_{ij} = [G_{ij,1}, G_{ij,2}, ..., G_{ij,K}]^T \in \mathbb{Z}_2^{K\times 1} \tag{3.17b}$$

denotes an indicator. Rewriting this for K communications

$$\begin{bmatrix} \mathbf{f}_{ij} & -\mathbf{f}_{ji} & c^{-1}\mathbf{g}_{ij} \odot \mathbf{f}_{ij} & -c^{-1}\mathbf{g}_{ji} \odot \mathbf{f}_{ji} \end{bmatrix} \begin{bmatrix} \omega_i \\ \omega_j \\ v_d\omega_i \\ v_d\omega_j \end{bmatrix} = \boldsymbol{\eta}_{ij} \tag{3.18}$$

and then assuming node i as clock reference, hence $\omega_i^{-1} \triangleq 1$, the equation simplifies to

$$\begin{bmatrix} -\mathbf{f}_{ji} & c^{-1}\mathbf{g}_{ij} \odot \mathbf{f}_{ij} & -c^{-1}\mathbf{g}_{ji} \odot \mathbf{f}_{ji} \end{bmatrix} \begin{bmatrix} \omega_j \\ v_d \\ v_d\omega_j \end{bmatrix} + \mathbf{f}_{ij} = \boldsymbol{\eta}_{ij}. \tag{3.19}$$

Again, assuming the noise vector $\boldsymbol{\eta}_{ij}$ to be i.i.d. zero mean Gaussian, equation (3.19) can be solved in a least squares sense for the clock skew $\hat{\omega}_j$ of node j and the relative velocity $\hat{v}_d$ and their product as

$$\boxed{\begin{bmatrix} \hat{\omega}_j \\ \hat{v}_d \\ \hat{v}_d\hat{\omega}_j \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{ji} & -c^{-1}\mathbf{g}_{ij} \odot \mathbf{f}_{ij} & c^{-1}\mathbf{g}_{ji} \odot \mathbf{f}_{ji} \end{bmatrix}^{\dagger} \mathbf{f}_{ij}}. \tag{3.20}$$

For this equation to hold the following requirements must be met:

1. Messages for frequency measurements must be sent in both directions. Otherwise, one of the $\mathbf{g}$ vectors will be all zeros and the matrix to be inverted will be rank deficient. For uni-directional measurements, the estimator presented in equation (3.14) in the previous subsection can be applied.

2. The number of messages fulfills $K \geqslant 3$, for lower number of messages the system of linear equations would be under-determined.

3. Requirement 1 and 2 imply that at least one node must transmit at least two messages to the other. The transmit frequencies of these two messages have to be different. For example, it is allowed for node i to transmit twice to node j and node j once to node i, where i uses frequencies f_1 and f_2 and j reuses f_1 or f_2. Would i use the same frequency twice, the system of linear equations would be under-determined. After this rule is fulfilled for $K = 3$ transmissions, the other transmissions can reuse any frequencies, meaning a minimum of two different frequencies are required for this approach. However, it is anticipated that using different frequencies for all communications might deliver better results.

It was shown in this subsection that for pairwise communication, with at least two different frequencies used for transmission, bi-directional transmission, and assuming one node as reference clock, the clock skew of the receiving node and the relative velocity between the two nodes can be estimated using the proposed Frequency Pairwise Least Squares (FPLS).

The synchronization algorithms based on frequency-domain measurements differ from the time-domain methods in several ways. They offer the decoupling of the clock skew and clock offset: The clock offset cannot be estimated, but the clock skew can be estimated without concern for the offset. Similarly, it allows a direct estimation of the relative velocity, while being unable to obtain the propagation delay. For some applications where offset and delay/distance are not of interest, using the frequency domain method can be beneficial over using time domain methods. Furthermore, the proposed FPLS allows for clock skew estimation using a constant velocity model using $K = 3$ communications. With prevalent methods, namely MPLS, a minimum of $K = 4$ messages would be required for using a constant velocity motion model. At least for clock skew and pairwise velocity estimation one goal of this thesis — reducing the number of pairwise communication to estimate clock parameters — has been achieved.

Compared to the previous subsection, the bi-directional formulation allows for information from messages sent in both directions to be used. Furthermore, it should be noted, that no constraints are levied on the order of transmissions or the number of transmissions per direction (apart from having at least one transmission in the opposite direction). That makes the FPLS very versatile, e.g. one could imagine applications where the synchronization messages 'piggy-back' on existing communications messages.

The attribute of frequency domain methods to only operate on skew and velocities are a limitation with respect to the application of anchorless networks of mobile asynchronous nodes where it is required to estimate both clock skews and offsets of the nodes in the network. Thus, the frequency domain algorithms like FPLS can never fully replace those algorithms based on time measurements. Instead, they should be employed to augment and improve the performance of time domain algorithms. What makes this idea so intriguing is the fact that one would gain two additional measurements per transmission without sending extra messages at the cost of additional processing requirements. The possible fusing of time and frequency information for time synchronization will be examined in section 3.6 of this chapter.

Another possible limitation of FPLS is that its model assumes constant velocity between the nodes during the message exchange. Compared to the most basic bi-directional algorithm in time domain, LCLS, FPLS has the advantage that it operates inherently one order higher i.e. with constant velocity assumption instead of constant delay assumption. Even so, for non-linear pairwise motion, extending FPLS to a higher order motion model could lead to better clock parameter estimation accuracy in scenarios with non-linear motion. Such a method will be proposed in the next subsection.

3.5.3 Higher Order Frequency Pairwise Least Squares

In this subsection, the previously presented FPLS solution is extended to a mobile scenario resulting in the Higher Order Frequency Pairwise Least Squares (HFPLS). The need for such a formulation arises from the fact that in the previous subsections constant relative velocity

between the two nodes to be synchronized was assumed for the synchronization period. For the applications of interest the relative movement of nodes is inherently non-linear, which motivates a higher order model for relative velocity between the nodes.

In [3, ch. 4], a stationary algorithm in time domain (LCLS) was extended to a mobile formulation (MPLS) using a Taylor expansion on the delay. It was shown in that work that in a network of nodes in linear motion (and consequently non-linear pairwise motion, see also Appendix B of this thesis), a mobile algorithm operating on a motion model of acceleration order leads to an improvement in clock parameter estimation over lower order models. To achieve such improvements in time domain, HFPLS employs a motion model with constant acceleration. It is yet unknown whether model orders beyond acceleration order will be beneficial for the application at hand, but should that be required in the future, the formulation from this section can easily be extended to a model of arbitrary order.

To derive the HFPLS, a similar approach as in [3, ch. 4] will be followed here, albeit in frequency domain.

Let the velocity between two nodes i and j as a function of true time, initial velocity and acceleration be denoted as

$$\mathcal{V}_{ij}(t) \approx v_{ij}^{(0)} + v_{ij}^{(1)} t, \tag{3.21}$$

where $v_{ij}^{(\cdot)}$ are velocity and acceleration. Using equation (3.2), the doppler frequency can be expressed as a function of frequency and time as

$$f_d(f,t) = c^{-1} f \mathcal{V}_{ij}(t) \tag{3.22}$$

$$f_d(f,t) \approx c^{-1} f \left(v_{ij}^{(0)} + v_{ij}^{(1)} t \right). \tag{3.23}$$

Given that the true times t and frequencies f at the nodes will be initially unknown, it is desirable to express (3.23) in terms of measured quantities i.e. local perceived times and frequencies t_i, f_i. Using (2.3, 3.5) the doppler frequency can be expressed in terms of the local variables

$$f_d(f_i,t_i) = c^{-1} \omega_i f_i \mathcal{V}_{ij}(\mathcal{C}_i(t_i)). \tag{3.24}$$

Next, one can define

$$\mathcal{P}_{ij}(t_i) \triangleq \mathcal{V}_{ij}(\mathcal{C}_i(t_i)) \tag{3.25}$$

$$\mathcal{P}_{ij}(t_i) \approx \psi_{ij}^{(0)} + \psi_{ij}^{(1)} t_i, \tag{3.26}$$

where the translated velocity parameters $\psi_{ij}^{(\cdot)}$ are uniquely related to the velocity parameters (see [3, p. 98+99]). Thus, (3.16) can be expressed in terms of the translated velocity parameters $\psi_{ij}^{(\cdot)}$:

$$\omega_i F_{ij,k} - \omega_j F_{ji,k} + (G_{ij,k}\omega_i F_{ij,k} - G_{ji,k}\omega_j F_{ji,k}) \left(\psi_{ij}^{(0)} + \psi_{ij}^{(1)} T_{ij,k} \right) = \eta_k \tag{3.27}$$

Next, this equation can be extended to K communications

$$\boxed{\begin{bmatrix}\mathbf{B}_{ij,1} & \mathbf{B}_{ij,2}\end{bmatrix}\begin{bmatrix}\omega_i \\ \omega_j \\ \\ \omega_i\psi_{ij}^{(0)} \\ \omega_i\psi_{ij}^{(1)} \\ \\ \omega_j\psi_{ij}^{(0)} \\ \omega_j\psi_{ij}^{(1)}\end{bmatrix} = \boldsymbol{\eta}_{ij}}, \tag{3.28}$$

where

$$\mathbf{B}_{ij,1} = \begin{bmatrix}\mathbf{f}_{ij} & -\mathbf{f}_{ji}\end{bmatrix} \in \mathbb{R}^{K\times 2} \tag{3.29}$$

$$\mathbf{B}_{ij,2} = c^{-1}\begin{bmatrix}\mathbf{Q}_{ij}\mathbf{V}_{ij} & -\mathbf{Q}_{ji}\mathbf{V}_{ij}\end{bmatrix} \in \mathbb{R}^{K\times 2L} \tag{3.30}$$

$$\mathbf{Q}_{ij} = \operatorname{diag}\left(\mathbf{g}_{ij}\odot\mathbf{f}_{ij}\right) \in \mathbb{R}^{K\times K} \tag{3.31}$$

$$\mathbf{Q}_{ji} = \operatorname{diag}\left(\mathbf{g}_{ji}\odot\mathbf{f}_{ji}\right) \in \mathbb{R}^{K\times K} \tag{3.32}$$

$$\mathbf{V}_{ij} = \begin{bmatrix}\mathbf{1}_K & \mathbf{t}_{ij}\end{bmatrix} \in \mathbb{R}^{K\times 2}. \tag{3.33}$$

Similar to other pairwise methods node i is assumed as clock reference, hence $\omega_i^{-1} \triangleq 1$. Equation (3.28) simplifies to

$$\begin{bmatrix}-\mathbf{f}_{ji} & \mathbf{B}_{ij,2}\end{bmatrix}\boldsymbol{\theta}_{ij,F} + \mathbf{f}_{ij} = \boldsymbol{\eta}_{ij} \qquad \Leftrightarrow, \tag{3.34}$$

with

$$\boldsymbol{\theta}_{ij,F} = \begin{bmatrix}\omega_j & \boldsymbol{\psi}_{ij}^T & \omega_j\boldsymbol{\psi}_{ij}^T\end{bmatrix}^T \in \mathbb{R}^{5\times 1} \tag{3.35}$$

$$\boldsymbol{\psi}_{ij} = \left[\psi_{ij}^{(0)}, \psi_{ij}^{(1)}\right]^T \in \mathbb{R}^{2\times 1}. \tag{3.36}$$

The least squares solution for $\boldsymbol{\theta}_{ij,F}$ is then

$$\boxed{\hat{\boldsymbol{\theta}}_{ij,F} = \begin{bmatrix}\mathbf{f}_{ji} & -\mathbf{B}_{ij,2}\end{bmatrix}^{\dagger}\mathbf{f}_{ij}}. \tag{3.37}$$

The first two entries of this vector are the clock skew of node j and the the relative velocity at $t = 0$ as $v = \psi_{ij}^{(0)} = \psi_{ji}^{(0)}$.

For this equation to hold the following requirements must be met:

1. As already established for the constant velocity bi-directional case, the messages need to be sent in both directions, otherwise one of the $\mathbf{Q}$ matrices would be all zeros and thus $\mathbf{B}_{ij,2}$ would be rank-deficient.

2. The number of messages fulfills $K \geqslant 5$, for lower number of messages the system of linear equations would be under-determined.

3. From requirements 1 and 2 it follows that from the set of all K transmissions a subset of 5 transmissions must fulfill the following conditions: The messages must be sent in both directions, and, the messages transmitted from each node must be transmitted at different frequencies. Splitting the transmissions as equal as possible between the nodes reduces the number of frequencies required.

   For example, if one exchanges the minimum number of messages required, i.e. $K = 5$, and one chooses node i to transmit $K_i = \lceil \frac{K}{2} \rceil$ and node j to transmit $K_j = \lfloor \frac{K}{2} \rfloor$ messages, where $K = K_i + K_j$, then all K_i transmit frequencies must be different from another. Similarly, all K_j transmit frequencies must different from each other, however, they may be a member of the set of K_i transmit frequencies. After this rule is fulfilled for $K = 5$ transmissions, any further transmissions can reuse any frequencies, meaning a minimum of 3 different frequencies are required for this approach.

   However, it is anticipated that using different frequencies for all communications might deliver more accurate results.

4. In contrast to the two other frequency domain methods, HFPLS requires timestamps for the construction of matrix $\mathbf{V}_{ij}$. It is still considered a frequency domain method as it only estimates skew, velocity and higher order dynamics and does not use the time-domain two way ranging framework.

In this subsection, the FPLS operating with a constant velocity was extended to a motion model with constant acceleration. It was shown that for $K \geqslant 5$ pairwise communications with at least $\lceil \frac{K}{2} \rceil$ frequencies used for transmission, bi-directional transmission, using a higher order motion model of L-th order using timestamps and assuming one node as reference clock, the clock skew of the other node and the relative velocity and acceleration between the 2 nodes can be estimated using the proposed Higher Order Frequency Pairwise Least Squares (HFPLS).

The remarks made in the prior subsection about decoupling skew and velocities from offset and distance, invariance to order of transmissions and resulting versatility also hold for the higher order method presented in this subsections.

When comparing HFPLS to MPLS, the prevalent higher order method in time domain, some possible drawbacks become apparent. Despite being only a frequency domain method, HFPLS requires frequency and time measurements whereas MPLS only requires timestamps. If both models operate at acceleration as highest model order, a minimum number of $K = 5$ messages is required by both algorithms. The number of required communications at different model orders will be more closely examined in section 3.7. Despite these drawbacks, potential superior performance of HFPLS might justify its use over time domain methods. The performance will be assessed in the next chapter.

3.6 Multi domain methods

3.6.1 On the need of combining methods form frequency and time domain

In this section, a multi domain method for clock synchronization combining time and frequency information will be proposed.

To start with, the question about the necessity of such methods has to be answered. In the beginning of this chapter it was motivated how frequency domain methods could augment current time domain synchronization methods: In applications of interest it is typically desirable to minimize communication and computation cost to achieve a certain clock parameter estimation accuracy. In current time domain methods, the frequency information that is inherently present in each message remains unused. As a solution, three frequency domain methods were proposed — two out of which feature bi-directional messaging and are thus suitable for the applications if interest. However, as frequency domain methods can only estimate clock skew and pairwise velocity, but not clock offset and range, the frequency domain methods are incapable of replacing time domain methods. Thus, to best utilize the available information in time and frequency, fusing the information from time and frequency domain has the potential for improved performance.

For an anchorless network of mobile asynchronous nodes, in time domain, the MPLS by Rajan and Van der Veen and in frequency domain FPLS and HFPLS proposed in this chapter are the most suitable candidates. For augmenting time domain information with frequency domain information, different approaches could be used.

1. A **joint time-frequency formulation** where a single equation describes the relation between time measurements, frequency measurements, clock parameters and pairwise dynamics would be the most elegant and likely the most efficient formulation. When comparing time and frequency domain methods, it becomes apparent that the frequency domain least squares methods express the solutions in terms of clock skews ω_i, whereas the time domain methods express the solutions in terms of the calibration parameters α, β which are uniquely related as $[\alpha_i, \beta_i] \triangleq \left[\omega_i^{-1}, -\phi_i\omega_i^{-1}\right]$ (see section 2.3 for more). For a joint formulation one would need to express both relations either in terms of calibration or clock parameters. As the calibration parameters are dependent on the inverse of the clock skew and vice-versa, finding a linear equation that can be solved in least squares sense is not possible.

2. A **parallel formulation** where the tasks are split between frequency and time domain methods and executed in parallel could be useful to reduce complexity of each operation, following a similar rationale as in parallel computing. The problem here is, while the frequency domain methods lend themselves well to this approach as they are decoupling

offset and skew and are only able to estimate the latter, time domain methods jointly estimate skew and offset and they cannot be decoupled. One could of course use a time domain formulation to only estimate the offset of which a plethora exist, but neglecting the skew would come at the cost of reduced accuracy in the clock offset estimation.

3. A **combined two-step formulation** would estimate part of the quantities to be estimated in one domain, then use the results as an input to the other domain to reduce the number of unknowns in that domain. As the frequency domain method FPLS can only estimate skew and velocity it is the natural candidate for the first step. In a second step, on could then formulate a new time domain method based on MPLS to estimate the remaining quantities, namely offset and range.

As out of the three solutions the last one is the only feasible solution, in the next subsection, we will implement this approach for FPLS as step 1 and a time domain estimator of up to velocity model order as step 2.

3.6.2 Combined Pairwise Least Squares

In this two-stepped approach, FPLS is used to estimate clock offset and skew to then use a modified MPLS of order $L = 2$ (up to velocity) to estimate distance and skew.

The novel Frequency Pairwise Least Squares algorithm described in subsection 3.5.2 can estimate the clock skew $\hat{\omega}_j$ and the pairwise range rate $\hat{v}_{ij}$. It requires at least $K = 3$ messages to be exchanged between the nodes.

The equation for the Mobile Pairwise Least Squares with $L = 2$ is

$$\begin{bmatrix} \mathbf{t}_{ij} & -\mathbf{t}_{ji} & \mathbf{1}_K & -\mathbf{1}_K & \mathbf{e}_{ij} & \mathbf{e}_{ij} \odot \mathbf{t}_{ij} \end{bmatrix} \begin{bmatrix} \alpha_i \\ \alpha_j \\ \beta_i \\ \beta_j \\ \gamma_{ij}^{(0)} \\ \gamma_{ij}^{(1)} \end{bmatrix} = 0 \tag{3.38}$$

as described in [3, ch. 4]. Similar to all pairwise methods, the calibration parameters of the reference node i are set to $\alpha_i = 1$ and $\beta_i = 0$ respectively. Next, the algorithm is modified to include prior results from frequency domain. Let $\alpha_j \triangleq \hat{\omega}_j^{-1}$ and $\gamma_{ij}^{(1)} \triangleq \hat{\gamma}_{ij}^{(1)} = c^{-1}\alpha_i \hat{v}_d$. Then, the known and unknown quantities in equation (3.38) can be separated as

$$\begin{bmatrix} -\mathbf{1}_K & \mathbf{e}_{ij} \end{bmatrix} \begin{bmatrix} \beta_j \\ \gamma_{ij}^{(0)} \end{bmatrix} = \begin{bmatrix} -\mathbf{t}_{ij} & \mathbf{t}_{ji} & -\mathbf{e}_{ij} \odot \mathbf{t}_{ij} \end{bmatrix} \begin{bmatrix} 1 \\ \hat{\omega}_j^{-1} \\ \hat{\gamma}_{ij}^{(1)} \end{bmatrix} \tag{3.39}$$

which can be solved for $\hat{\beta}_j$ and $\hat{\gamma}_{ij}^{(0)}$

$$\begin{bmatrix} \hat{\beta}_j \\ \hat{\gamma}_{ij}^{(0)} \end{bmatrix} = \left[-\mathbf{1}_K \quad \mathbf{e}_{ij}\right]^{\dagger} \left[-\mathbf{t}_{ij} \quad \mathbf{t}_{ji} \quad -\mathbf{e}_{ij} \odot \mathbf{t}_{ij}\right] . \begin{bmatrix} 1 \\ \hat{\omega}_j^{-1} \\ \hat{\gamma}_{ij}^{(1)} \end{bmatrix} \tag{3.40}$$

This system of linear equations can be solved for $K \geqslant 2$ and the two communications being in opposite directions, meaning compared to the frequency domain algorithm which was a prerequisite for this formulation and required $K \geqslant 3$, no additional messages have to be sent.

In conclusion, in a combined time-frequency methods using equations (3.20) and (3.40), both clock parameters and dynamic parameters of up to velocity order can be estimated requiring only three messages between the nodes. The proposed method thus outperforms existing algorithms (MPLS with $L = 2$, $K \geqslant 4$) in terms of minimum number of communications. For this method to work, the requirements for FPLS given in the respective section must be fulfilled. In addition, on at least two of the three message exchanges, time measurements must be recorded.

Remark on matrix pseudo inversion: The matrix to be pseudo inverted in equation (3.40) only consists of -1 and 1 and does not depend on measurements. The order of messages can be chosen such that this matrix is always the same, and hence its pseudo inverse can be known beforehand to the node performing the computation. This reduces the computational effort of the node by one matrix pseudo inversion of size $K \times 2$.

3.7 Overview

In this section, an overview over all bi-directional least squares time-synchronization methods discussed in this thesis is provided in form of Table 3.2. The table compares the attributes of stationary and mobile time domain methods as well as basic and higher order frequency domain methods. For each entry, the domain(s) they operate in, the type of operating on the network (pairwise/global), the clock and pairwise dynamic parameters are estimated.

Table 3.2: Comparison of least-squares time synchronization algorithms for anchorless networks in time and frequency domain

algorithm	domain	type	K	offset	skew	distance	velocity	acceleration	source	remark
LCLS	t	p	$\geqslant 3$	✓	✓	✓	x	x	[1], [15]	MPLS with $L = 1$ equals LCLS
MPLS $L = 2$	t	p	$\geqslant 4$	✓	✓	✓	✓	x	[23]	generally $K \geqslant 2 + L$
MPLS $L = 3$	t	p	$\geqslant 5$	✓	✓	✓	✓	✓	[3, ch. 4]	
FPLS	f	p	$\geqslant 3$	x	✓	x	✓	x	subsection 3.5.2	HFPLS with $L = 1$ equals FPLS
HFPLS $L = 2$	$f^{\dagger}$	p	$\geqslant 5$	x	✓	x	✓	✓	subsection 3.5.3	
CPLS	t & f	p	$\geqslant 3$	✓	✓	✓	✓	x	subsection 3.6.2	

Legend: L = model order K = number of pairwise communications, t = time domain, f = frequency domain, p = pairwise, g = global, ✓= algorithm estimates this quantity, x = algorithm does not estimate this quantity, * = dependent on model order L, † = uses time domain measurements for higher order model, but does not do time domain estimations, LCLS = Low Complexity Least Squares, MPLS = Mobile Pairwise Least Squares, FPLS = Frequency Pairwise Least Squares, HFPLS = Higher Order Frequency Pairwise Least Squares, CPLS = Combined Pairwise Least Squares

3.8 Summary

This chapter proposed a novel method of augmenting time domain clock synchronization algorithms with frequency domain measurements. In other wireless sensing applications or timing and ranging applications, frequency domain information is frequently used to augment time domain information. For anchorless networks of asynchronous mobile nodes, prevalent methods only use time domain information, which leaves the frequency information present in the electromagnetic wave unexploited. In line with the research goal of this thesis it is thus proposed to decrease communication and computation cost by incorporating frequency domain information. The doppler effect and some basic relations are introduced to then derive a frequency domain two-way ranging framework.

Based on that, three novel frequency domain methods are proposed. First, the uni-directional frequency pairwise least-squares method that under the assumption of constant pairwise velocity estimates clock skew and radial velocity using at least two messages, all with the same direction of transmission. Due to its limitation to uni-directional transmission, this method will not be further pursued in this thesis, however it might be of high interest for some broadcast applications. Second, the bi-directional Frequency Pairwise Least Squares (FPLS) was pro-

posed. Again under the assumption of constant pairwise velocity it estimates clock skew and radial velocity using at least three messages, of which at least one must be sent in the opposite direction of transmission. Its bi-directionality makes it more versatile and thus more useful to the applications in the scope of this thesis. Thirdly, with the Higher Order Frequency Pairwise Least Squares an extension of FPLS to a higher order motion model with constant acceleration is proposed, catering to non-linear pairwise motion which can be encountered even when the motion of individual nodes is governed by a linear model. HFPLS can estimate clock skew, velocity and acceleration with $K \geqslant 5$ messages. Frequency domain methods are inherently unable to estimate clock offset and range, thus for effective use in clock synchronization, they must be used in conjunction with time domain methods. Hence, a novel, two-stepped method named CPLS is proposed. In a first step it uses FPLS to estimate clock skew and radial velocity to then, in a second step it incorporates these estimates as prior information into a time domain mobile formulation, which with no extra messages to be sent estimates clock offset and range. To provide a comparison between existing time domain least squares methods and novel frequency and multi domain methods proposed in this chapter an overview table is included.

Finally, the novel frequency and multi domain synchronization methods proposed in this chapter constitute a key contribution of this thesis. It was shown that for constant pairwise velocity, the proposed CPLS multi domain method reduces the number of required messages to $K \geqslant 3$ compared to the prevalent MPLS time domain method with $K \geqslant 4$. The following chapter addresses verification and application aspects of these methods.

4 | Measurements and Simulation

4.1 Introduction

In this chapter, the topic of measuring in time and frequency domain will be addressed, before designing a simulation architecture and executing simulations to assess the performance of the novel algorithms. In the pairwise algorithms the timestamp or, in case of the novel class of algorithms, frequency stamps are the basis for the computation. Recording those time and frequency measurements at the receiver is itself an estimation problem, and the accuracy of these measurements has an effect on the accuracy of the clock parameter estimation. The process of measuring both quantities is addressed in section 4.2, where particular emphasis is put on frequency measurements in subsection 4.2.2, as the new class of frequency augmented algorithms brings the extra requirement on the receiver electronics to sense the received frequency. Besides, the topic of how to model noise on time and frequency markers and how to relate their magnitudes is addressed. Next, in section 4.3, a simulation is conducted to firstly verify the function of the proposed class of frequency domain and multi domain algorithms to then, secondly, compare the performance of the novel algorithms to prevalent algorithms. In subsection 4.3.1, the architecture of the simulation is illustrated by a detailed representation of the computer program in A. Subsection 4.3.2 briefly lays out the parameter choice for the simulation leading to the results presented and analyzed in subsection 4.3.3.

4.2 Measurements

In prior work on time synchronization in anchorless networks of mobile nodes, the issue of how timestamps are obtained was not covered [3, ch. 4], [2]. However, the correct measurement of times (and frequencies) is detrimental to the successful application of any synchronization method. Therefore, this section is used for some brief remarks on the matter of obtaining measurements, starting with the time measurements.

4.2.1 Time measurements

As was mentioned in section 2.4, one can distinguish between application and physical layer time stamping. For the scope of this thesis, physical layer time stamping is assumed. While

more complex implementation-wise, physical time stamping allows for more accurate time measurements than application layer time stamping as it is not subject to non-deterministic queuing delays [15], [21]. As was shown in equation (2.12), the remaining total delay is comprised of transmission delay, propagation delay and reception delay. It is assumed this total delay τ has a dominant deterministic component that can be estimated. To measure this delay in the timestamps on transmitter the time of transmission must be recorded, which is straightforward. On the receiver, the reception time must be measured to obtain a time-of-flight (TOF) measurement.

TOF measurement is essentially an estimation problem, where methods like envelope moment method using Hilbert transformation or cross-correlation can be used to estimate the time of arrival. Going more in depth on these methods would go beyond the scope of this work.

4.2.2 Frequency measurements

In this subsection, the implementation aspect of frequency measurements will be addressed. For the frequency domain methods, the measurements $F_{ij,k}$ and $F_{ji,k}$ need to be recorded. Transmission from node i to node j is assumed in this section, meaning $F_{ij,k}$ is the frequency selected at the transmitter and $F_{ji,k}$ the frequency measured at the receiver.
So far, the frequency selected at the transmitter has only been represented by a single variable $F_{ij,k}$, however, it can be broken down into a baseband frequency f_b and carrier frequency f_c such that

$$F_{ij,k} \triangleq f_b + f_c. \tag{4.1}$$

It is assumed that the frequencies are known to both nodes. As both frequencies are synthesized in node i, they are exposed to its clock skew ω_i resulting in

$$f_t \triangleq \omega_i F_{ij,k} = \omega_i \left(f_b + f_c\right), \tag{4.2}$$

the frequency physically transmitted from node i. This is illustrated in figure 4.1.

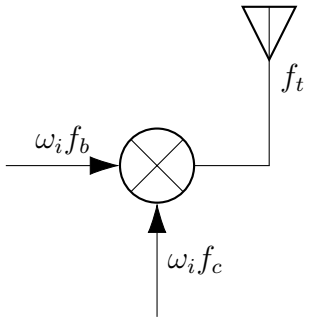


Figure 4.1: Transmitter architecture for transmission from node i

For measuring $F_{ji,k}$, two methods — digital and analogue — are proposed.

First, the digital receiver involves analogue to digital conversion (ADC) as well as discrete fourier transformation (DFT) of the digitized signal. The fidelity of the measured signal depends on the sampling rate of the ADC, the measurement time & SNR of the time domain signal and the number of points of the fast fourier transformation (FFT).

To analyze the effects of these parameters the conceptual depiction of the transmission process (figure 3.1) is taken to a high-level hardware architecture of transmitter and receiver in figure 4.2.

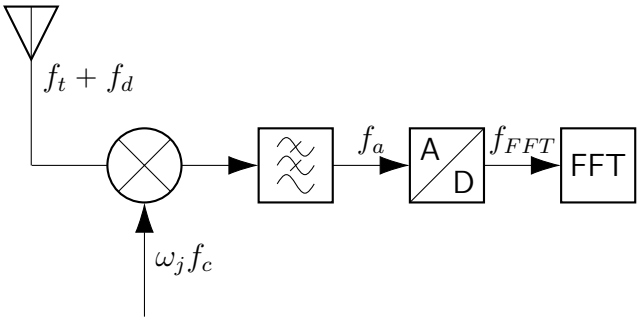


Figure 4.2: Receiver architecture for transmission from node i to node j to sense the frequency of an incoming signal using in the digital domain. Arrows left of the ADC represent analogue signals, arrows right of the ADC digital signals.

The transmitter and receiver architecture only depicts elements that are relevant to the time synchronization problem, and for simplicity not all those that are relevant in a wireless communication system (e.g. baseband signal synthesis, IQ mixer, ...). However, the block diagram can be extended to fulfil all functions of a wireless communications system and conclusions about the time synchronization aspects should still hold.

The frequency received by the antenna of node j is thus $f_t + f_d$. After demodulation into baseband (which suffers from the clock skew at node j) and a lowpass filter at the receiver, the analogue frequency can be expressed as

$$\begin{aligned} f_a &\triangleq f_t + f_d - \omega_j f_c \\ f_a &= \omega_i \left(f_b + f_c\right) + f_d - \omega_j f_c \end{aligned} \tag{4.3}$$

which is then digitized by the ADC which also suffers from clock skew ω_i. On the resulting digitized signal with frequency

$$\begin{aligned} f_{FFT} &\triangleq \omega_j^{-1} f_a \\ f_{FFT} &= \omega_j^{-1} \left(\omega_i \left(f_b + f_c\right) + f_d\right) - f_c, \end{aligned} \tag{4.4}$$

the FFT will be performed. The frequency measured at the receiver can then be expressed as

$$\begin{aligned} F_{ji,k} &= f_{FFT} + f_c \\ \boxed{F_{ji,k} &= \omega_j^{-1} \left(\omega_i \left(f_b + f_c\right) + f_d\right)} \end{aligned} \tag{4.5}$$

which — when rearranged — equals equation (3.10) confirming that the more in depth representation of the receiver and transmitter architecture matches the earlier representation.

A second option to sense the frequency is an analogue control loop using a phase detector as illustrated in 4.3. Similar to the digital approach, after down modulation f_a shall be sensed. A phase detector compares the difference of this signal and an artificially generated signal $\hat{f}_a$. The error signal v_e is an input to a variable frequency oscillator. The VFO generates a frequency that suffers from the local clock skew at the receiver:

$$f_{VFO} \triangleq \omega_j^{-1} f_a \tag{4.6}$$

The relation to the other quantities is the same for f_{VFO} as for f_{FFT}.

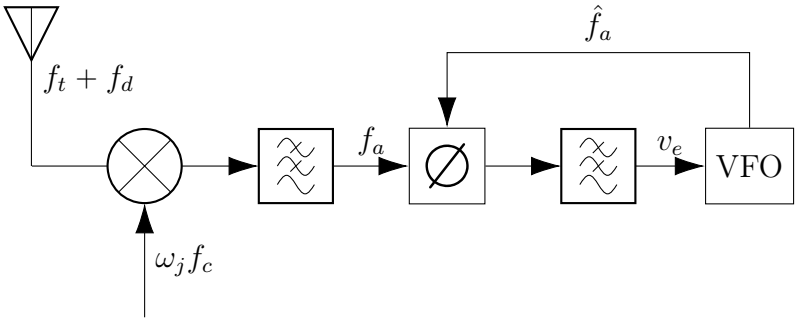


Figure 4.3: Receiver architecture for transmission from node i to node j to sense the frequency of an incoming signal using a control loop based on a phase detector

It has been illustrated in this subsection how a typical analogue and digital receiver can be used for frequency sensing. As signal processing on receivers moves increasingly into the digital domain, it is anticipated that in most wireless sensor networks in general and satellite systems in particular will use a receiver as depicted in 4.2, where the need to sense frequency would not lead to any additional hardware requirements, but rather to an additional FFT computation being required in the receiver.

4.2.3 Noise and SNR definition

In the previous chapters of this work, the time and frequency at which a message is transmitted and received were treated as scalar quantities measured at the nodes without going in detail on how they are measured. Measurement of local time and frequency at the transmitter is fairly straightforward, as it is an input parameter to when and at which frequency the message will be transmitted. Time and frequency measurements at the receiver are more involved — possible implementations were briefly outlined in the two previous subsections. Time and frequency measurements are again estimation problems, that are subject to estimation errors. The estimation performance for time and frequency of arrival depends on receiver architecture, clock hardware, ADC hardware, the estimator as well frequency, bandwidth, signal duration,

noise on the wireless channel and waveform of the signal used for transmission. Thus, the accuracy of time and frequency measurements is highly dependant on the respective communication system. Note, that time and frequency measurement accuracy might be affected differently by some system parameters, e.g. high bandwidth caters to higher TOA measurement accuracy while being less important for frequency measurements.

For an accurate representation of time and frequency measurement inaccuracies, an end-to-end design and simulation of the communication system used for message exchange would be necessary, however, this is beyond the scope this work. To nevertheless be able to verify the correct function of the algorithms under noisy time and frequency measurements, noise is added to measurements. In the remainder of this section, the type of distribution and the definition of the signal to noise ratio will be laid out. The noise on time and frequency markers is a sum of random variables, such as noise on the wireless signal, quantization errors, clock jitter and inaccuracies/non-linearities in positions and velocities of nodes. Therefore, in accordance with the central limit theorem, prior work in this area [3, ch. 4],[14], [22] the noise on time and frequency markers is assumed to be i.i.d. zero mean Gaussian.

The magnitude of noise in relation to the signal strength is defined as signal to noise ratio (SNR). As the noisy time and frequency measurements are themselves estimations rather that signals with a certain power finding a SNR definition is not straightforward. As time and frequency measurements are dependent on node distance and velocity, an SNR definition can be based on these quantities such that an increasing distance and velocity of the nodes leads to increasing noise on time and frequency markers respectively. Positions and velocities are drawn from uniform distributions, where the uniform distributions variance is

$$\sigma_s^2 = \frac{1}{12}(2a)^2, \tag{4.7}$$

where the random variable is uniformly distributed in the interval $[-a, a]$. We can then express the standard deviation of the time domain noise as

$$\sigma_t = c^{-1}\mathrm{SNR}^{-1}\sqrt{\frac{1}{12}(2x_e)^2}, \tag{4.8}$$

where SNR is the SNR in dbMeter and x_e describes the uniform distribution from which node positions are drawn. Similarly, for the frequency domain noise variance one can write

$$\sigma_f = \frac{f_{min} + f_{max}}{2}c^{-1}\mathrm{SNR}^{-1}\sqrt{\frac{1}{12}(2y_e)^2}, \tag{4.9}$$

where y_e describes the uniform distribution from which node velocities are drawn. In appendix A, program 7 provides details on the application. It should be noted that this SNR definition is only dependent on node velocities and positions whereas actually the accuracy of time and frequency measurements depends on various parameters as mentioned above. Note that particularly in simulated scenarios where there is strong mismatch between distances and velocities this SNR definition poses the risk of overly favouring either time or frequency domain methods.

4.3 Simulation

This section serves the purpose of validating the novel time synchronization methods based on frequency measurements, and, furthermore to compare their performance to existing time-domain algorithms.

4.3.1 Architecture

This subsection shows the simulation architecture. The flow chart in figure 4.4 depicts the setup of a virtual test bench where the blue blocks constitute inputs and the red blocks represent the function of the algorithms. Inputs include parameters, such as number of nodes in the network, link topology (adjacency matrix of nodes in the network) and number of pairwise communications. The number of nodes in the network is not yet of importance when only evaluating the performance of pairwise algorithms, but as this simulation setup is to be reused in chapter 5 this parameter is already included. In addition to the aforementioned parameters, the clock and position models are inputs to the simulation. For modeling the clocks, only an affine model has been realized. In terms of position models, a stationary situation as well as a linear model are considered here. The stationary model falls short of the problem set of this thesis as a network of mobile nodes is studied. Nevertheless, the stationary case is useful as a simplified case to verify the functioning of algorithms. The linear model will be used as main test model for this section. It imposes linear movement as constraint on each node, such that its position is

$$\mathbf{p}_i(t) \triangleq \mathbf{x}_i + t\mathbf{y}_i \qquad \mathbf{p}_i \in \mathbb{R}^{N_{dim}\times 1}. \tag{4.10}$$

where a three-dimensional euclidean space is considered. The Hill-Clohessy-Wiltshire (HCW) dynamic model implementation is specific for satellite networks and is thus not included in this part of the thesis but rather in chapter 5, as this more elaborate representation of the dynamics is only required for the in-depth system study.

The various inputs are used to first initialize nodes. In case of the affine clock model and linear dynamic model clock parameters, positions and velocities are drawn from uniform distributions as described in programs 2 and 3. Generally, a detailed representation of the whole simulation is included in the form of pseudo-code in Appendix A, starting with the main simulation in program 1. After the node initialization message exchange noise is added as described in program 7. Then, pre-processing is conducted to create the vectors containing time and frequency measurements of each node. Then, the various time synchronization algorithms estimate clock and ranging parameters which are subsequently evaluated by computing the root mean square error (RMSE) of the respective parameter.

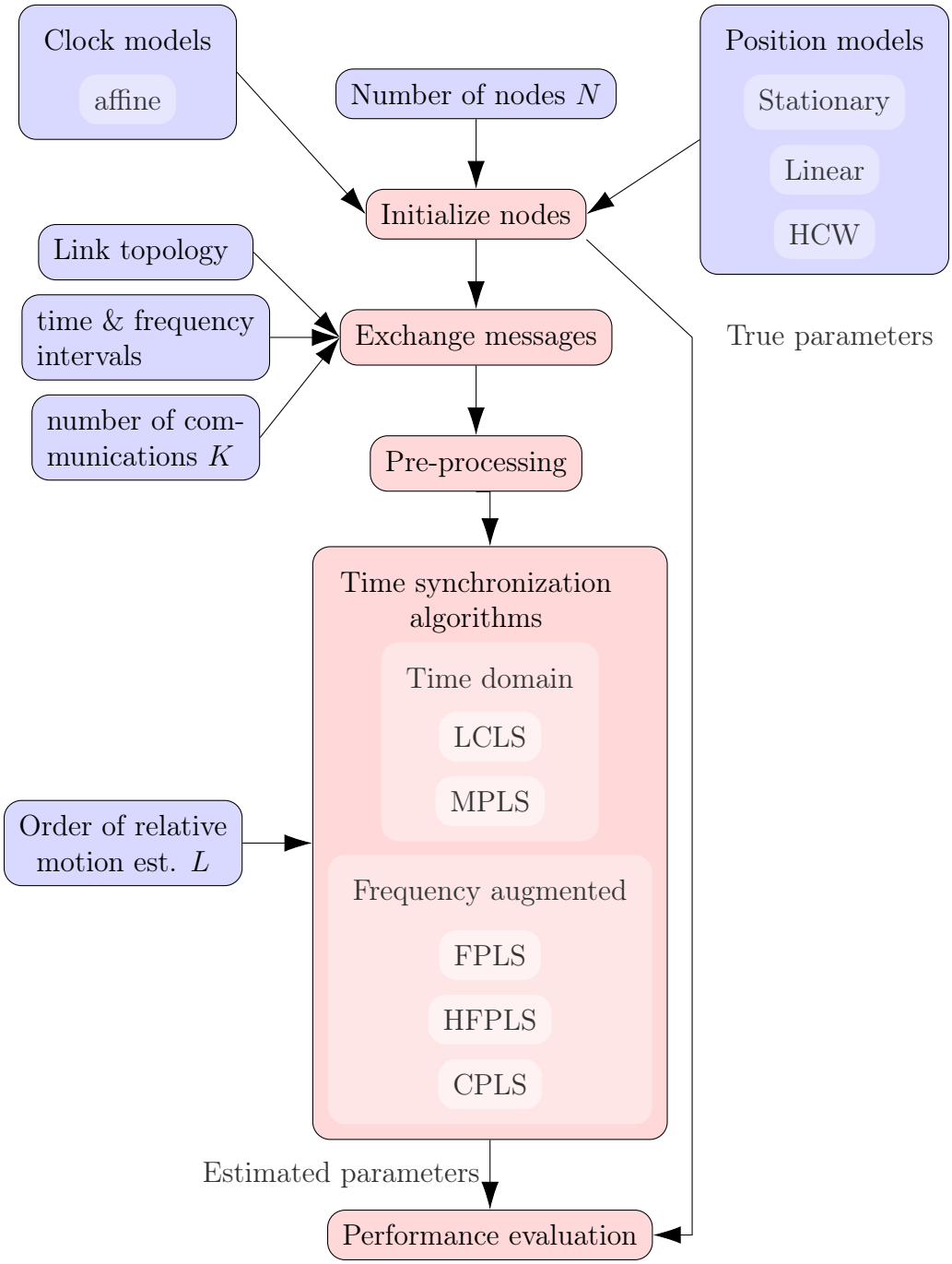


Figure 4.4: Simulation architecture for time synchronization performance comparison of different algorithms

4.3.2 Parameter choice

For this simulation, $N = 5$ nodes are simulated. Regarding the position model, the linear model with $N_{dim} = 3$ dimensions corresponding to the three spacial dimensions. Each component of the position vector is drawn from a uniform distribution where $x_e = 5\text{km}$ is the maximal absolute value and the velocity vector is drawn from a uniform distribution where $y_e = 50\frac{\text{m}}{\text{s}}$ is the maximum absolute value. The clock errors are uniformly distributed with parameters

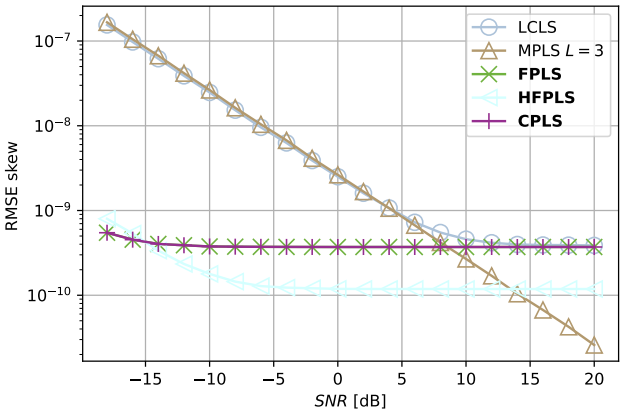
$\phi_e = 5$s on the clock offset and $\omega_e = 10^{-5}$ on the skew. The message exchange will take place between $t_{min} = 0$s and $t_{max} = 3$s. The target transmission frequencies will be varied from $f_{min} = 0.9 \times 3 \times 10^9$Hz to $f_{max} = 1.1 \times 3 \times 10^9$Hz corresponding to S-band. Default SNR is set to 0dB and the number of pairwise communications is set to K = 10 as default. For the effect of each parameter Appendix A can be consulted.

4.3.3 Results and Analysis

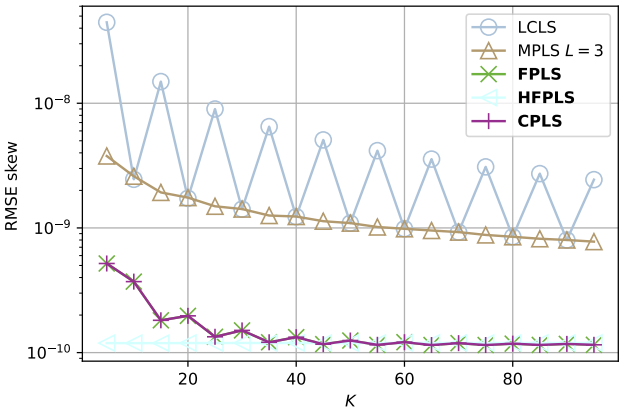
In this subsection, the results of the simulation with the aforementioned parameters are presented. A detailed look on the pairwise dynamics on the network during the simulation period is provided in Appendix C. Results of a stationary network with otherwise equal choice of parameters is included in Appendix D. Results for a mobile network for additional algorithms parameters and for distance and velocity estimation are provided in Appendix E.

The simulation was conducted for varying SNR and number of communications K to illustrate the effect of both parameters on the RMSE on the various algorithms. The left column of subfigures shows the RMSE for varying SNR and the right column for varying K. Pairwise algorithms in time and frequency domain were considered using different model orders. The novel algorithms proposed in this work are highlighted with **bold font** in the legend. Figure 4.5 shows the RMSE for clock parameters. The upper row shows the clock skew, which is estimated by all algorithms under test, thus one of the most interesting comparison metrics. In Figure 4.5a, it can be observed that all methods improve with higher SNR. Furthermore, methods with a motion model with distance as highest order (LCLS) or velocity as highest order (MPLS $L = 2$, FPLS) improve but only up to certain threshold, from where RMSE remains constant despite increasing SNR. Algorithms operating at acceleration order (MPLS $L = 3$, HFPLS) exhibit different performance. The former exhibits linear behaviour on log scale, with the error decreasing for the whole of the depicted SNR range. The latter reaches a threshold early, however, it is lower than that of the time domain methods. To look more closely at these results, a histogram of the residual error is plotted for FPLS in figure 4.6. Regarding the varying K in figure 4.5b, it can be discovered that time domain methods — and to a degree FPLS — benefit from increased K. However, for larger K, the improvement that additional number of communications can deliver decreases. It is noteworthy, that the higher order frequency domain methods do not benefit from additional number of communications K beyond their minimum required communications and the FPLS only up to $K \approx 20$. The observation that at SNR = 0dB frequency domain methods are superior to mobile time domain methods which in turn are superior over LCLS in terms of offset estimation can be confirmed for the whole K range depicted.

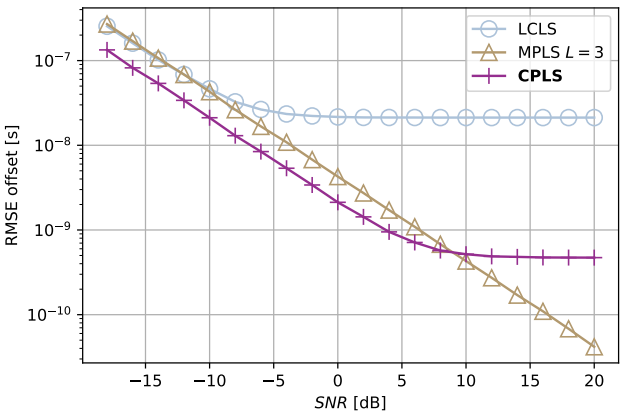
The second row of Figure 4.5 shows clock offset error. As frequency domain methods are by nature unable to estimate offset, existing time domain algorithms are represented alongside the novel proposed Combined Pairwise Least Squares (CPLS). The results for existing methods



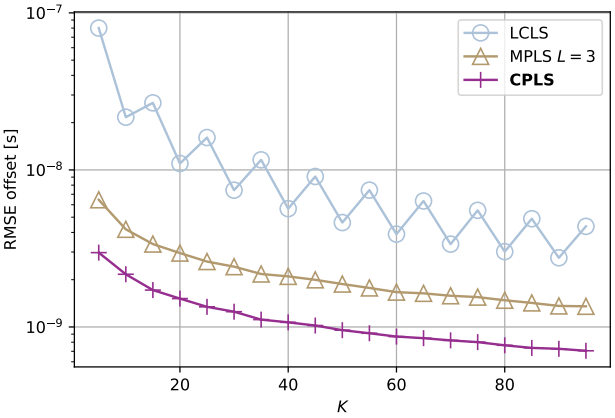
(a) clock skew ω



(b) clock skew ω



(c) clock offset ϕ



(d) clock offset ϕ

Figure 4.5: Clock parameter estimation error using pairwise algorithms time domain and frequency domain; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for SNR = 0dB

seem to confirm prior results by [3, ch. 4]. In 4.5c, it can be observed that for low and medium SNR the combined method achieves superior performance over time domain methods. As the offset and distance estimations of CPLS are based on frequency domain skew and velocity measurements, where the frequency domain methods achieve superior performance, it is plausible that these superior estimates plugged into CPLS are the cause for its superiority in offset estimation. Furthermore, the SNR from where MPLS with $L = 3$ achieves better offset estimation (7dB) coincides with that SNR where it starts to outperform FPLS/CPLS in skew estimation. Looking at the 0dB SNR cut for varying K in figure 4.5d, it can be seen that all methods benefit from increased K, however, for larger K, the improvement that additional number of communications can deliver decreases. The observation that at SNR = 0dB CPLS is superior to mobile time domain methods which in turn are superior over LCLS in terms of offset estimation can be confirmed for the whole K range depicted.

Figure 4.6 shows the histograms of the residual error for the Frequency Pairwise Least

Squares algorithm for SNR varying from -18dB to 20dB. The skew estimate errors are Gaussian distributed around a small bias in the order of 10^{-10} that seems to be the result of the non-linear pairwise motion compared to the motion model of limited order. The FPLS and HFPLS motion models assert a constant velocity and constant pairwise acceleration respectively. The actual pairwise motion features higher order non-linearities as can be seen in Appendix C. However, for HFPLS a further decrease in error for high SNR would have been anticipated as it uses the same motion model as MPLS with $L = 3$.

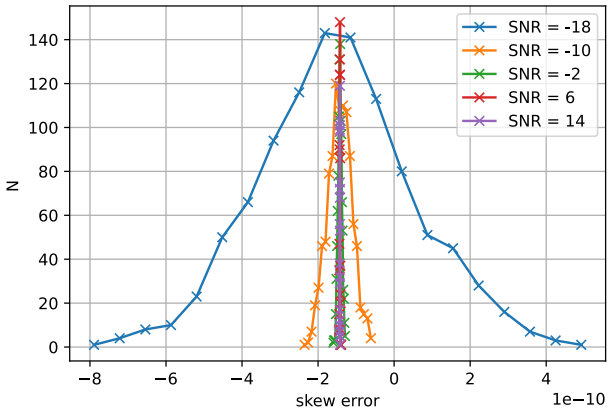


Figure 4.6: Histogram of the residual error of skew estimation for the FPLS algorithm with SNR varying from -18dB to 14dB

4.4 Summary

In this chapter, the topic of time and frequency measurements that form the basis for all pairwise synchronization algorithms was addressed. A simulation was designed and conducted in order to verify the novel algorithms and compare their performance with prevalent time domain algorithms.

Time and frequency domain measurements serve the purpose of finding the time and frequency difference between transmission and reception. Measuring the time/frequency on transmission is often straightforward as they are input parameters of the message to be transmitted. At the receiver the incoming signal with a certain duration, frequency, bandwidth and waveform can be used to estimate the time and frequency of arrival. Various methods can be used for TOA estimation in time domain. The novel frequency and multi domain methods proposed in chapter 3 require the receiver to additionally estimate frequency domain measurement. Two receiver architectures — analogue and digital — were shown to be suitable for this purpose. Still, it is anticipated that particularly the digital implementation will be used in wireless sensor networks, as it would require little to no additional hardware. As measurements will be subject to noise, a multi domain SNR definition was proposed based on the network dynamics.

A simulation architecture was devised with detailed pseudo-code in appendix A. The simulation initializes a number of nodes with clock parameters and three different position models to use. Next the message exchange is performed, where network link topology in the from of an adjacency matrix, time and frequency interval and number of communications are input parameters. After applying noise and some pre-processing, the different pairwise algorithms can be applied to the network. The estimated clock parameters were analyzed to give insight into algorithm performance. The novel proposed algorithms in frequency domain — Frequency Pairwise Least Squares (FPLS) and Higher Order Frequency Pairwise Least Squares (HFPLS) — achieve a rmse of between 2×10^{-9} to 2×10^{-10} for clock skew estimation depending on algorithm type, SNR and K for nodes in linear motion (resulting in non-linear distances and velocities). The skew estimate errors are Gaussian distributed around a small bias that seems to be the result of the non-linear pairwise motion compared to the motion model of limited order. Still, for HFPLS more analysis into the reason for the residual bias is warranted. The simulation showed that the performance of the proposed frequency domain methods in low to medium SNR range, for HFPLS even over almost the whole SNR range is significantly better than of prevalent time domain methods. Nevertheless, it is yet to early to claim outperforming time domain algorithms, as the performance depends heavily on the SNR definition. The relation of noise magnitudes on time and frequency measurements which ultimately govern the algorithms performance will always depend on the specific communications scenario. For offset estimation, the novel multi domain CPLS was successfully used. It showed similar performance in offset estimation w.r.t prevalent time domain methods as did FPLS for skew estimation which matches expectations.

Reflecting on this and the previous chapter — the main contributions of the first part of this thesis — two novel bi-directional pairwise frequency domain algorithms were proposed and successfully tested. In addition, a multi domain algorithm was proposed that can estimate clock skew and offset at 25% less communication cost, lower computation cost and likely higher accuracy than the best prevalent time domain algorithm i.e. MPLS. In the second part of this thesis, the use of prevalent and novel pairwise time synchronization methods for network time synchronization and then on mission level will be examined on the example of the OLFAR distributed interferometry mission.

5 | OLFAR Mission

5.1 Introduction

This chapter marks the beginning of the second part of this thesis. In the two coming chapters, the goal is to apply the frequency augmented synchronization techniques in a case study. For this, the OLFAR mission was chosen. OLFAR is a space-based ultra low frequency interferometer standing representative for interferometry missions. In the previous chapter, the performance of frequency-augmented time synchronization was shown for a network of nodes in linear motion. This chapter shows the feasibility of using the proposed technique in a space-based interferometry mission with multiple spacecraft. Proving the feasibility of a time synchronization method for an interferometry mission can give confidence that that this method will hold for anchorless satellite networks in general, as interferometry missions have very stringent clock synchronization requirements.

In this chapter, the OLFAR mission will be introduced. First, the general mission rationale and its science objectives will be presented in section 5.2. Second, the clock requirements of the mission will be introduced in section 5.3. Third, in section 5.4, the relative motion of OLFAR will be addressed. In the previous chapter, the performance of synchronization algorithms for nodes in linear motion was simulated, which served as suitable performance assessment of the algorithms in general. For an accurate representation, the various proposals for OLFAR orbit design will be analyzed with the goal of analytically representing the orbital dynamics of OLFAR. For that purpose, the Hill-Clohessy-Wiltshire (HCW) equations will be introduced.

5.2 OLFAR mission

OLFAR stands for Orbiting Low Frequency Array for Radio astronomy [11]. Its concept has been developed mainly in the last decade by a group of universities and research institutions in the Netherlands. They aim to augment the capabilities of the ground-based Low-Frequency Array (LOFAR) operated by the Netherlands Institute for Radio astronomy (ASTRON) [12]. The use case for OLFAR is to provide a space-borne large aperture radio interferometric array platform in a frequency range from 0.3 MHz to 30 MHz [9], [13]. By stationing OLFAR in space far away from Earth, in lunar orbit or Earth-Moon L2, interference from Earth can be

mitigated [9]. OLFAR is proposed to consist of a swarm of $\geqslant 10$ satellites [13], but the exact number of elements varies across the different publications. The performance of OLFAR as distributed radio telescope is strongly dependent on achieving time synchronization among its nodes.

5.3 Clock error requirements

In chapter 2, section 2.2, the Allan deviation as measure for higher order clock synchronization was introduced. Provided that the clock hardware chosen for the OLFAR project fulfils the Allan deviation requirement, we assume that during the coherence period τ_c, the residual synchronization errors will be dominant over higher order clock effects. For OLFAR, the requirement was set forth that during a snapshot integration time, baseline distance should not change by more than 0.1λ (observed wavelength) [32] which for $\lambda = 30$m and speed of light as propagation speed leads to a maximum total clock error of $\delta t_i = 10$ns. As this is the requirement for the snapshot integration time of 1s, achieving it for the whole synchronization period of > 1s will satisfy the clock accuracy requirement. The resynchronization threshold is chosen such that when the total clock error of the nodes in the network exceeds δt_i, a resynchronization has to be performed. The resynchronization intervals that can be achieved for the OLFAR mission will be examined in the following chapter in section 6.4.

5.4 Position model

In this section, a position model simulating the orbital dynamics of OLFAR will be introduced. First, different possible deployment locations of OLFAR will be listed and the most demanding one in terms of relative motion will be chosen for further analysis. Second, the HCW equations will be introduced to analytically describe the relative orbital motion of the OLFAR satellites. Third, following the work of Dekens et al. [32], the orbital motion of OLFAR will be described using HCW equations.

5.4.1 Deployment location

For the OLFAR mission, several different staging locations have been considered. In this subsection, the potential staging locations will be aggregated to then motivate one of them to be examined more closely in the following subsections.

Different deployment locations are discussed in [33]:

- Sun-Earth Lagrange point L2
- Sun-Earth Lagrange points L4/L5

- Earth-Moon Lagrange point L2
- Earth-Moon Lagrange points L4/L5
- Elliptic heliocentric orbit
- Circular heliocentric orbit
- Lunar orbit

For further analysis, the Lunar orbit is selected. Among the more recent publications on OLFAR, Lunar orbit has emerged as a preferred deployment location [34], largely thanks to its proximity to Earth which will minimize communication requirements for the downlink. Besides, among the possible deployment locations, the highest orbital velocities and the highest relative velocities between the nodes are expected. This means, if a satisfactory clock synchronization performance can be achieved for this scenario, it shows the feasibility also for other less demanding deployment locations.

5.4.2 Hill-Clohessy-Wiltshire equations

In this subsection, we will briefly introduce the HCW equations, a tool that uses analytical linearized description of relative orbital motion to study relative motion between closely located spacecraft. This subsection will follow the notation of Wertz in his standard work on Mission Geometry; Orbit and Constellation Design and Management [35].

The HCW equations were initially formulated by Hill in 1878 [36] to be rediscovered by Clohessy and Wiltshire for the use in relative orbital design, formation flying and spacecraft rendezvous. The approach of HCW equations is to analytically describe the relative position of a spacecraft with relation to an actual or virtual reference spacecraft. The Hill reference frame has its origin at the position of the reference spacecraft, and its axis is oriented with respect to the orbit of this reference spacecraft as along-track axis ϕ, cross-track axis z and radial axis r. The HCW equations provide an analytical solution that approximates the relative position of satellites to a virtual reference satellite and thus allows to circumvent potentially computationally intensive orbital calculations using numerical integration. For the approximations of the HCW equations to be valid, the following condition must hold

$$a \gg \sqrt{r^2(t) + \phi^2(t) + z^2(t)} \quad \forall t \leqslant t_{max} \tag{5.1}$$

where a is the distance of the (virtual) satellite at the origin of the Hill frame to the center of the celestial body it is orbiting and $r(t)$, $\phi(t)$ and $z(t)$ are the time dependent distances in its three dimensions of the other spacecraft to the origin of the Hill frame and t_{max} is the duration of the propagation simulation.

For OLFAR, the maximum baseline requirement was set to 100km [13], thus

$$\sqrt{r^2(t) + \phi^2(t) + z^2(t)} \leqslant 100\text{km}.$$

The orbit radius is composed of the radius of Moon r_M and the orbit height h_O as

$$a = r_M + h_O. \tag{5.2}$$

With $r_M = 1737.4$km, even for low orbits the condition set out in equation (5.1) will always hold for the OLFAR case.

The following equations constitute the HCW equations providing the three aforementioned position components in the Hill frame as a function of time as [35]

$$r(t) = -A\cos(nt+\alpha) + 2\left(\frac{\dot{\phi}_0}{n} + 2r_0\right) \tag{5.3a}$$

$$\phi(t) = 2A\sin(nt+\alpha) + \frac{\dot{\phi}_0 - 2\dot{r_0}}{n} - 3\left(\dot{\phi}_0 + 2n\dot{r_0}\right)t \tag{5.3b}$$

$$z(t) = A_z\sin(nt+\alpha_z) \tag{5.3c}$$

$$A = \sqrt{\left(\frac{\dot{r}_0}{n}\right)^2 + \left(\frac{2\dot{\phi}_0}{n} + 3r_0\right)^2} \tag{5.3d}$$

$$\alpha = \arctan\left(\frac{\dot{r}_0}{2\dot{\phi}_0 + 3r_0 n}\right)^2 \tag{5.3e}$$

$$A_z = \sqrt{z_0^2 + \left(\frac{\dot{z}_0}{n}\right)^2} \tag{5.3f}$$

$$\alpha_z = \arctan\left(\frac{nz_0}{\dot{z}_0}\right)^2 \tag{5.3g}$$

$$n = \frac{2\pi}{T_0} \tag{5.3h}$$

where ϕ_0, z_0, r_0 are initial positions and $\dot{\phi}_0$, $\dot{z}_0$, $\dot{r}_0$ are the initial velocities on along-track, cross-track and radial axis respectively and T_0 is the orbit period of the reference orbit. The latter can be calculated using Kepler's 3rd law

$$T_0 = 2\pi\sqrt{\frac{a^3}{G(m_1+m_2)}}, \tag{5.4}$$

which relates the duration of an orbit T_0 with the distance a of the centers of the two bodies, their masses m_i and the gravitational constant $G = 6.674\times10^{-11}\,\frac{\mathrm{m}^3}{\mathrm{kgs}^2}$. When $m_1 \gg m_2$ — which is the case for artificial satellites orbiting celestial bodies — the mass of the satellite can be neglected. For simplicity, only circular orbits will be considered in this thesis.

5.4.3 Relative Orbital motion in lunar orbit

In this subsection, the relative motion of OLFAR in Lunar orbit will be modeled, with the goal of constructing a scenario of an orbiting satellite swarm that is at least as demanding as

the OLFAR mission in terms of pairwise distances and velocities. Various works have been carried out on OLFAR orbit design, most notably by Dekens et al. in 2013 [32], van t'Hoff [37] and Mok et al. in 2020 [38]. As [37] addresses orbit design for EML4, it is not immediately applicable to this work. In [32], [38], lunar orbit design for OLFAR is explored, however, neither of those works propose an orbit design in lunar orbit that satisfies all requirements of the OLFAR mission. Dekens et al. propose orbits that fail to meet the baseline rate requirement of OLFAR. Mok et al. address this issue, proposing a new set of algebraic constraints under which the original requirements are met, but they do not propose specific orbit parameters to fulfil these. Thus, this work will follow [32] and use their approach to OLFAR orbit design. Despite its violation of some of the OLFAR mission requirements it is sufficient for the purpose of validating time synchronization techniques.

The requirements for orbit design follow from the general mission requirements and science objectives as the imaging performance is tightly linked to the relative orbital motion. OLFAR requires an angular resolution θ of 1 arcminute, following

$$\theta = \frac{\lambda}{B}, \tag{5.5}$$

where B is the largest baseline distance and λ the observed wavelength [39]. It was found by [32] that a relative orbital design where all satellites drift freely within a sphere of 100km without active orbit determinations is sufficient for achieving the baseline requirement of the mission. Furthermore, during the snapshot integration time (time over which one measurement is taken), the baseline distance should not change by more than 0.1λ (observed wavelength) which for $\lambda = 30$m leads to a resultant maximum relative velocity (baseline rate) of $3\frac{\mathrm{m}}{\mathrm{s}}$. This potentially conflicts with the coverage requirement stating that a high variation in spacial configuration over the mission duration is beneficial for imaging quality and thus desirable [32]. For a swarm in lunar orbit with the aforementioned baseline and baseline rate requirements, these two are fundamentally incompatible as shown by [32], thus a relaxation of one of the requirements might be necessary for the OLFAR.

Nevertheless, Dekens et al. proposed two orbits for OLFAR where baseline rates of $116\frac{\mathrm{m}}{\mathrm{s}}$ and $30\frac{\mathrm{m}}{\mathrm{s}}$ are achieved for orbit heights of 200km and 3000km respectively. While these orbits violate the baseline rate requirement, the increased pairwise velocity makes the time synchronization effort more difficult. The potential options to meet OLFAR mission requirements include relaxing the snapshot integration time and thus the baseline rate requirement, decreasing coverage requirements, introducing active orbit control or choosing a different staging location. All of these options would if at all lead to a decrease in pairwise velocities compared to the two orbits proposed in [32] and therefore relax the requirements for time synchronization. Hence, if the performance of time synchronization algorithms can be demonstrated for these two orbital scenarios it can be concluded that the time synchronization techniques will meet OLFAR requirements.

Dekens et al. use the HCW equations in a de-rotated Hill frame, called sky frame. Compared to the Hill frame that is oriented relative to the reference orbit, the sky frame is oriented relative to the fixed stars. In [32], this representation is used to gauge the imaging performance of OLFAR. The positions in a Cartesian coordinate systems as a function of time are given as

$$x(t) = a\left(\gamma\left(-\frac{3}{2}\cos(\xi) + \frac{1}{2}\cos(2tn - \xi)\right) - \beta\sin(tn) + \alpha\frac{3}{2}tn\sin(tn)\right) \tag{5.6a}$$

$$y(t) = a\left(\gamma\left(-\frac{3}{2}\sin(\xi) + \frac{1}{2}\sin(2tn - \xi)\right) + \beta\cos(tn) - \alpha\frac{3}{2}tn\cos(tn)\right) \tag{5.6b}$$

$$z(t) = a\delta\sin(tn - \psi) \tag{5.6c}$$

$$n = \frac{2\pi}{T_0}, \tag{5.6d}$$

where α, β, γ, δ, ξ and ψ are the relative semi-major axis, relative anomaly, relative eccentricity, relative inclination, relative periapsis and relative ascending node respectively [32], [40] and a is the radius of the circular reference orbit.
For OLFAR to meet its baseline requirements, Dekens et al. chose these parameters as shown in table 5.1, and initialized the parameters of N satellites as uniform random distributed over the allowed interval.

Table 5.1: Boundaries on relative orbital parameters taken from [32] with baseline $B = 100km$ and orbit radius a

Parameter	Minimum	Maximum
α	0	0
β	$-\frac{B}{2a}$	$\frac{B}{2a}$
γ	0	0
δ	0	$\frac{B}{2a}$
ξ	0	2π
ψ	0	2π

As several parameters are set to 0, equations (5.6) simplify as follows.

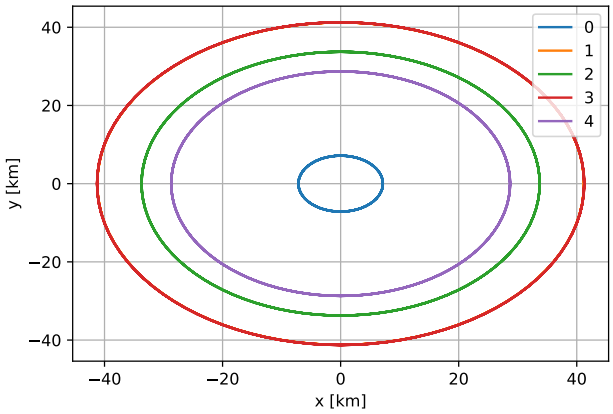
$$x(t) = -a\beta\sin(tn) \tag{5.7a}$$

$$y(t) = a\beta\cos(tn) \tag{5.7b}$$

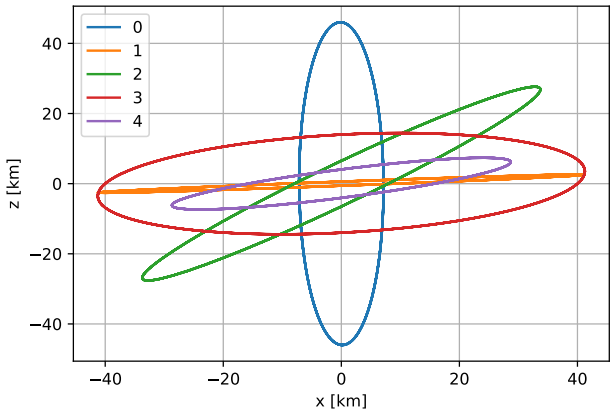
$$z(t) = a\delta\sin(tn - \psi) \tag{5.7c}$$

$$\tag{5.7d}$$

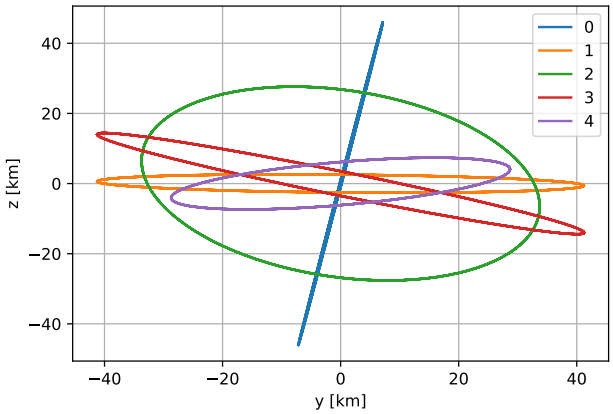
Figures 5.1 and 5.2 show the results of a realization of equations (5.6) with parameters from table 5.1 in the three 2D planes and one 3D view. As expected, the equations lead to concentric circles in xy-plane and ellipses of varying orientations, semi-major and semi-minor axes in xz and yz planes.



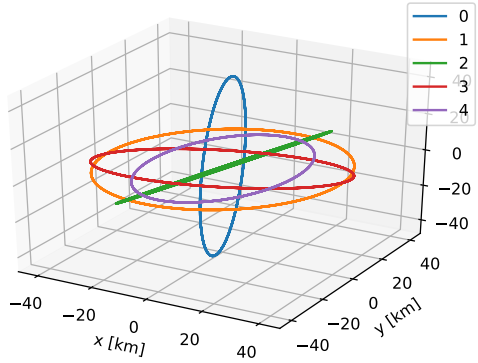
(a) Sky frame HCW xy-plane for 200km orbit height



(b) Sky frame HCW xz-plane for 200km orbit height



(c) Sky frame HCW yz-plane for 200km orbit height



(d) Sky frame HCW 3D for 200km orbit height

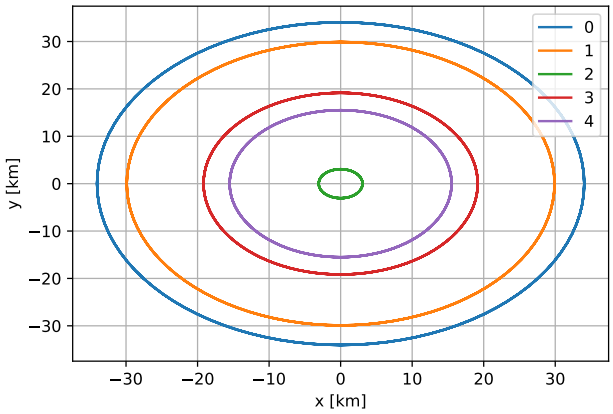
Figure 5.1: Sky frame HCW tracks for 200km orbit height for $N = 5$ nodes for 1 orbital period

5.5 Summary

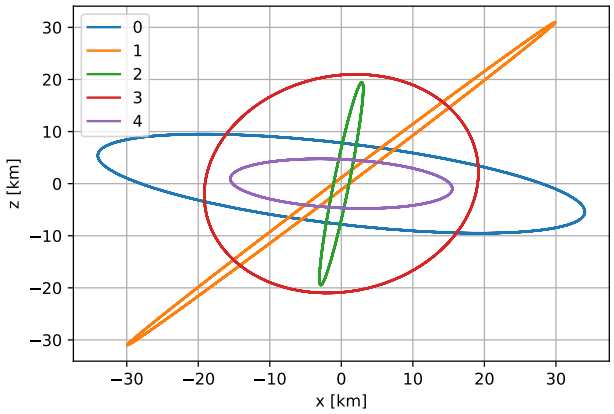
In this chapter, an overview over the Orbiting Low Frequency Array for Radio astronomy (OLFAR) mission was provided as a typical example for a distributed space-based interferometry mission deployed in GNSS-denied environment.

First, the science goals of OLFAR, namely performing low-frequency interferometry without disturbance by Earth-based interference or Earth's atmosphere, were introduced. It was found from existing literature on OLFAR, that a residual clock error may not exceed 10ns.

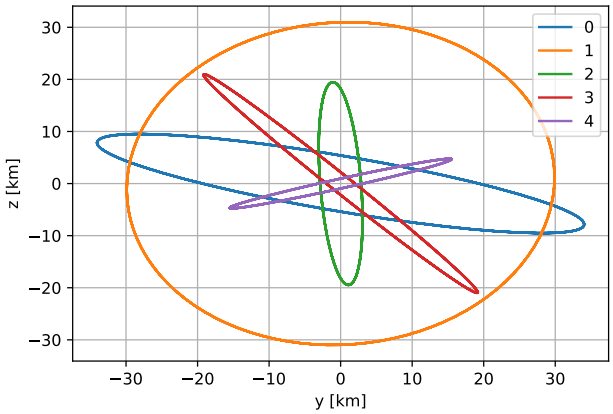
From the different potential deployment locations of OLFAR, the one in Lunar orbit is most demanding in terms of relative motion of the nodes. Thus verifying that the time synchronization algorithms achieve satisfying accuracy for this worst case also serves as verification deployment locations with less demanding relative motion. In an effort to model the relative motion of the nodes in Lunar orbit, a suitable model for the relative motion of OLFAR satellites had to be found. For that purpose the Hill-Clohessy-Wiltshire (HCW) model in the Hill frame



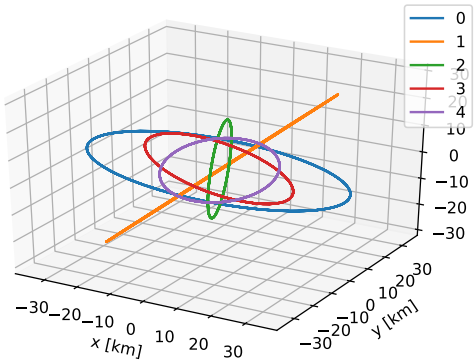
(a) Sky frame HCW xy-plane for 3000km orbit height



(b) Sky frame HCW xz-plane for 3000km orbit height



(c) Sky frame HCW yz-plane for 3000km orbit height



(d) Sky frame HCW 3D for 3000km orbit height

Figure 5.2: Sky frame HCW tracks for 3000km orbit height for $N = 5$ nodes for 1 orbital period

and a fixed sky frame were introduced. An OLFAR orbit design by [32] using HCW equations in the sky frame was implemented and will be used for the mission level time synchronization in the next chapter.

6 Network and mission time synchronization

6.1 Introduction

In this chapter, the topics of network and mission synchronization as shown in figure 1.1 will be addressed using the OLFAR example. The topic of network synchronization tackles the question of how pairwise methods can be used to synchronize a network of nodes, or as formulated in research question 2:
How can pairwise algorithms be used on a fully connected network? Which synchronization path should be chosen?
The goal of this section is to answer the research question in a general sense for anchorless satellite missions, but to also propose a time synchronization protocol for the OLFAR mission and derive recommendations and constraints for satellite and mission design for OLFAR.

Firstly, a reference node needs to be chosen among the asynchronous nodes. The clock of the reference node will then be assumed to be the true time and the other nodes will be synchronized to it. The questions of how to choose such a reference among a swarm of non-hierarchical, identical nodes is discussed in section 6.2. A procedure for randomly choosing the reference node is proposed and the statistical duration of such a procedure is examined. Furthermore, the potential integration of external time references into the synchronization procedure is briefly illustrated.

Secondly, after a network has determined its reference node, the time synchronization itself can commence. This will be addressed in section 6.3. For that, the path through the network i.e. the order in which asynchronous nodes are synchronized to the network needs to be determined. As pairwise methods require exactly $N - 1$ edges in a graph of N nodes for full network synchronization, but the unweighted graph is fully connected with $\frac{N(N-1)}{2}$ edges, the subset of edges to be used needs to be chosen. Three methods for path planning are proposed, and their advantages and disadvantages are presented. On the example of the OLFAR mission with $N = 25$ nodes, two methods are recommended for use.

Thirdly, the topic of mission synchronization and thus research question 3 is addressed in section 6.4 of this chapter:
On the example of the OLFAR mission, how can the network synchronization be employed to

achieve the mission level synchronization requirements? How frequently does resynchronization need to be performed?
Compared to the fairly general results on network synchronization, this section will be specific to the OLFAR mission. The requirement on OLFAR clocks and the motion model presented in chapter 5, will be used to show that the pairwise algorithms work with satisfactory accuracy on the OLFAR mission, and what resynchronization periods can be achieved. The results can give valuable insights to OLFAR mission and hardware designers as to how time synchronization on the mission can be implemented, what accuracy can be achieved, how much resources are required and what requirements will be imposed on the hardware.

6.2 Reference node

6.2.1 Choice of reference in a swarm of identical elements

All pairwise synchronization methods require a reference node. The other nodes in the network are then synchronized to the clock of the reference node. In hierarchical or heterogeneous networks, the choice of reference node can arise naturally, e.g. choosing the master node or the node with the best clock hardware as reference. However, for swarms of identical nodes like the OLFAR mission, there is no designated reference. This is why before any time or frequency domain pairwise synchronization method can be applied, the reference node needs to be determined. There are different options for determining the reference:

1. A reference could be **designated a priori**, by command from the ground station or even pre-programmed into the satellites before launch.

2. A reference could be determined **randomly online** without intervention from ground control.

3. A reference could be selected **figure of merit based online** without intervention from ground control, for example based on position, charge, etc.

The first option is essentially violating the requirement of not having a hierarchical system as at least temporarily the ground station would impose a hierarchy on the system. The second and third options would be fully online and thus be a more autonomous solution decreasing ground workload and communications. A figure of merit could be used to determine the best suited reference node online. Depending on the exact implementation of the pairwise synchronization in a network of nodes, it could be advantageous to use the node with e.g. the most stable temperature control, with least average distance to other nodes or the one that has most energy stored in its battery as the reference. Such approaches that select nodes based on their available energy resources can be found in [41], [42]. However, such an approach would require sensing these quantities, defining a cost function and possibly reaching consensus among the nodes and

is thus beyond the scope of this work. The second option of randomly determining the reference node is fully autonomous and can be performed without requiring further information.

In the remainder of this subsection, a protocol to select a reference node will be proposed. Let a network of N asynchronous nodes determine one reference node i between times t_0 and t_1 agreed upon before, i.e. in an interval of $T = t_1 - t_0$ as follows: The first node to transmit a message during this period will become the reference node. After receiving the message from node i, all other nodes in the network will have information about node i being the reference node. In [41], similarly, the reference node (or cluster head) is chosen randomly, although probabilities are dependent on the energy resources available at each node. Because no prior information is able in the scenario considered here, the choice of reference node is fully random.

Let each node schedule its transmission time randomly within the interval using a uniform distribution such that

$$t_n = \mathcal{U}(t_0, t_1) \quad \forall\, n. \tag{6.1}$$

A transmission is considered successful if no other node transmits during the propagation time τ of the message sent by the first node. The probability that the transmission of the a node will collide with that of another node is thus

$$p_c = \sum_{\forall n \neq i} \Pr\left(t_i \leqslant t_n \leqslant t_i + \tau\right) = (N-1)\frac{\tau}{T}. \tag{6.2}$$

For the procedure to be successful in most of the cases, a very low probability of collision is desired. Let $p_c = 10^{-4}$ be sufficient. At a maximum distance of 100km for OLFAR and speed of light as propagation speed, the propagation time is $\tau \leqslant 334\mu$s. We can then calculate the duration of the time interval for a network of $N = 25$ nodes as $T = 80$s. This interval from which nodes draw their scheduled transmission times should not be confused with the actual duration of the procedure, as after one message is transmitted, the procedure will terminate. While a larger N leads to a longer required interval T for the same collision probability, it at the same time reduces the expected fraction of the interval when the first node will transmit. The probability that at least one node will have transmitted at time $t_0 + t$ is

$$p_t(t) = 1 - \prod_{n=0}^{N} \Pr\left(t < t_n\right) = 1 - \left(1 - \frac{t}{T}\right)^N, \tag{6.3}$$

or solved for the time

$$t(p_t) = T\left(1 - (1 - p_t)^{\frac{1}{N}}\right). \tag{6.4}$$

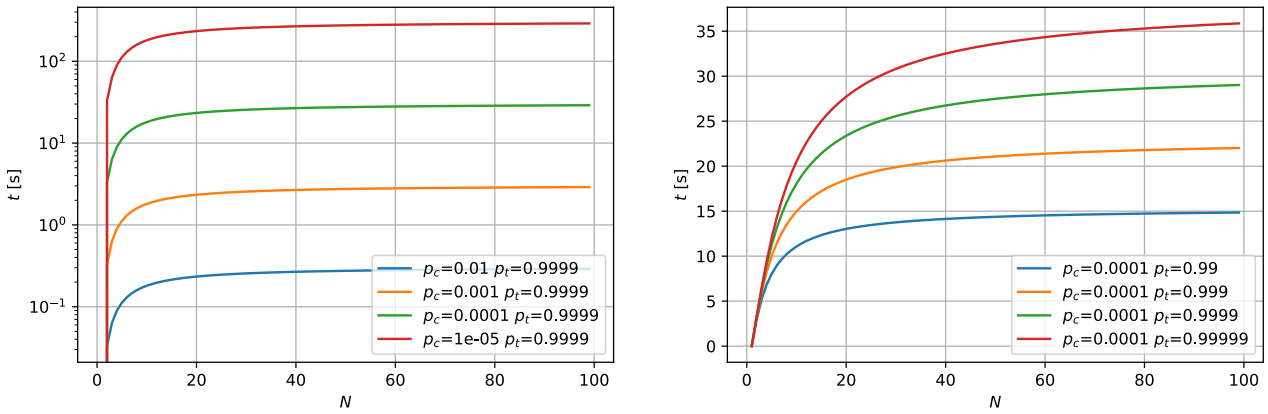
Using equations (6.4,6.2) the time t at which the transmission for choosing the reference node t will be completed can be expressed in terms of p_c, p_t, τ and the number of nodes N as

$$t(p_c, p_t, \tau, N) = (N-1)\frac{\tau}{p_c}\left(1 - (1 - p_t)^{\frac{1}{N}}\right). \tag{6.5}$$

Figure 6.1 shows how the time the procedure takes to complete with probability p_c will change for larger N, again under the assumption of certain collision probabilities p_c. It is important to notice that even for large networks the time required for finding a reference node is finite with

$$t_{max} = \lim_{N\to\infty} t(N) = \lim_{N\to\infty} (N-1)\frac{\tau}{p_c}\left(1-(1-p_t)^{\frac{1}{N}}\right) = -\frac{\tau}{p_c}\ln(1-p_t), \tag{6.6}$$

and hence the proposed procedure is applicable regardless of the number of nodes in the network.



(a) time for reference node selection with constant $p_t = 99.99\%$ and varying collision probability p_c

(b) time for reference node selection with varying p_t and collision probability $p_c = 0.01\%$

Figure 6.1: Time required for reference node selection with probability p_t, the probability of message collision p_c, $\tau = 334\mu$s as a function of number of nodes N

Table 6.1 shows a set of parameters chosen for the OLFAR mission to chose the reference node.

Table 6.1: Parameters for OLFAR random reference node choice message exchange

parameter	value	unit	description
N	25		number of nodes
τ	334	μs	max. propagation delay
p_c	10^{-4}		probability of collision of 1st message
T	80	s	Duration of interval scheduled for reference node selection i
p_t	0.9999		probability that transmission has occurred at time t
t	24.7	s	time at which the reference node is found with probability p_t

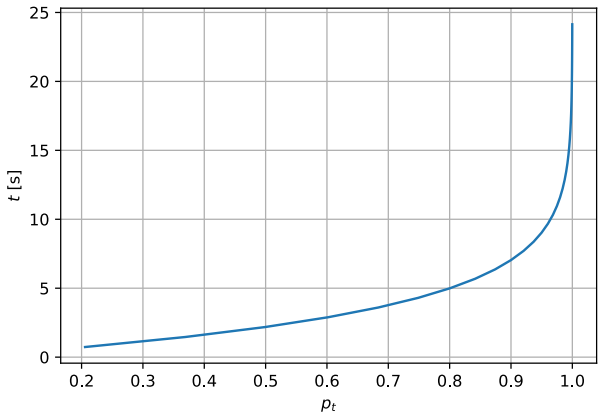


Figure 6.2: Time t required for reference node selection with probability p_t for OLFAR settings in table 6.1

Remark on graph connectivity: For the procedure proposed in this subsection it has been assumed that all N nodes form a complete graph i.e. in case of OLFAR every satellite can directly communicate with any other satellite of the swarm. This assumption is reasonable for OLFAR and other satellite swarms, because the relative position and attitude is unknown, thus all swarm elements need to be designed such that they could communicate with the farthest element of the swarm. For non complete but connected graphs, the distributed consensus problem would need to be solved in another way, for example using gossip algorithms.

Remark on the effect of initial clock asynchrony: While performing the aforementioned procedure to find a reference node, the nodes are asynchronous, meaning their scheduling will suffer from clock errors. Clock errors are assumed to be $\phi_i = \pm 5$s for the offset and $\omega_i = 1 \pm 10^{-3}$ for the skew. For $T \gg \phi_i + \omega_i t$, which is the case for OLFAR with $T = 80$, the effect of clock errors is negligible. Even when the effects of the offset and skew are in the order of magnitude of the inverval duration T, the procedure still works as it will still generate random time stamps. However, then the distribution of scheduled transmissions will not be uniform anymore but rather Irvin-Hall distributed with $n = 3$. That will adversely affect collision probabilities. Nevertheless, this can be remedied by drawing t_n from a distribution that counters the effect of ϕ_i and ω_i, such that the sum distribution of the resulting timestamps is again uniform.

6.2.2 On external time references

A potential limitation of the proposed time synchronization methods for networks of asynchronous nodes is that they synchronize among the nodes with their erroneous clocks. When synchronization is achieved in the whole network, the network time still differs from the true time with the offset and skew from of the reference node. Depending on the application, this

might be a non-issue, a nuisance that can be corrected in post-processing or a serious issue. For interferometry missions like OLFAR a residual skew in the network will lead widening or narrowing of the field of view, which can be corrected in post-processing.

However, using pairwise methods, it would be simple to extend the synchronization protocol such that it includes an external reference. After choosing the reference node i of network, this reference node could first synchronize to an external reference and then synchronize the rest of the network to its clock (which is now synchronized with the external reference). While integrating the external reference into the protocol is straightforward, finding a suitable reference, preferably in proximity to the swarm, can be the bigger challenge depending on the deployment location of the swarm. In the following paragraph, the possible external clock references for OLFAR with Lunar orbit as deployment location are briefly discussed.

The types of external resources used could include resources on Earth or in Earth's orbit, other spacecraft in lunar orbit or potentially even a ground station on the surface of the Moon. Ground stations on Earth are the classic option for timing and ranging applications. Recently, the use of satellites in Earth's orbit — namely GNSS constellations — for missions beyond orbit became a focus of research. In 2019, NASA demonstrated the successful reception of GPS signals for navigation at 187,000km distance from Earth as part of the Magnetospheric Multiscale Mission (MMS) mission, a formation of satellites in a highly elliptical orbit (HEO) tasked with studying Earth's magnetic field [43]. The European Space Agency (ESA) plans on taking this to the next step, by using GALILEO navigation signals in their Lunar Pathfinder mission, which would more than double the distance of use of GNSS signals compared to NASA's MMS mission [44]. It would create a precedent for using GNSS signals on or near a celestial body other than Earth and might be an interesting example for the OLFAR mission. Besides Earth (orbit) based techniques, other spacecraft in Lunar orbit with precise clocks, such as the projected Lunar Gateway station or even a potential base on Moon's surface could be interesting sources of closer-by clock references.

6.3 Synchronization path planning

After determining a reference node, the network needs to perform the synchronization itself. There are various options for the synchronization path in the network. In this section, the goal is to show how the choice of synchronization path can influence system level considerations. It is not, however, the goal of this section to propose an optimal solution or to give a comprehensive overview over different path planning methods.

Generally, network path planning and routing is well understood. Prevalent methods like [42] relay on Dijkstra's algorithm to find a the shortest path between to specific nodes or search for a minimum spanning tree [45], i.e. the subgraph that connects all nodes at the lowest possible sum of edge weights. However, the prevalent methods have one thing in common: They operate

on a weighted graph to minimize communication cost. In the scenario considered in this thesis, no prior information on node's distances is available. Thus, the graph is unweighted and the aforementioned methods cannot be used.

In the network of N nodes the reference node i needs to synchronize all other nodes to the network. As mentioned in the previous section, for OLFAR and similar interferometry missions the swarm elements form an unweighted, undirected and complete graph, i.e. all nodes can communicate with each other in both directions and the cost of communication or distance is initially unknown. A graph is denoted as

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \tag{6.7}$$

where $\mathcal{V}$ are the vertices and $\mathcal{E}$ the edges of the graph and

$$\mathbf{A} \in \mathbb{R}^{|V| \times |V|} \tag{6.8}$$

represents the adjacency matrix. For a complete graph with $N = |V_c|$ nodes or vertices and thus $\frac{N(N-1)}{2}$ edges, the adjacency matrix is an all-ones matrix with zeros on the main diagonal

$$\mathbf{A}_c = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{N \times N}. \tag{6.9}$$

Pairwise synchronization methods synchronize two nodes at a time. Hence, for the synchronization to propagate through the network, a path has to be chosen. In terms of graph theory, the complete graph needs to be reduced to a simple graph, i.e. the set of edges reduced from $|E_c| = \frac{N(N-1)}{2}$ to $|E_s| = N - 1$, where $E_s \subseteq E_c$. The set of edges then indicates the pairwise synchronization path to be used. In the following, three methods of executing the time synchronization in the network using different subsets $|E_s|$ will be devised. For that, m is introduced, representing the number of iterations required. Let during an interval of $m = 1$ a pair of two nodes exchange K messages and complete their pairwise synchronization process.

6.3.1 Single path

First, let a straightforward path be chosen such that each node — after being synchronized to the network — synchronizes with one other node until the whole Network is synchronized, i.e. reference node i synchronizes with node j at $m = 1$, then node j synchronizes with another node at $m = 2$ until at $m = N - 1$ all nodes are synchronized to the network. The process is visualized in figure 6.3 below. $N = 10$ nodes is chosen in the visualization for visual clarity compared to showing more nodes.

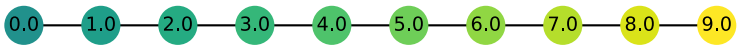


Figure 6.3: Synchronization for a single path in a network of $N = 10$ nodes. Number on the node indicates interval m.

The number of synchronized nodes N_s can then be expressed as a function of elapsed intervals m as

$$\boxed{N_s(m) = 1 + m \quad m \in \mathbb{N}_0 \qquad \Leftrightarrow \qquad m(N_s) = N_s - 1}. \tag{6.10}$$

This procedure has the advantage that it will be equally demanding to all nodes, apart from the first and last, as all other nodes are involved in two synchronizations – first their own synchronization to the network and then synchronizing one other node to the network. Another advantage is that this procedure does not require any multiple access methods, as only one synchronization is taking place at any given time. The disadvantage of this method is large number of $m = N - 1$ intervals required for full synchronization.

6.3.2 Broadcast

Second, a broadcast-based method is proposed to reduce the number of intervals required for synchronizing the network and to exploit the fact that all nodes can listen to the initial transmission of the reference node. For the other two methods presented in this section, a pairwise synchronization between two nodes would consist of consecutive messages, i.e. the synchronization between two nodes will be completed 'in one go' and not be intermitted by messages between other nodes. For such a complete synchronization of K messages a duration of $m = 1$ interval was defined. For this broadcast method, let $\frac{K}{2}$ messages be sent from node $i \rightarrow j$ during $m = \frac{1}{2}$ interval and the reverse transmission $i \leftarrow j$ of the same number of messages and duration occur at a later point in time.

Let now the initial transmission by the reference node i during the first half interval be executed as a broadcast, meaning all other $N-1$ nodes receive the transmission and record the $\frac{K}{2}$ time and/or frequency measurements. Now it remains for each node to transmit the second set of messages back to the reference node, taking $m = \frac{1}{2}$ per node. Then, the number of synchronized nodes can be expressed as a function of the number of required intervals m as

$$N_s(m) = \begin{cases} 1 & \text{for } m = 0 \\ 2m & \text{for } \geqslant 0.5 \end{cases} \quad 2m \in \mathbb{N}_0 \quad \Leftrightarrow \quad m(N_s) = \begin{cases} 0 & \text{for } N_s = 1 \\ 0.5N_s & \text{for } N_s \geqslant 2 \end{cases}. \tag{6.11}$$

After the second node, two nodes per interval can be synchronized rather than only one node when choosing a single path in the network. The process of using a broadcast from the initial node and then individual sequential transmissions by the other nodes back to the reference node is visualized in figure 6.4

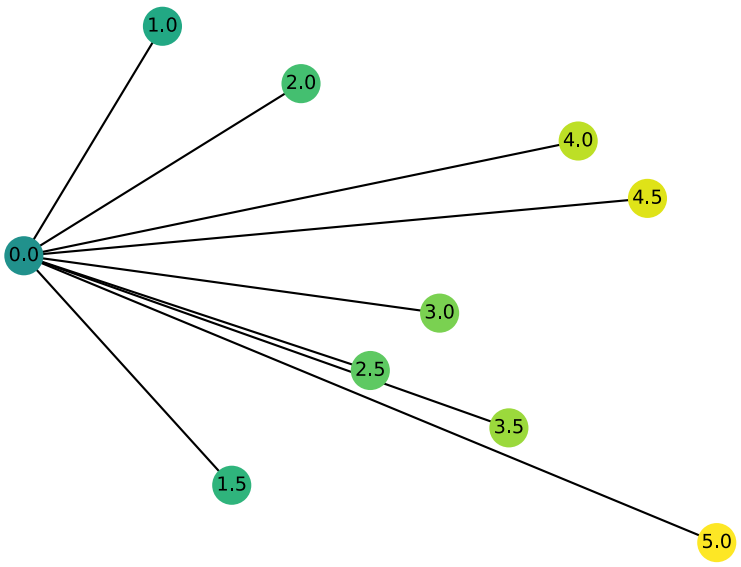


Figure 6.4: Synchronization for a broadcast message distribution in a network of $N = 10$ nodes. Number on the node indicates interval m.

Remark on message order and measurement vector composition: The legacy time domain methods and novel frequency and multi domain methods presented in chapters 2 and 3 respectively are all invariant to the order of transmission, i.e. any message order is acceptable. Furthermore, the number messages sent in the two directions can be of arbitrary composition provided there is at least one transmission in each direction. For using the broadcast method, additional limitations are imposed, namely, that exactly $\frac{K}{2}$ messages are sent in each direction,

and, that the first half of messages in one direction is sent in one block and the second half in the opposite direction in a second block.

Remark on triggering transmissions from the asynchronous nodes: Between $m = 0.5$ and 1 the first asynchronous node will transmit its messages to the reference node i. Upon completion, the second asynchronous node has to begin its transmission to the reference node, however as the node is still asynchronous this event cannot be scheduled. Instead, this node needs to engage in passive listening to know when the prior node has completed the synchronization to start transmitting its own messages. The same holds true for the subsequent nodes.

6.3.3 Tree propagation

Third, in pursuit of a solution requiring even less intervals for full network synchronization, another procedure will be proposed. Let the reference node i synchronize a other nodes. In the next step, let each of the $a + 1$ synchronized nodes synchronize a further nodes, that means after the second step a total of $a + 1 + a(a + 1) = (a + 1)^2$ nodes will be synchronized. Each of these nodes then again synchronizes a further nodes such that after the third step $(a + 1)^2 + a(a + 1)^2 = (a + 1)^3$ are synchronized and so on.

We can thus write for the number of synchronized nodes N_s as a function of elapsed intervals m and number of new nodes that each node synchronizes a as

$$N_s(m, a) = (1 + a)^m. \tag{6.12}$$

It was defined that, during an interval m, a node can perform bi-directional message exchange to synchronize with one other node, which sets $a = 1$, leading to

$$\boxed{N_s(m) = 2^m \quad m \in \mathbb{N}_0 \qquad \Leftrightarrow \qquad m(N_s) = \lceil \log_2(N_s) \rceil}, \tag{6.13}$$

intervals required to synchronize a network of N nodes using this procedure. The advantage of this method over the single path is the lower number of intervals needed to synchronize a network of a given size. Instead of increasing linearly with intervals m, the number of synchronized nodes increases exponentially. The synchronization method is visualized in figure 6.5 for $N = 10$ nodes.

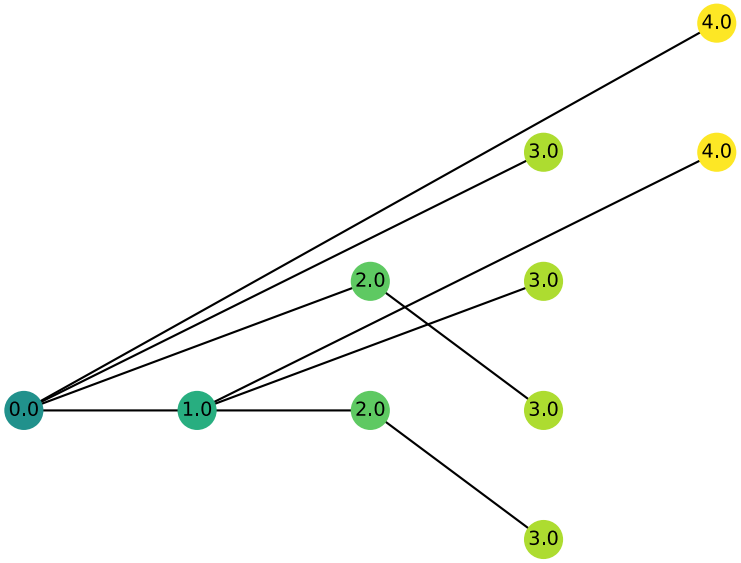


Figure 6.5: Synchronization for a tree path in a network of $N = 10$ nodes with $a = 1$. Number on the node indicates interval m.

For other methods — single path and broadcast — only one transmission at a time happens among all nodes in the network. For this method after interval $m \geqslant 2$, more than one message will be sent in the network. As nodes communicate wirelessly and can generally communicate with all other nodes, messages sent simultaneously without further provisions would collide in the wireless channel. To avoid this, a multiple access method that provides multiple channels at any given time needs to be employed, such as frequency division multiple access (FDMA) that divides the available spectrum into multiple channels each of smaller bandwidth, code division multiple access (CDMA) that uses orthogonal codes to spread the message on the spectrum or space division multiple access (SDMA) that uses spatially separated messages e.g. through directed antennas. Generally, multiple access methods increase implementation complexity and divide the resources available among multiple channels and consequently reduce the bandwidth available (FDMA and CDMA). While it is not the goal of this thesis to recommend a particular multiple access method, the number of channels required for this tree propagation needs to be computed for comparison with the other methods.

The number of channels needed in each interval is dependent on the number of additional nodes that are synchronized in that interval. The highest number of nodes synchronized in one interval over all intervals determines the number of channels required for the whole procedure. As the total number of synchronized nodes N_s grows exponentially in m, the last intervals are the ones with most new nodes synchronized. Now, two cases have to be distinguished:

1. **Most channels N_c required on the last interval if $N_s \geqslant \frac{3}{4}2^m$:** On the example

illustrated in figure 6.5, $m = 4$ intervals are shown for $N = 10$ nodes. It can be seen that for $N = 10$ nodes, this condition is not fulfilled. The last interval is not populated with as many nodes as it could be (8 in this case). However, it can easily be seen that for $N \geqslant \frac{3}{4} \times 16 = 12$, 4 or more nodes need to be synchronized in the last step, and the condition would be fulfilled. Then, the number of channels in the last interval can be expressed as $N - 2^{m-1}$.

2. **Most channels N_c required on the second-last interval if $N_s < \frac{3}{4}2^m$:** On the example from figure 6.5, with $N = 10$ nodes, the maximum number of nodes is reached at the second-last interval. The number of nodes in the second-last interval is always fully populated, thus for the number of channels required in the second last interval 2^{m-2} holds.

As the decisive quantity is the maximal number of channels, one can take the maximum of the number of channels required at the last and second-last interval and furthermore express m in terms of the number of nodes as per equation (6.13) leading to the following equation for the number of channels

$$N_c(N_s) = \max\left(N_s - 2^{(\lceil \log_2(N_s) \rceil - 1)}, \quad 2^{(\lceil \log_2(N_s) \rceil - 2)}\right). \tag{6.14}$$

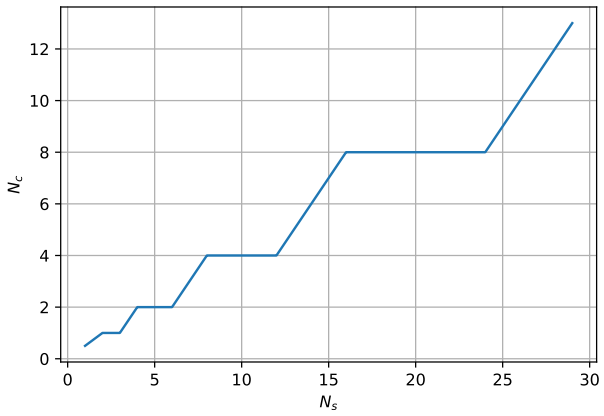


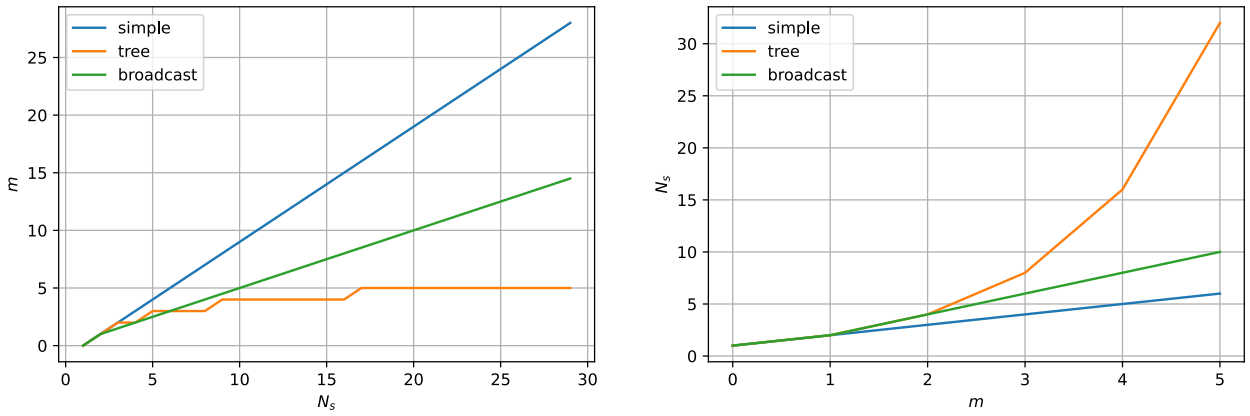
Figure 6.6: Maximum number of channels N_c required as a function of Number of nodes to be synchronized N_s using tree propagation

Figure 6.6 shows a plot of this equation. Going back to the example from figure 6.5, it becomes apparent that $N = 10$ lies exactly in a plateau, while for $N > 12$ the required channels increase again.

6.3.4 Comparison

In this subsection, the three methods for synchronization path planning in an anchorless network of asynchronous nodes are compared. All three methods operated on a fully connected network

of N nodes and use a subset of $N-1$ connections for the synchronization path. In figure 6.7, the relation between network size and intervals required for synchronization is illustrated. The number of intervals is proportional to the time it takes to synchronize a network. Therefore, it is an important figure of merit to compare the different possible synchronization paths. For the single path and broadcast methods, N_s follows a linear trend in m with slope 1 and 0.5 respectively, whereas the tree method is governed by a stepped logarithmic function. In terms of intervals required, the broadcast method is superior to the single path method for all network sizes ($N > 2$). For more than 6 nodes in the network, the tree method outperforms the broadcast method. Especially for larger networks the tree algorithm offers a significant reduction in number of required intervals and thus duration of the synchronization process. Hence, it should be recommended more networks with a large number of nodes.



(a) Number of intervals m required to synchronize N_s nodes

(b) Number of nodes N_s that can be synchronized in m intervals

Figure 6.7: Relation of Number of synchronized nodes N_s and number of intervals m for the three synchronization methods

To compare the performance of the three different synchronization path methods for OLFAR, figure 6.2 aggregates the performance parameters for a network of 25 nodes. The number of intervals m is highest for the simple path. Using the broadcast method almost halves the required intervals whereas the tree method requires a little more than a fifth and is thus by far the fastest method. As mentioned in the prior subsection, due to the fact that multiple nodes are synchronized to the network simultaneously, the tree method requires multiple channels — 9 for this network size. The requirement for multiple channels could limit the bandwidth and therefore negatively affect the pairwise synchronization performance. Besides, the table shows the number of required messages N_t to be transmitted by the network as a whole and can serve as a measure of energy required. Single path and tree method require $(N-1)K$ total transmissions, whereas the broadcast method reduces the number of transmissions requiring only $0.5NK$ transmissions as the reference node requires $0.5K$ messages for the initial broadcast and the other nodes then require another $0.5K(N-1)$ for their individual reverse transmis-

sions. Nearly halving the number of transmissions, the broadcast method is the method that minimizes the total required energy for transmission over the network.

Table 6.2: Parameters for OLFAR path planning with $N = 25$

parameter	description	single path	broadcast	tree
m	intervals required	24	12.5	5
N_c	number of channels required	1	1	9
N_t	total number of transmissions	$24K$	$12.5K$	$24K$
$N_{t,max}$	max. number of transmissions by a single node	$\frac{K}{2}$	$\frac{K}{2}$	$5\frac{K}{2}$

In conclusion, for networks with $N > 6$, the tree method offers the fastest synchronization time. For the OLFAR example of $N = 25$ it requires only $m = 5$ intervals compared to 2.5 and almost 5 times that number for the broadcast and single path methods respectively. However, this comes at the cost of needing 9 channels. The tree method is suggested for OLFAR, should the mission designers want to optimize time required for network synchronization. In case communication energy consumption is chosen as a metric, the OLFAR mission should rely on the broadcast method for synchronization path planning as it requires about half the amount of total energy from the network compared to both other methods.

Remark on incorporation of prior information on node distance: A fully connected unweighted graph was considered to represent the network, meaning firstly that all nodes can communicate, which is accurate for OLFAR and similar satellite swarms as argued in section 6.2. Secondly, the unweighted graph represents an equal cost of communication for all edges, which is obviously not true because distances between nodes vary. Nonetheless, as an anchorless network is considered, the relative positions are unknown when the synchronization process is executed. This is why the representation as an unweighted graph is accurate for anchorless networks as are considered this thesis. Should one however want to incorporate prior information on e.g. node distances, the network could be represented as weighted graph, with weights indicating communication cost. In such a case, an (energy)-optimal synchronization path would be the minimum spanning tree, and thus existing algorithms such as Prim's or Kruskal's algorithms could be used for path planning.

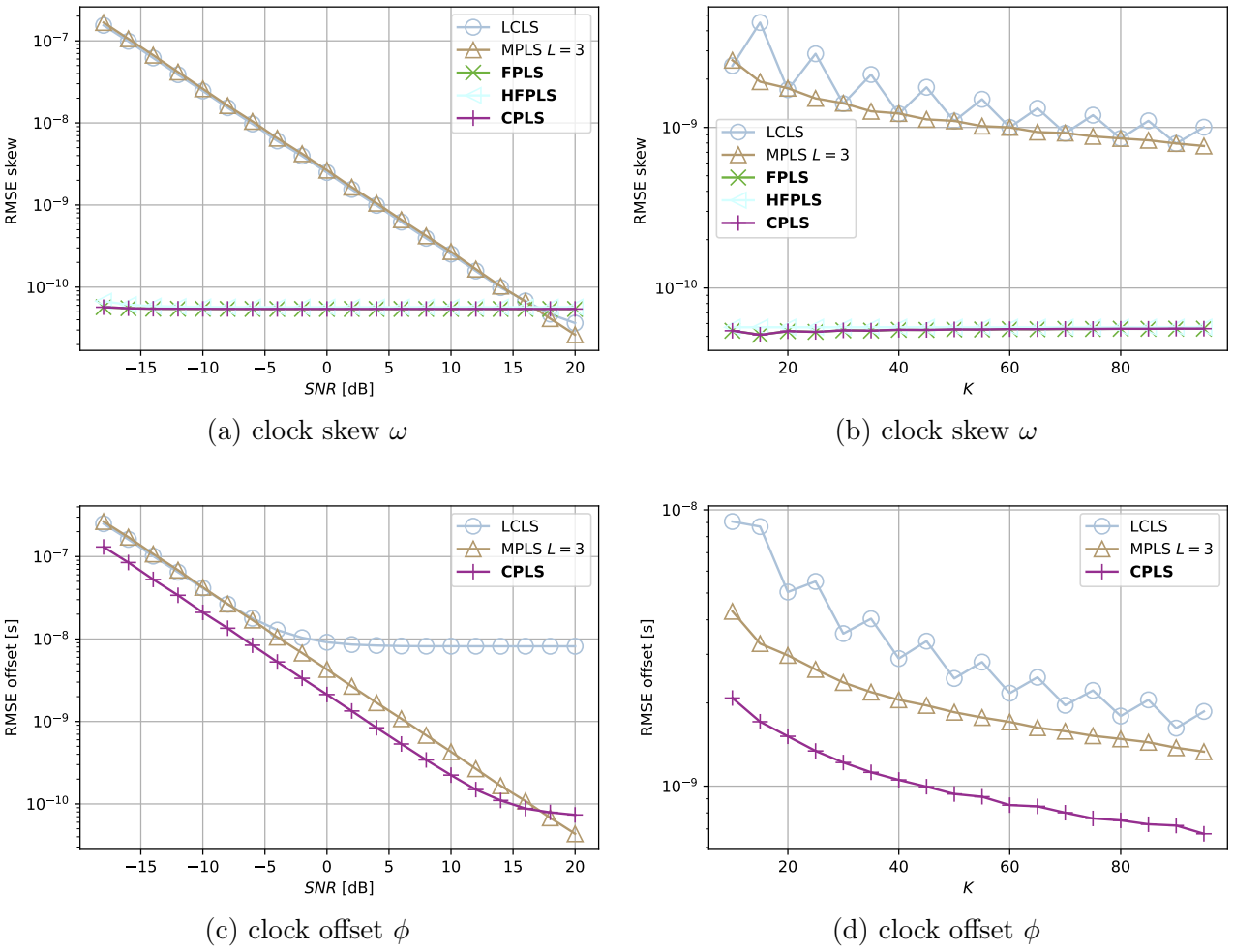
6.4 Mission synchronization

In this section, synchronization over the whole mission duration will be examined. The goal is firstly to show how the different algorithms perform on the OLFAR mission, and secondly to determine the resynchronization period.

For the simulation of the orbital motion of the OLFAR nodes, the Sky-HCW-model presented in section 5.4 is used. Out of the two orbital designs presented by Dekens et al. the one with the lower orbital height of 200km is selected, as the lower orbit leads to higher relative

velocities between the nodes compared to the a higher orbit. The OLFAR motion model chosen here is hence a worst case representation of dynamic models for the OLFAR mission.

Figure 6.8 shows the results of clock parameter estimation for various time and frequency domain algorithms. Generally the simulation setup is similar to the one presented in chapter 4, with the difference that the linear motion model is replaced by the OLFAR orbital dynamics. The results are in many ways similar to the ones obtained in chapter 4, therefore no further in-depth discussion the results of each individual algorithm shall be conducted here.



(a) clock skew ω

(b) clock skew ω

(c) clock offset ϕ

(d) clock offset ϕ

Figure 6.8: Clock parameter estimation error using pairwise algorithms time domain and frequency domain; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for $SNR = 0$dB for OLFAR at orbit height of 200km

The goal of this simulation was to show, that when considering relative orbital motion of OLFAR, the pairwise methods deliver satisfactory results. This was successful, as depending on the choice of algorithm and SNR scenario, RMSEs below 10^{-10} can be achieved both for offset and skew estimation.

In the next step, the resynchronization period for the OLFAR mission is considered. Fol-

lowing equation (2.3), let the total clock error of the i-th node be defined as

$$\begin{aligned}\delta t_i(t) &= t - t_i \\ \delta t_i(t) &= t - (1 + \delta\omega_i)t + \delta\phi_i \\ \delta t_i(t) &= \delta\omega_i t + \delta\phi_i,\end{aligned} \tag{6.15}$$

where t is the true time elapsed since the beginning of the last synchronization and $\delta\omega_i$ and $\delta\phi_i$ are the clock skew and clock offset estimation errors respectively. For OLFAR, the maximum acceptable local clock error as discussed in section 5.3 is

$$|\delta t_i(t)| = 10\text{ns}.$$

Now, resynchronization period t_r can be found by solving the previous equation for the true time t.

$$t_r = \frac{\delta t_i - \langle|\delta\phi_i|\rangle}{\langle|\delta\omega_i|\rangle}, \tag{6.16}$$

where, the maximum acceptable local clock error is specified and the expected absolute values for the clock errors are known from simulations. Let a scenario with $K = 10$ messages be considered. The required resynchronization period can then be shown as a function of the SNR. The results are shown in figure 6.9.

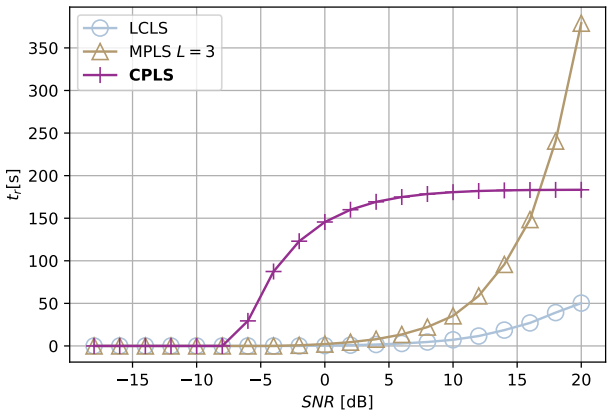


Figure 6.9: OLFAR resynchronization period t_r for $K = 10$ and varying SNR at $\delta t_i(t) = 10$ns for OLFAR at orbit height of 200km.

Consider a SNR of 10dB. Using CPLS a resyncronization period of 180s could be achieved, whereas the prevalent method MPLS with $L = 3$ only allows for a 40s resynchronization period. Moreover, the figure shows that the novel CPLS outperforms prevalent methods particularly in low to medium SNR conditions, whereas in higher SNR conditions MPLS with $L = 3$ can outperform CPLS. The fact that CPLS is going into saturation can be attributed to the residual bias in the skew estimate discussed in chapter 4.

For a network size on $N = 25$ nodes, $K = 10$ and the procedure for network synchronization presented in sections 6.2 and 6.3, one whole synchronization will take in the order of few seconds depending on the synchronization path planning method chosen and the required processing and guard times between messages. At a resynchronization period of 180s with CPLS, only a small fraction of the mission time will be required for synchronization purposes, and the OLFAR system will be available for scientific work during most of the mission time. It was furthermore important to show that the resynchronization period is much larger than the snapshot integration time of 1s, which was herewith achieved. Besides, in chapter 2 it was determined that the coherence time of the clock hardware must be larger than the synchronization period. In [3, ch. 3], suitable clock hardware exceeding 1000s coherence time for OLFAR requirements as per equation (2.2) was presented. Thus, the resynchronization periods of 180 seconds are well in line with available clock hardware.

6.5 Summary

In this chapter, topics of network and mission synchronization as depicted in figure 1.1 were addressed and research questions 2 and 3 were answered. A time synchronization protocol for anchorless fully connected satellite networks was proposed and tailored to the Orbiting Low Frequency Array for Radio astronomy (OLFAR) mission assuming a network of $N = 25$ nodes.

Pairwise time synchronization algorithms require a reference node. In a swarm of identical and non-hierarchical elements, the reference node needs to be chosen randomly among all nodes before the synchronization itself can commence. For that purpose, the following procedure was proposed in section 6.2: Let all nodes draw their transmission times at random from a uniform distribution. The first node to transmit successfully to all other nodes becomes the reference node. It was shown that this procedure can — with message collision probability p_c in a network of N nodes — with probability p_t in time t determine the reference node. Furthermore it was shown that t is upper-bounded in N, i.e. the procedure also holds for large networks. For OLFAR with $N = 25$ nodes, a maximum propagation delay $\tau = 334\mu$s and collision probability $p_c = 10^{-4}$, it will take an expected time to complete of less than 2.5s. With probability of $p_t = 99.99\%$ it will complete within 24.7s. In subsection 6.2.2, the possibility to include external clock references into the proposed framework is briefly illustrated.

After designating a reference node, the actual synchronization triggered by and starting from this reference node can be performed. In a fully connected graph, a subgraph has to be chosen for the network to operate on. Three options for choosing this subgraph and thereby the synchronization path through the network were presented in section 6.3. First, a 'single-path' method is proposed where the reference node performs pairwise synchronization with one other node, that in turn synchronizes another asynchronous node. This procedure of the latest synchronous node synchronizing one other node continues until clock parameter estimation in

the whole network is completed. This method requires $m = N - 1$ intervals to synchronize a network of N nodes. Second, a 'Broadcast' method was proposed, where K messages required for a pairwise synchronization are split such that the reference transmits $\frac{K}{2}$ messages to all other nodes simultaneously who then — one after another — transmit their remaining other $\frac{K}{2}$ messages in the other direction to the reference node. Consequently, after the initial broadcast by the reference node, each individual node only requires half an interval to complete its synchronization and thus $m = 0.5N \quad \text{for} N \geqslant 2$, half the time required by the 'single path' method. Third, a 'Tree propagation' method was proposed, where during every interval m each node already synchronized to the network serves one asynchronous node, resulting in an exponentially growing number of nodes synchronized to the network. The number of time intervals required is therefore $m = \lceil \log_2(N) \rceil$. Generally, the Broadcast method is recommended for minimizing total communication cost in the network while the Tree propagation method is recommended for reducing the time that the synchronization procedure takes. For OLFAR with $N = 25$ nodes, a duration of 5 intervals for using Tree propagation can be achieved compared to 12.5 intervals for Broadcast. Broadcast propagation in turn nearly halves the number of messages to be sent with $12.5K$ messages compared to $24K$ messages for the Tree method. Ultimately, it will be up to the mission designer to prioritize time or communication cost and choose the network synchronization method accordingly.

During a space mission, or generally during the operation of a distributed system of asynchronous nodes, network synchronization needs to be performed repeatedly to always ensure that the required level of synchronization is achieved. Using the dynamic model for OLFAR presented in chapter 5, the different pairwise synchronization algorithms were tested and it was verified that the algorithms perform satisfactory. Moreover, the resynchronization period was determined for different pairwise algorithms as a function of SNR and number of messages K. For an SNR of 10dB and $K = 10$, with the novel CPLS, a resyncronization period of 180s could be achieved, whereas with prevalent method MPLS with $L = 3$ only allows for 40s resynchronization period. It was found that the novel CPLS outperforms prevalent methods particularly in low to medium SNR conditions, whereas in higher SNR conditions MPLS with $L = 3$ can outperform CPLS.

The resulting time synchronization protocol covers all mission aspects from pairwise synchronization methods on the microscopic level, network synchronization on a medium level and mission synchronization on a macroscopic level.

7 | Conclusions and Future work

7.1 Conclusion

In this chapter, concluding remarks on the work carried in this thesis, the limitations of the results and recommendations for future work are presented.

To provide an overview over the results, the summary of the literature research and the answers to the research questions are presented here.

As shown in chapter 2, prevalent pairwise synchronization algorithms operate in time domain, use the affine clock model to represent clock errors and use different dynamic models. Prevalent synchronization methods for anchorless networks of mobile asynchronous nodes assume physical layer time stamping and assume the stochastic component of the delay to be zero mean Gaussian. One can distinguish between the Low Complexity Least Squares (LCLS) [1], [15] using a constant delay model and Mobile Pairwise Least Squares (MPLS) [3, ch. 4] using an L-th order dymamic model to represent a mobile scenario. The two latter methods are applicable to anchorless satellite networks, but LCLS's performance suffers from the insufficient dynamic model as satellites are in relative motion. The plethora of other prevalent synchronization algorithms were not applicable in the scope of this thesis either for not operating on mobile networks [16]–[18] or for their global formulation [2], [19].

R1 Frequency information: Can frequency information be used to improve pairwise time synchronization performance? If yes, how?

Frequency information is present in the message transmitted in prevalent time synchronization methods for anchorless mobile networks, but has not been exploited for this application. As presented in chapter 3, the clock skews of the nodes and the pairwise relative velocity are related to the frequencies measured at the two nodes. Based on this frequency-TWR framework, three novel frequency domain least squares estimators were proposed. First, the uni-directional frequency pairwise least-squares method that under the assumption of constant pairwise velocity estimates clock skew and radial velocity using at least two messages, all with the same direction of transmission. Second, the bi-directional Frequency Pairwise Least Squares (FPLS) was proposed. It estimates clock skew and radial velocity using at least three messages, of which at least one must

be sent in the opposite direction of transmission. Thirdly, with the Higher Order Frequency Pairwise Least Squares an extension of FPLS to a higher order motion model of constant acceleration L is proposed. It caters to non-linear pairwise motion which can be encountered even when the motion of individual nodes is governed by a linear model. HFPLS can estimate clock skew, velocity and acceleration with $K \geqslant 5$ messages. The proposed frequency domain methods are inherently unable to estimate clock offset and range. For effective use in clock synchronization, a novel two-stepped method — CPLS — is proposed. In a first step it uses FPLS to estimate clock skew and radial velocity to then use these known quantities in a time domain mobile estimator for offset and range estimation. The novel frequency and multi domain synchronization estimators constitute a key contribution of this work. It was shown that for constant pairwise velocity the proposed CPLS multi domain method reduces the number of required messages to $K \geqslant 3$ compared to the prevalent MPLS time domain method with $K \geqslant 4$ and significantly reduces computational requirements.

The novel frequency and multi domain methods proposed require the receiver to additionally estimate frequency of the arriving signal for which two receiver architectures — analogue and digital — where shown in chapter 4. A simulation architecture was devised and implemented to verify the novel algorithms and compare their performance to prevalent time domain algorithms. Simulations showed a better performance of the proposed frequency domain methods in low to medium SNR range, for HFPLS even over almost the whole SNR range. Nevertheless, it is too early yet to claim outperforming time domain algorithms, as this depends on the SNR definition. The relation of noise magnitudes on time and frequency measurements which ultimately governs the algorithms performance is determined by the specific communication scenario. Finally, the proposed multi domain CPLS can estimate clock skew and offset at 25% less communication cost, lower computation cost and likely higher accuracy than the best prevalent time domain algorithm — MPLS.

R2 Network synchronization: How can pairwise algorithms be used on a fully connected network? Which synchronization path should be chosen?

In a network of identical non-hierarchical elements, the reference node needs to be chosen randomly among all nodes before the synchronization itself can commence. For that purpose, a robust procedure suitable even for large network sizes was proposed in section 6.2 For OLFAR with $N = 25$ nodes, a maximum propagation delay $\tau = 334\mu$s and collision probability $p_c = 10^{-4}$, the procedure will take an expected time of less than 2.5s. With probability $p_t = 99.99\%$ it will complete within 24.7s.

For the synchronization path choice, three options were evaluated in section 6.3. Generally, the 'Broadcast' method is recommended for minimizing total communication cost in

the network while the 'Tree propagation' method is recommended for reducing the time that the synchronization procedure takes. For OLFAR with $N = 25$ nodes, a duration of 5 intervals for using Tree propagation can be achieved compared to 12.5 intervals for Broadcast. Broadcast propagation in turn nearly halves the number of messages to be sent with $12.5K$ messages compared to $24K$ messages for the Tree method.

R3 Mission synchronization: In the OLFAR mission, how can the network synchronization be employed to achieve the mission level synchronization requirements? How frequently does resynchronization need to be performed?

During a space mission, or generally during the operation of a distributed system of asynchronous nodes, network synchronization needs to be performed repeatedly to ensure that the required level of synchronization is always maintained. Hill-Clohessy-Wiltshire (HCW) equations in the sky frame were introduced in chapter 5 to be applied in an OLFAR orbit design by [32]. Using this dynamic model the different pairwise synchronization algorithms were tested and it was verified that the algorithms perform satisfactory specifically for the OLFAR mission. Furthermore, the resynchronization period was determined for different pairwise algorithms as a function of SNR and number of messages K. For an SNR of 10dB and $K = 10$, with the novel CPLS, a resyncronization period of 180s could be achieved, whereas with prevalent method MPLS with $L = 3$ only allows for 40s resynchronization period. It was found that the novel CPLS outperforms prevalent methods particularly in low to medium SNR conditions, whereas in higher SNR conditions MPLS with $L = 3$ can outperform CPLS.

The resulting time synchronization protocol covers all mission aspects from pairwise synchronization methods on the microscopic level, network synchronization on a medium level and mission synchronization on a macroscopic level.

7.2 Contributions and implications

This thesis provided several contributions in the field of time synchronization for anchorless networks of mobile asynchronous nodes, particularly for the application of satellite networks. First, a class of novel frequency and multi domain algorithms for pairwise timing and ranging was proposed and tested through simulation. The proposed methods improve communication cost, computational effort and likely also estimation performance over prevalent time domain algorithms. They are widely applicable to stationary and mobile wireless sensor network (WSN)s in various network configurations. The decrease in communication and computational requirements and increase in estimation performance can lower cost, mass and energy consumption for WSNs and potentially enable new applications with demanding requirements on clock parameter estimation.

Second, the synchronization of an asynchronous network of nodes — represented by a fully connected unweighted graph — using pairwise algorithms was examined. Three possible network synchronization algorithms were presented of which two were recommended for use in WSNs, namely the 'Broadcast' method for reducing communication cost and 'Tree propagation' method for reducing time required for synchronization. Furthermore, the question of resynchronization periods over the whole operational duration or mission of a WSN was examined, resulting in a synchronization protocol for anchorless networks of mobile asynchronous nodes that covers aspects from microscopic level (pairwise algorithms), to medium level (network synchronization) all the way to macroscopic level (mission synchronization). This protocol offers a tool for designers of mobile WSNs in GNSS-denied environments on how to achieve synchronization among nodes.

Third, all prior contributions were applied to OLFAR. It was found that the novel CPLS and prevalent MPLS can achieve satisfactory synchronization performance in the most challenging dynamic conditions possible for the OLFAR mission. A relative orbital motion model was used rather than linear motion like in prior work on OLFAR clock synchronization. It was found that for OLFAR with network size $N = 25$ the whole synchronization procedure can be on average performed within several seconds, and using the novel CPLS resynchronization periods of 180s for an SNR of 10dB and $K = 10$ could be achieved. Thus, the synchronization duration is only a fraction of the resynchronization period leaving most of the valuable mission time for science operations. This further validates the OLFAR mission concept in the aspect of clock synchronization and provides input to mission and node hardware design. A design space for number of nodes and their associated time to synchronize has been derived. The work can serve as a basis for System level trade-offs between number of nodes, synchronization time, node capability and resynchronization frequency in the context of the OLFAR mission.

7.3 Limitations

A central limitation when comparing the performance of novel frequency domain algorithms to prevalent time domain algorithm is the SNR definition on the time and frequency markers. When comparing algorithms only in either one of the domains, this is of little concerns as algorithms would then be faced with the same noisy measurements allowing for a fair comparison. When comparing performance over multiple domains, an SNR definitions that correctly relates noise on time and frequency markers is required. The position and velocity-based SNR definition used in this work only covers one aspect influencing the SNR. When position and velocity magnitudes are mismatched, that can lead to wrong results for algorithm performance comparison. The accuracy of time and frequency of arrival estimation depends on receiver architecture, clock hardware, ADC hardware, the estimator, frequency, bandwidth, signal duration, noise on the wireless channel and waveform of the signal used for transmission. Thus, for a more robust

performance comparison of different algorithms the respective communication system should be simulated in detail. Then, the most suitable pairwise algorithm for the specific scenario can be selected.

For the generic simulation of frequency domain methods a residual bias in the estimation was discovered, which for the FPLS can be explained by a higher order non-linearities, but for HFPLS more investigation should be performed into the reason for the residual bias.

The proposed synchronization algorithm work under the assumption of Gaussian noise — as do most legacy algorithms. For other assumptions on the statistical component of the frequency, new estimators would need to be derived.

The proposed synchronization protocol, both on pairwise and network synchronization level, does not integrate prior information e.g. on node positions or distances. While this was intentional to develop the synchronization protocol for on a truly anchorless network, in many applications some prior information on the dynamics of the network is available. This information could be used on pairwise synchronization level for not needing to estimate pairwise distances and velocities or on network level to compute more expedient synchronization paths through the network. The proposed methods are not able to make use of such information — neither on pairwise nor on network synchronization level.

7.4 Future research

Given the results from this thesis, several potential areas for future research can be identified.

7.4.1 Frequency augmented time synchronization

Ideas for future work with respect to frequency domain techniques are presented here.

1. **Derive ML estimators** for frequency domain under non-Gaussian noise assumptions.

2. **Correlated noise on measurements** might be experienced in broadcast scenarios or due to spatial or temporal correlation of measurement of time and frequency. Extending the statistical model to account for such correlation in the noise might improve performance in a real world scenario.

3. **Global formulation for frequency domain** both of HFPLS and FPLS can be formulated, similar to MGLS in time domain. As globalized formulations have the potential of achieving lower estimation errors, they might be beneficial for some applications.

4. **For the Combined formulation for higher order models** use HFPLS instead of FPLS to formulate a higher order combined method to further improve performance under non-linear relative motion.

5. **Adaptive combined formulation** investigate an adaptive formulation combining time and frequency domain methods. It could look like this: Make an initial estimate using one of the methods, then use the first estimate as input for the other method and keep alternately updating between the methods while increasing the model order L.

6. **Simulation of the whole communication link** to investigate the effects of frequency, bandwidth, transmitter and receiver architecture on the performance comparison of time and frequency methods and arrive at a more confident performance comparison of time and frequency domain methods.

7.4.2 Network and mission level synchronization for OLFAR and other anchorless satellite networks

Future work could be carried out on network and mission level synchronization for OLFAR and other anchorless satellite networks with the following ideas.

1. **Use a more accurate motion model** including anomalies of the gravitational field and solar radiation pressure for the simulation of the performance of time synchronization algorithms. The more detailed the model, the higher the confidence with which the performance of the time synchronization can be predicted.

2. **More sophisticated optimal path planning for network synchronization** In a fully connected graph, there are many paths to connect all nodes. Incorporating prior information on distances one could further optimize the nodes pairs chosen for pairwise synchronization, e.g. by using the minimum spanning tree protocol for network path planning. For OLFAR, such prior information could be obtained from orbital mechanics.

3. **Assess effect of residual skew in interferometry missions.** In pairwise synchronization techniques, the clocks of all other nodes are synchronized to one reference node. The node has an erroneous clock itself, thus the offset and skew propagate into the network. While the offset does not pose a problem, the residual skew might lead to distorted imaging (beam widening or narrowing). It should be determined to what degree this can be identified and mitigated in post processing, or, how external time references can be used to decrease the absolute skew of the network.

4. **Examine external clock references** for the OLFAR mission, especially the potential use of GNSS signals in lunar orbit could be promising [44].

7.4.3 Other applications

Identify applications other than space missions that could significantly benefit from the novel frequency and multi domain algorithms developed and verified in chapters 3 and 4.

1. **Apply CPLS to non-space application** such as indoor and underwater localization and perform full system simulations as was done for OLFAR in this work, to verify the applicability for this specific application.

2. **Identify applications where only skew (and velocity) estimation are needed**, in such application FPLS or HFPLS could be used standalone.

3. **Identify broadcast applications where only skew and velocity estimation are needed** and for which uni-directional frequency pairwise least squares can be used.

Acronyms

ADC analog to digital converter. 11, 36, 73

ASTRON Netherlands Institute for Radio astronomy. 3, 44

CDMA code division multiple access. 62

CPLS Combined Pairwise Least Squares. 16, 31, 32, 39–41, 43, 67–69, 71–73, 76, 96

ESA European Space Agency. 57

FDMA frequency division multiple access. 62

FDOA frequency difference of arrival. 17

FFT fast fourier transform. 35

FMCW frequency modulated continuous wave. 17

FPLS Frequency Pairwise Least Squares. iii, 16, 22–24, 27–32, 40–43, 70, 71, 74, 76, 96, 97

GNSS Global Navigation Satellite System. 6, 50, 57, 75

GPS Global Positioning System. 57

HCW Hill-Clohessy-Wiltshire. iii, 38, 44–47, 49–51, 65, 72

HEO highly elliptical orbit. 57

HFPLS Higher Order Frequency Pairwise Least Squares. 16, 24, 25, 27, 28, 31, 32, 39, 40, 42, 43, 71, 74, 76, 90, 97

i.i.d. independent and identically distributed. 23, 37

LCLS Low Complexity Least Squares. 11–13, 15, 22, 24, 25, 31, 40, 41, 70

LOFAR Low-Frequency Array. 3, 44

MAC medium access control. 10

MGLS Mobile Global Least Squares. 7, 11, 12, 14, 74

MLE maximum likelihood estimator. 11

MMS Magnetospheric Multiscale Mission. 57

MPLS Mobile Pairwise Least Squares. 11, 12, 14, 15, 24, 25, 27–32, 42, 43, 67, 69–73, 90, 96, 97

NASA National Aeronautics and Space Administration. 57

OLFAR Orbiting Low Frequency Array for Radio astronomy. iii–v, 3–5, 14, 15, 18, 43–58, 64–69, 71–73, 75, 76

PHY physical. 10

RMSE root mean square error. 38, 40, 66

SDMA space division multiple access. 62

SNR signal to noise ratio. 37, 42, 43, 66, 71

SPOF single point of failure. 14

TOA time of arrival. 37, 42

TOF time-of-flight. 34

TWR two way ranging. 8, 15, 16

VFO variable frequency oscillator. 36

WSN wireless sensor network. 7, 72, 73

References

[1] M. Leng and Y. Wu, "On clock synchronization algorithms for wireless sensor networks under unknown delay," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 182–190, 2010. DOI: 10.1109/TVT.2009.2028147.

[2] R. T. Rajan and A. van der Veen, "Joint ranging and synchronization for an anchorless network of mobile nodes," *IEEE Transactions on Signal Processing*, vol. 63, no. 8, pp. 1925–1940, 2015. DOI: 10.1109/TSP.2015.2391076.

[3] R. Rajan, *Relative Space-Time Kinematics of an Anchorless Network*, 3rd ed. 2016. DOI: https://doi.org/10.4233/uuid:0bcfc55b-be81-4326-855c-3a97ba126521.

[4] P. Sundaramoorthy, E. Gill, and C. Verhoeven, "Systematic identification of applications for a cluster of femto-satellites," Jan. 2010.

[5] D. J. Barnhart, T. Vladimirova, A. M. Baker, and M. N. Sweeting, "A low-cost femtosatellite to enable distributed space missions," *Acta Astronautica*, vol. 64, no. 11, pp. 1123–1143, 2009, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2009.01.025.

[6] G. McVittie and K. Kumar, "Design of a cots femtosatellite and mission," in *AIAA SPACE 2007 Conference & Exposition*. DOI: 10.2514/6.2007-6034.

[7] O. Brown and P. Eremenko, "Fractionated space architectures: A vision for responsive space," in *4th Responsive Space Conference*. Los Angeles, CA: American Institute of Aeronautics & Astronautics, Apr. 2006, p. 15.

[8] J. Leitner, "Formation flying: The future of remote sensing from space," in *International Symposium on Space Flight Dynamics*. Munich, Oct. 2004, pp. 11–15.

[9] S. Engelen, C. Verhoeven, and M. Bentum, "Olfar, a radio telescope based on nano satellites in moon orbit," in *24th Annual Conference on Small Satellites*, Utah State University, Aug. 2010, pp. 1–7.

[10] K. Robinson, R. Cox, S. Spearing, and D. Hitt, "Space launch system artemis | cubesats smallsat vanguards of exploration, science and technology," in *32nd Annual AIAA/USU Conference on Small Satellites*, Utah State University, Aug. 2020.

[11] R. T. Rajan, M. Bentum, and A. Boonstra, "Synchronization for space based ultra low frequency interferometry," in *2013 IEEE Aerospace Conference*, 2013, pp. 1–8. DOI: 10.1109/AERO.2013.6496931.

[12] M. Bentum, M. Verma, R. Rajan, A. Boonstra, C. Verhoeven, E. Gill, A. van der Veen, H. Falcke, M. K. Wolt, B. Monna, S. Engelen, J. Rotteveel, and L. Gurvits, "A roadmap towards a space-based radio telescope for ultra-low frequency radio astronomy," *Advances in Space Research*, vol. 65, no. 2, pp. 856–867, 2020, High-resolution space-borne radio astronomy, ISSN: 0273-1177. DOI: https://doi.org/10.1016/j.asr.2019.09.007.

[13] R. T. Rajan, S. Engelen, M. Bentum, and C. Verhoeven, "Orbiting low frequency array for radio astronomy," in *2011 Aerospace Conference*, 2011, pp. 1–11. DOI: 10.1109/AERO.2011.5747222.

[14] K. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 766–777, 2007. DOI: 10.1109/TCOMM.2007.894102.

[15] Y. Wu, Q. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, 2011. DOI: 10.1109/MSP.2010.938757.

[16] T. Kazaz, M. Coutino, G. J. M. Janssen, G. Leus, and A.-J. van der Veen, "Joint ranging and clock synchronization for dense heterogeneous iot networks," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 2169–2173. DOI: 10.1109/ACSSC.2018.8645210.

[17] H. Wang, H. Zeng, and P. Wang, "Linear estimation of clock frequency offset for time synchronization based on overhearing in wireless sensor networks," *IEEE Communications Letters*, vol. 20, no. 2, pp. 288–291, 2016. DOI: 10.1109/LCOMM.2015.2510645.

[18] L. Zhu, S. Shi, and X. Gu, "A consensus-based distributed clock synchronization for wireless sensor network," in *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2018, pp. 828–832. DOI: 10.1109/IWCMC.2018.8450444.

[19] X. Gu, G. Zhou, J. Li, and S. Xie, "Joint time synchronization and ranging for a mobile wireless network," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2363–2366, 2020. DOI: 10.1109/LCOMM.2020.3001138.

[20] A. R. Thompson, J. M. Moran, and G. W. Swenson, "Interferometry and synthesis in radio astronomy," in. 2017, ch. 9.5.2 Oscillator Coherence Time, p. 434. DOI: 10.1007/978-3-319-44431-4.

[21] H. Kopetz and W. Ochsenreiter, "Clock synchronization in distributed real-time systems," *IEEE Transactions on Computers*, vol. C-36, no. 8, pp. 933–940, 1987. DOI: 10.1109/TC.1987.5009516.

[22] Q. Chaudhari, E. Serpedin, and Y. .-. Wu, "Improved estimation of clock offset in sensor networks," in *2009 IEEE International Conference on Communications*, 2009, pp. 1–4. DOI: 10.1109/ICC.2009.5199072.

[23] R. T. Rajan and A.-J. van der Veen, "Joint motion estimation and clock synchronization for a wireless network of mobile nodes," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 2845–2848. DOI: 10.1109/ICASSP.2012.6288510.

[24] R. T. Rajan and A.-J. van der veen, "Joint non-linear ranging and affine synchronization basis for a network of mobile nodes," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.

[25] R. T. Rajan and A. van der Veen, "Joint ranging and clock synchronization for a wireless network," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2011, pp. 297–300. DOI: 10.1109/CAMSAP.2011.6136008.

[26] M. A. Richards, J. A. Scheer, and W. A. Holm. SciTech Publishing, 2010, ISBN: 978-1-891121-52-4. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpPMRVIBP8/principles-modern-radar/principles-modern-radar.

[27] K. Cheung, D. Divsalar, and S. Bryant, "Two-way ranging and doppler for multiple orbiting spacecraft at mars," in *2018 IEEE Aerospace Conference*, 2018, pp. 1–17. DOI: 10.1109/AERO.2018.8396652.

[28] J. R. Jensen and R. S. Bokulic, "Highly accurate, noncoherent technique for spacecraft doppler tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 3, pp. 963–973, 1999. DOI: 10.1109/7.784066.

[29] K. Ho and Y. Chan, "Geolocation of a known altitude object from tdoa and fdoa measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 770–783, 1997. DOI: 10.1109/7.599239.

[30] C. Bo, D. Enqing, L. Xiaoyang, Z. Dejing, and W. Jiaren, "A time synchronization algorithm based on bimodal clock frequency estimation," in *2012 18th Asia-Pacific Conference on Communications (APCC)*, 2012, pp. 75–78. DOI: 10.1109/APCC.2012.6388105.

[31] S. Roehr, P. Gulden, and M. Vossiek, "Method for high precision clock synchronization in wireless systems with application to radio navigation," in *2007 IEEE Radio and Wireless Symposium*, 2007, pp. 551–554. DOI: 10.1109/RWS.2007.351890.

[32] E. Dekens, S. Engelen, and R. Noomen, "A satellite swarm for radio astronomy," *Acta Astronautica*, vol. 102, pp. 321–331, 2014, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2013.11.033.

[33] R. Rajan, A. Boonstra, and M. e. a. Bentum, "Space-based aperture array for ultra-long wavelength radio astronomy," *Exp Astron*, vol. 41, pp. 271–306, 2016. DOI: https://doi-org.tudelft.idm.oclc.org/10.1007/s10686-015-9486-6.

[34] V. Karunanithi, R. Rajan, P. Sundaramoorthy, M. Verma, C. Verhoeven, M. Bentum, and E. McCune, "High data-rate inter-satellite link (isl) for space-based interferometry," English, in *70th International Astronautical Congress*, 70th International Astronautical Congress, IAC 2019, IAC 2019 ; Conference date: 21-10-2019 Through 25-10-2019, Oct. 2019. [Online]. Available: http://www.iafastro.org/publications/iac-papers/.

[35] J. Wertz, "Mission geometry; orbit and constellation design and management," pp. 527–533, Jan. 2001.

[36] G. W. Hill, "Researches in the lunar theory," *American Journal of Mathematics*, vol. 1, no. 1, pp. 5–26, 1878, ISSN: 00029327, 10806377. [Online]. Available: http://www.jstor.org/stable/2369430.

[37] J. A. van 't Hoff, *Numerical optimisation of constellation and orbit design for the OLFAR radio interferometry swarm in orbit of the Earth-Moon L4 point.* 2020. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%5C%3Ae1dd90a2-3130-49ba-aa13-0a436536e4dd.

[38] S. .-H. Mok, J. Guo, E. Gill, and R. T. Rajan, "Lunar orbit design of a satellite swarm for radio astronomy," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–9. DOI: 10.1109/AERO47225.2020.9172468.

[39] S. Jester and H. Falcke, "Science with a lunar low-frequency array: From the dark ages of the universe to nearby exoplanets," *New Astronomy Reviews*, vol. 53, no. 1, pp. 1–26, 2009, ISSN: 1387-6473. DOI: https://doi.org/10.1016/j.newar.2009.02.001.

[40] S. D'Amico, "Autonomous formation flying in low earth orbit," English, NEO, Ph.D. dissertation, Delft University of Technology, 2010, ISBN: 978-90-5335-253-3.

[41] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002. DOI: 10.1109/TWC.2002.804190.

[42] P. Li, S. Xu, K. Sun, X. Qiu, and F. Qi, "A path planning method of wireless sensor networks based on service priority," in *2017 13th International Conference on Network and Service Management (CNSM)*, 2017, pp. 1–4. DOI: 10.23919/CNSM.2017.8255984.

[43] D. Baird. (2021). "Record-breaking satellite advances nasa's exploration of high-altitude gps," [Online]. Available: https://www.nasa.gov/feature/goddard/2019/record-breaking-satellite-advances-nasa-s-exploration-of-high-altitude-gps (visited on 07/28/2021).

[44] European Space Agency. (2021). "Galileo will help lunar pathfinder navigate around moon," [Online]. Available: https://www.esa.int/Applications/Navigation/Galileo_will_help_Lunar_Pathfinder_navigate_around_Moon (visited on 07/28/2021).

[45] S. Rai and S. Sharma, "Determining minimum spanning tree in an undirected weighted graph," in *2015 International Conference on Advances in Computer Engineering and Applications*, 2015, pp. 637–642. DOI: 10.1109/ICACEA.2015.7164769.

A | Simulation

This appendix illustrates the implementation of a software testbench to simulate different algorithms. Algorithm 1 is the main simulation which makes use of the other algorithms 2 - 9 that contain function definitions called by the main simulation or other functions.

Program 1 Simulation main

Inputs:
$N, N_{dim}, x_e[\text{km}], y_e\left[\frac{\text{m}}{\text{s}}\right], \phi_e[\text{s}], \omega_e, K, \text{SNR}_{\text{dB}}[\text{dB}], c\left[\frac{\text{m}}{\text{s}}\right], t_{max}[\text{s}], f_{min}[\text{Hz}], f_{max}[\text{Hz}]$
$\boldsymbol{\phi}, \boldsymbol{\omega} = \text{InitClocks}(N, \phi_e, \omega_e)$
$\mathbf{x}, \mathbf{y} = \text{InitNodesLinear}(N, N_{dim}, x_e, y_e)$
$N_{messages} = \frac{N(N-1)}{2}K$
$\mathbf{t}, \mathbf{f}, \mathbf{L} = \text{MessageSetup}(N_{messages}, t_{max}, f_{min}, f_{max})$
$\mathbf{t}_{ij}, \mathbf{f}_{ij}, \mathbf{e}_{ij} = \text{MessageExchange}(N, N_{messages}K, \mathbf{x}, \mathbf{y}, \mathbf{t}, \mathbf{f}, \mathbf{L}, \boldsymbol{\omega}, \boldsymbol{\phi})$ ▷ $\forall i, j \leqslant N \wedge i \neq j$
for $\forall i, j \leqslant N \wedge i \neq j$ **do**
    $\mathbf{t}_{ij}, \mathbf{f}_{ij} = \text{AddNoise}(\mathbf{t}_{ij}, \mathbf{f}_{ij}, \text{SNR}_{\text{dB}}, c, x_e, y_e, f_{min}, f_{max})$
$\hat{\boldsymbol{\phi}}_{\boldsymbol{L}}, \hat{\boldsymbol{\omega}}_{\boldsymbol{L}}, \hat{\boldsymbol{d}}_{\boldsymbol{L}} = \text{MPLS}(\mathbf{t}_{ij}, \mathbf{t}_{ji}, \mathbf{e}_{ij}, L = 1, i_{ref})$ ▷ LCLS
$\hat{\boldsymbol{\phi}}_{\boldsymbol{M}}, \hat{\boldsymbol{\omega}}_{\boldsymbol{M}}, \hat{\boldsymbol{d}}_{\boldsymbol{M}} = \text{MPLS}(\mathbf{t}_{ij}, \mathbf{t}_{ji}, \mathbf{e}_{ij}, L, i_{ref})$ ▷ MPLS
$\hat{\boldsymbol{\omega}}_{\boldsymbol{F}}, \hat{\boldsymbol{v}}_{\boldsymbol{F}} = \text{FPLS}(\mathbf{t}_{ij}, \mathbf{t}_{ji}, \mathbf{f}_{ij}, \mathbf{f}_{ji}, \mathbf{e}_{ij}, i_{ref})$ ▷ FPLS

Program 2 Clock initialization affine clock model

function InitClocks(N, ϕ_e, ω_e)
    Initialize:
    $\boldsymbol{\phi}, \boldsymbol{\omega} \in \mathbb{R}^N$
    $\boldsymbol{\phi} \sim \boldsymbol{\mathcal{U}}(\phi_e, \phi_e)$ ▷ clock offsets
    $\boldsymbol{\omega} \sim \boldsymbol{\mathcal{U}}(1 - \omega_e, 1 + \omega_e)$ ▷ clock skews
    return $\boldsymbol{\phi}, \boldsymbol{\omega}$

Program 3 Node initialization linear movement

function InitNodesLinear(N, N_{dim}, x_e, y_e)
    Initialize:
    $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{N_{dim} \times N}$
    $\mathbf{x} \sim \boldsymbol{\mathcal{U}}(-x_e, x_e)$ ▷ initial positions
    $\mathbf{y} \sim \boldsymbol{\mathcal{U}}(-y_e, y_e)$ ▷ velocities
    return $\mathbf{x}, \mathbf{y}$

Program 4 Node position linear movement

function PosLinear(t,i,$\mathbf{x}, \mathbf{y}$)
  $\mathbf{x_i} = \mathrm{col}_i(\mathbf{x})$
  $\mathbf{y_i} = \mathrm{col}_i(\mathbf{y})$
  $\mathbf{p}_i = \mathbf{x}_i + t\mathbf{y}_i$
  return $\mathbf{p_i}$

Program 5 Message setup

function MessageSetup($N_{messages}$,t_{max}, f_{min}, f_{max})
  Initialize:
  $\mathbf{t}, \mathbf{f} \in \mathbb{R}^{N_{messages}}, \quad \mathbf{L} \in \mathbb{Z}_{\not\Delta}^{\bar{N}\times 2}$
  $\mathbf{t} = \left[0, \frac{1}{N_{messages}} t_{max}, \ldots, \frac{N_{messages}-1}{N_{messages}} t_{max}\right]^T$ ▷ transmission times
  $\mathbf{f} = \left[f_{min}, f_{min} + \frac{1}{N_{messages}}(f_{max} - f_{min}), \ldots, f_{min} + \frac{N_{messages}-1}{N_{messages}}(f_{max} - f_{min})\right]^T$ ▷ transmission frequencies
  $a = 1$
  for $i = 1, \ldots, N$ **do**
    for $j = 1, \ldots, N$ **do**
      if $i \neq j$ **then**
        $\mathbf{L}_{a,1} = i$
        $\mathbf{L}_{a,2} = j$
        $a = a + 1$
  return $\mathbf{t}, \mathbf{f}, \mathbf{L}$

Program 6 Message exchange

function MessageExchange($N, N_{messages}K, \mathbf{x}, \mathbf{y}, \mathbf{t}, \mathbf{f}, \mathbf{L}$, $\boldsymbol{\omega}$, $\boldsymbol{\phi}$)
  Initialize:
  $\mathbf{c} = \mathbf{0}^{N\times N}, \mathbf{t}_{ij}, \mathbf{f}_{ij}, \mathbf{e}_{ij} \in \mathbb{R}^K$ ▷ $\forall i, j \leqslant N \wedge i \neq j$
  $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \triangleq [\boldsymbol{\omega}^{\odot -1}, -\boldsymbol{\phi} \odot \boldsymbol{\omega}^{\odot -1}]$ ▷ calibration parameters
  for $a = 1, \ldots, N_{messages}$ **do**
    $i = \mathbf{L}_{a,1}$ ▷ transmitting node
    $j = \mathbf{L}_{a,2}$ ▷ receiving node
    $\mathbf{c}_{i,j} = \mathbf{c}_{i,j} + 1$
    $k = \mathbf{c}_{i,j}$
    $\mathbf{e}_{ij,k} = 1$
    $\mathbf{e}_{ji,k} = -1$
    $\mathbf{t}_{ij,k} = \mathbf{t}_a$ ▷ local time at transmitting node
    $t = \boldsymbol{\alpha}_i \mathbf{t}_a + \boldsymbol{\beta}_i$ ▷ true transmission time
    $\tau = \mathrm{PropagationTime}(t, c, \mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j)$ ▷ propagation delay
    $\mathbf{t}_{ji,k} = \boldsymbol{\omega}_j(t + \tau) + \boldsymbol{\phi}_j$ ▷ local time at receiving node
    $\mathbf{f}_{ij,k} = \mathbf{f}_a$ ▷ local frequency at transmitting node
    $f = \boldsymbol{\omega}_i \mathbf{f}_a$ ▷ true transmission frequency
    $v_r = \mathrm{RelativeVelocity}(t, \tau, \mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j)$ ▷ relative velocity
    $\mathbf{f}_{ji,k} = \boldsymbol{\alpha}_j f(1 + \frac{v_r}{c})$ ▷ local frequency at receiving node
  return $\mathbf{t}_{ij}, \mathbf{f}_{ij}, \mathbf{e}_{ij}$ ▷ $\forall i, j \leqslant N \wedge i \neq j$

Program 7 Noise addition

function ADDNOISE($\mathbf{t}_{ij}, \mathbf{f}_{ij}, \mathrm{SNR}_{\mathrm{dB}}, c, x_e, y_e, f_{min}, f_{max}$)
  Initialize:
  $\sigma_t, \sigma_f, \mathrm{SNR} = 10^{\frac{\mathrm{SNR}_{\mathrm{dB}}}{10}}$
  $\sigma_t = c^{-1}\mathrm{SNR}^{-1}\sqrt{\frac{1}{12}(2x_e)^2}$
  $\sigma_f = \frac{f_{min}+f_{max}}{2}c^{-1}\mathrm{SNR}^{-1}\sqrt{\frac{1}{12}(2y_e)^2}$
  $\mathbf{t}_{ij} = \mathbf{t}_{ij} + \mathcal{N}(0, \sigma_t)$
  $\mathbf{f}_{ij} = \mathbf{f}_{ij} + \mathcal{N}(0, \sigma_f)$
  return $\mathbf{t}_{ij}, \mathbf{f}_{ij}$ ▷ $\forall i, j \leqslant N \wedge i \neq j$

Program 8 MPLS as described in [3]

function MPLS($\mathbf{t}_{ij}, \mathbf{t}_{ji}, \mathbf{e}_{ij}, L, i$)
  Initialize:
  $\hat{\boldsymbol{\phi}}_M, \hat{\boldsymbol{\omega}}_M, \hat{\boldsymbol{d}}_M \in \mathbb{R}^N$
  $\hat{\boldsymbol{\omega}}_{M,i} = 1$
  $\hat{\boldsymbol{\phi}}_{M,i} = 0$
  $\hat{\boldsymbol{d}}_{M,i} = 0$
  for $j = 1, \ldots, N$ **do**
    if $i \neq j$ **then**
      $\mathbf{E}_{ij} = \mathrm{diag}(\mathbf{e}_{ij})$
      $\mathbf{V}_{ij} = \begin{bmatrix}\mathbf{t}_{ij}^{\odot 0} & \mathbf{t}_{ij}^{\odot 1} & \mathbf{t}_{ij}^{\odot 2} & \ldots & \mathbf{t}_{ij}^{\odot L-1}\end{bmatrix}$
      $\begin{bmatrix}\hat{\omega}_j \\ \hat{\phi}_j \\ \hat{d}_j\end{bmatrix} = \begin{bmatrix}\mathbf{t}_{ji} & \mathbf{1}_K & -\mathbf{E}_{ij}\mathbf{V}_{ij}\end{bmatrix}^{\dagger}\mathbf{t}_{ij}$
      $\hat{\boldsymbol{\omega}}_{M,j} = \hat{\omega}_j$
      $\hat{\boldsymbol{\phi}}_{M,j} = \hat{\phi}_j$
      $\hat{\boldsymbol{d}}_{M,j} = \hat{d}_j$
  return $\hat{\boldsymbol{\phi}}_M, \hat{\boldsymbol{\omega}}_M, \hat{\boldsymbol{d}}_M$

Program 9 FPLS

function FPLS($\mathbf{f}_{ij}, \mathbf{f}_{ji}, \mathbf{e}_{ij}, L, i, N$)
  Initialize:
  $\hat{\boldsymbol{\omega}}_F, \hat{\boldsymbol{v}}_F \in \mathbb{R}^N$
  $\mathbf{g}_{ij} = 0.5(\mathbf{e}_{ij} + \mathbf{1})$
  $\mathbf{g}_{ji} = 0.5(\mathbf{e}_{ij} - \mathbf{1})$
  $\hat{\boldsymbol{\omega}}_{F,i} = 1$
  $\hat{\boldsymbol{v}}_{F,i} = 0$
  for $j = 1, \ldots, N$ **do**
    if $i \neq j$ **then**
      $\begin{bmatrix}\hat{\omega}_j \\ \hat{v} \\ \star\end{bmatrix} = \begin{bmatrix}\mathbf{f}_{ji} & -c^{-1}\mathbf{g}_{ij} \odot \mathbf{f}_{ij} & c^{-1}\mathbf{g}_{ji} \odot \mathbf{f}_{ji}\end{bmatrix}^{\dagger}\mathbf{f}_{ij}$
      $\hat{\boldsymbol{\omega}}_{F,j} = \hat{\omega}_j$
      $\hat{\boldsymbol{v}}_{F,j} = \hat{v}$
  return $\hat{\boldsymbol{\omega}}_F, \hat{\boldsymbol{v}}_F$

B | Linear movement

This part of the appendix introduces a linear movement model. Let the position of node i at time t be

$$\mathbf{p}_i(t) \triangleq \mathbf{x}_i + t\mathbf{y}_i \qquad \mathbf{p}_i \in \mathbb{R}^{N_{dim}\times 1} \tag{B.1}$$

where N_{dim} denotes the dimension of the space. In the following sub-part the propagation time for a transmission from node i to node j and their relative velocities are derived.

B.1 Propagation time

Let τ be the propagation delay for a message being sent at time t from node i to node j. The position of node j upon arrival of the signal is then

$$\mathbf{p}_j(t+\tau) = \mathbf{x}_j + (t+\tau)\mathbf{y}_j \tag{B.2}$$

In order to do correct time-stamping during the simulation, this delay needs to be computed. The distance between the point where the message originated and the receiving node j is thus

$$\begin{aligned} d_r(t,\tau) &= \|\mathbf{p}_i(t) - \mathbf{p}_j(t+\tau)\| \\ d_r(t,\tau) &= \|\mathbf{x}_i + t\mathbf{y}_i - \mathbf{x}_j - t\mathbf{y}_j - \tau\mathbf{y}_j\| \end{aligned} \tag{B.3}$$

We now define

$$\begin{aligned} \mathbf{e} &\triangleq -\mathbf{y}_j \\ \mathbf{f} &\triangleq \mathbf{x}_i + t\mathbf{y}_i - \mathbf{x}_j - t\mathbf{y}_j \end{aligned} \tag{B.4}$$

and can then express

$$d_r(\tau) = \|\mathbf{e}\tau + \mathbf{f}\| \tag{B.5}$$

The distance that the signal has propagated from the point of origin can be expressed as

$$d_s(\tau) = \tau v \tag{B.6}$$

where v is the propagation speed of the wave. The message arrives at node j when the distance of node j and the distance of the wavefront to the origin $\mathbf{p}_i(t)$ are are equal. To find τ, we thus

need to solve

$$\begin{aligned} d_s(\tau) &= d_r(\tau) \\ \tau v &= \|\mathbf{e}\tau + \mathbf{f}\| \\ (\tau v)^2 &= \|\mathbf{e}\tau + \mathbf{f}\|^2 \\ (\tau v)^2 &= (\mathbf{e}\tau + \mathbf{f})^T (\mathbf{e}\tau + \mathbf{f}) \\ \tau^2 v^2 &= \tau^2 \mathbf{e}^T\mathbf{e} + \tau 2\mathbf{e}^T\mathbf{f} + \mathbf{f}^T\mathbf{f} \\ 0 &= \tau^2 \underbrace{\left(\mathbf{e}^T\mathbf{e} - v^2\right)}_{a} + \tau \underbrace{2\mathbf{e}^T\mathbf{f}}_{b} + \underbrace{\mathbf{f}^T\mathbf{f}}_{c} \end{aligned} \tag{B.7}$$

The result is a scalar quadratic equation with the following solutions:

$$\tau_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{B.8}$$

Out of $\tau_{1,2}$, we need to find the solution that fullfils $\tau > 0$. As the nodes move slower than the propagation speed of electromagnetic waves they exchange, $\mathbf{e}^T\mathbf{e} < v^2$ holds. Thus, $a < 0$ and the numerator needs to be negative. As $b < \sqrt{b^2 - 4ac}$, the solution is

$$\tau(t, v, \mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j) = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \tag{B.9}$$

B.2 Relative velocity

In this section, the relative velocity of nodes i and j is derived. The distance between 2 nodes at time t can be expressed as follows:

$$\begin{aligned} d_r(t) &= \|\mathbf{p}_i(t) - \mathbf{p}_j(t)\| \\ d_r(t) &= \|t(\mathbf{y}_i - \mathbf{y}_j)\mathbf{x}_i - \mathbf{x}_j\| \\ d_r(t) &= \sqrt{(t(\mathbf{y}_i - \mathbf{y}_j) + \mathbf{x}_i - \mathbf{x}_j)^T (t(\mathbf{y}_i - \mathbf{y}_j) + \mathbf{x}_i - \mathbf{x}_j)} \\ d_r(t) &= \sqrt{t^2 \underbrace{(\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{y}_i - \mathbf{y}_j)}_{a} + t \underbrace{2(\mathbf{y}_i - \mathbf{y}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}_{b} + \underbrace{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}_{c}} \\ d_r(t) &= \sqrt{t^2 a + tb + c} \end{aligned} \tag{B.10}$$

Note, that a, b and c are redefined here and have different meanings than in appendix B.1. The relative velocity between nodes i and j is the time-derivative of their distance.

$$\begin{aligned} v_r(t) &= \frac{\mathrm{d}d_r(t)}{\mathrm{d}t} \\ v_r(t, \mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j) &= \frac{ta + 0.5b}{\sqrt{t^2 a + tb + c}} \end{aligned} \tag{B.11}$$

For a message transmitted at time t and propagating for time τ the average relative velocity can be expressed as:

$$v_r(t, \tau, \mathbf{x}_i, \mathbf{x}_j, \mathbf{y}_i, \mathbf{y}_j) \approx \frac{v_r(t) + v_r(t + \tau)}{2} \tag{B.12}$$

C | Relative movement of nodes

This part of the appendix extends subsection 4.3.3 with additional plots showing pairwise distance, velocity and acceleration for one node w.r.t. all other nodes in the network. The purpose of this appendix is to illustrate the relative motion of the nodes involved in the simulation.

While the nodes follow a linear dynamic model as described in subsection 4.3.3, the resulting relative distances and their derivatives are inherently nonlinear. Figures C.1–C.3 show plots of a distance, velocity and acceleration for the first 500s of simulation time. During the first part, even the relative accelerations exhibit nonlinear behaviour, albeit rather small in magnitude. In contrast, towards the end of the illustrated time period, the relative distance converges to linear function in time, as the relative acceleration approaches 0. As time progresses, the velocity vectors of the individual nodes become dominant over their initial positions which explains the observed behaviour.

Estimating clock parameters is more challenging in the nonlinear region of the pairwise distance function towards the beginning of the time scale. The simulation conducted in subsection 4.3.3 takes place in the first 3 seconds, thus in the nonlinear region. This necessitates the use of higher order algorithms such as Mobile Pairwise Least Squares (MPLS) and Higher Order Frequency Pairwise Least Squares (HFPLS) that can approximate the nonlinear relative motions. Yet, it could be argued that the period of 3 seconds is so short that the effects of the nonlinearities are rather small. To best illustrate the advantages of higher order algorithms one could extend the period over which to spread the message exchange for time synchronization.

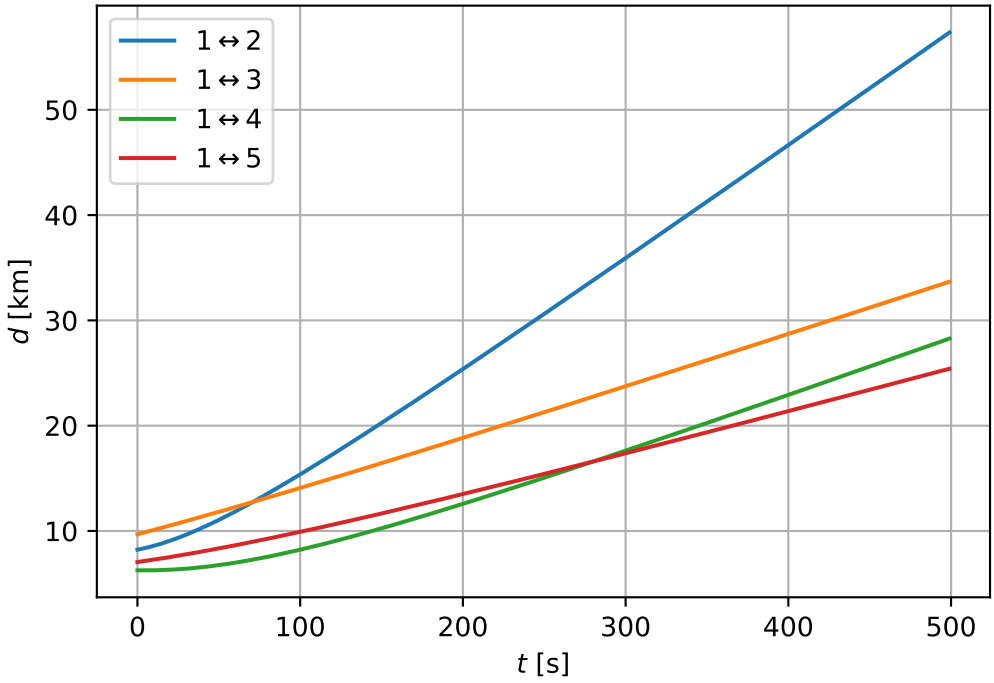


Figure C.1: Pairwise distance over time for all nodes w.r.t node 1

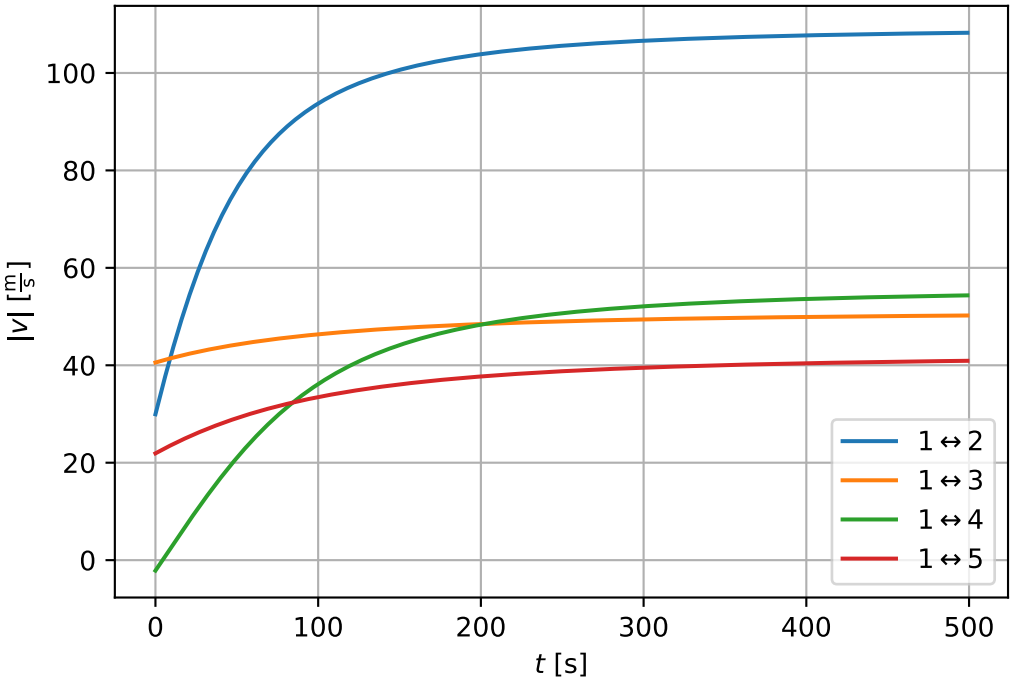


Figure C.2: Pairwise absolute relative velocity over time for all nodes w.r.t node 1

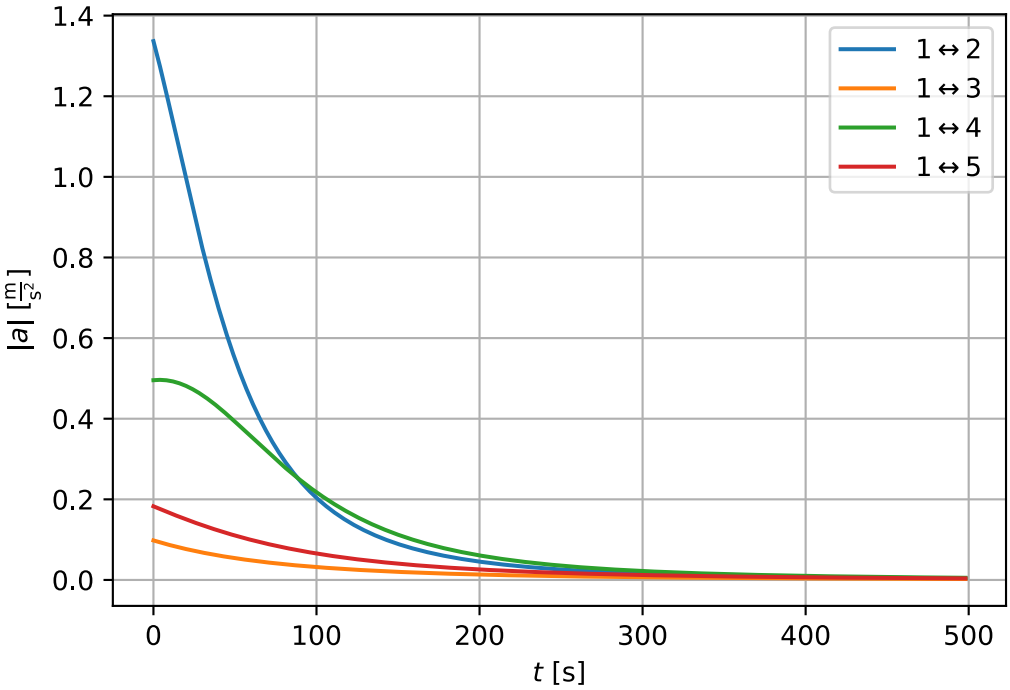
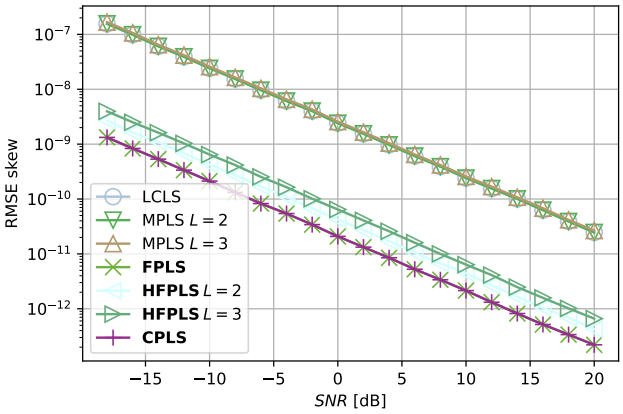


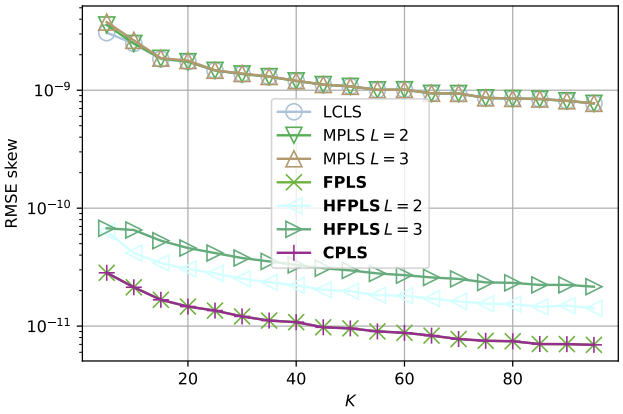
Figure C.3: Pairwise absolute relative acceleration over time for all nodes w.r.t node 1

D | Stationary network results

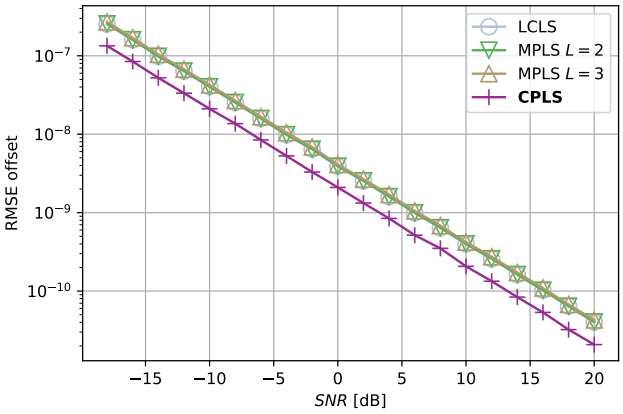
This part of the appendix extends subsection 4.3.3 with additional plots showing simulation results for an immobile network using an adapted SNR definition.



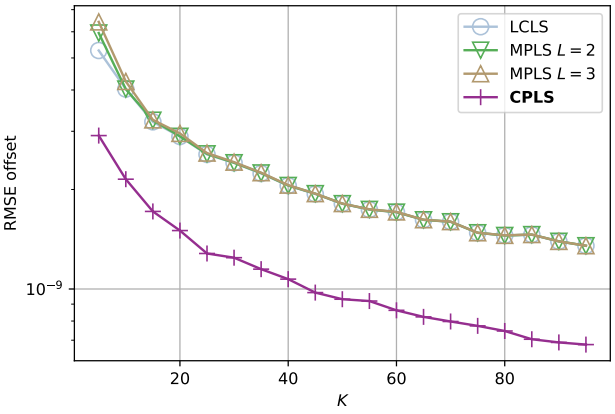
(a) clock skew ω



(b) clock skew ω

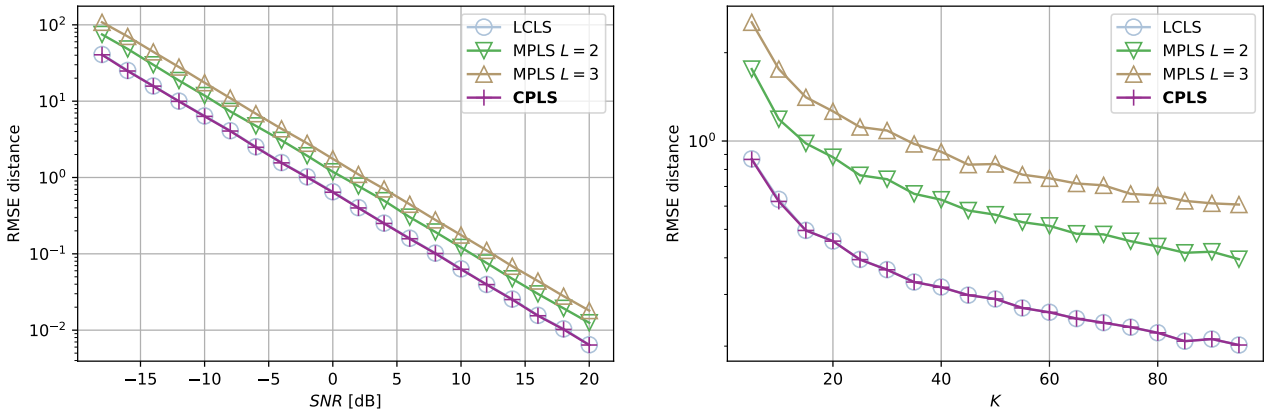


(c) clock offset ϕ



(d) clock offset ϕ

Figure D.1: Clock parameter estimation error using pairwise algorithms time domain and frequency domain in an immobile network; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for $SNR = 0$dB

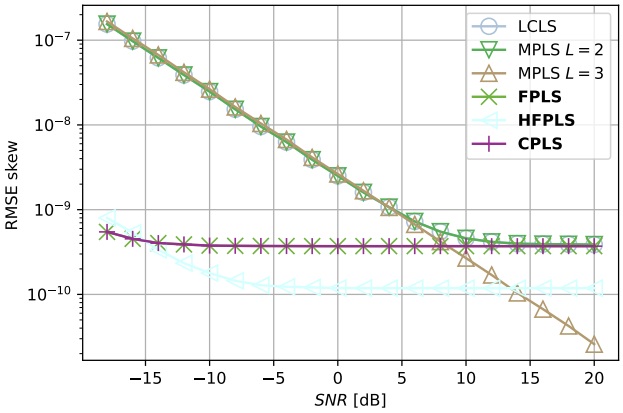


(a) distance (b) distance

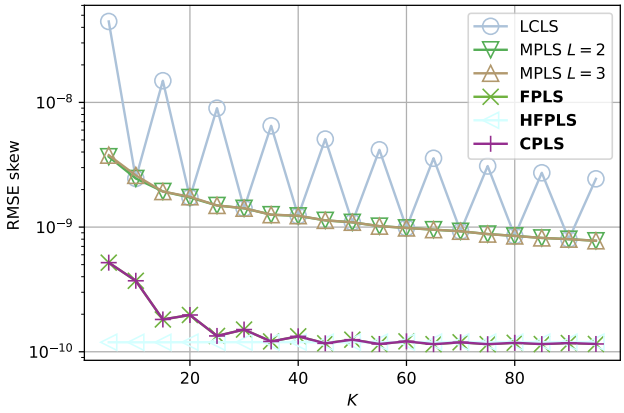
Figure D.2: Distance estimation error using pairwise algorithms in time domain and frequency domain in an immobile network; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for $SNR = 0$dB

E | Mobile network additional results

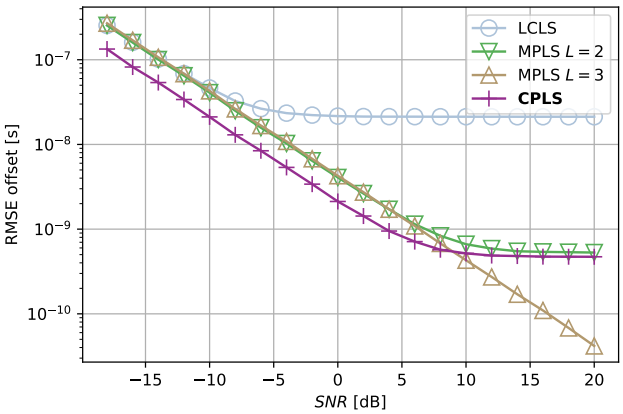
This part of the appendix extends subsection 4.3.3 with additional plots showing simulation results for additional algorithms and distances and velocities.



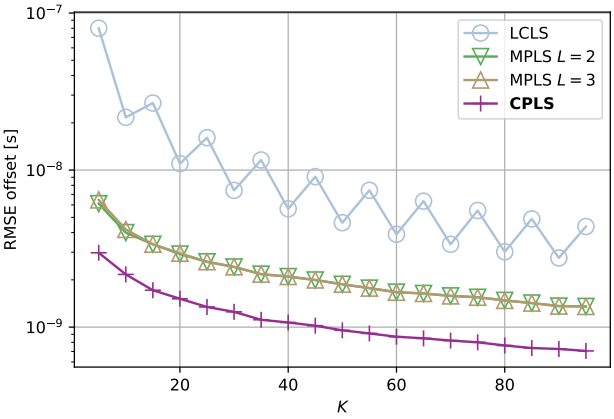
(a) clock skew ω



(b) clock skew ω



(c) clock offset ϕ

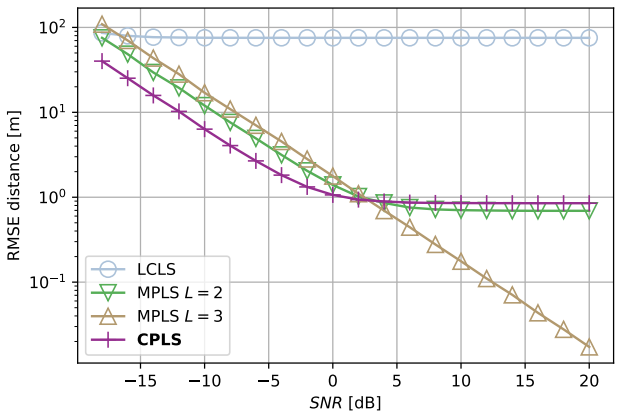


(d) clock offset ϕ

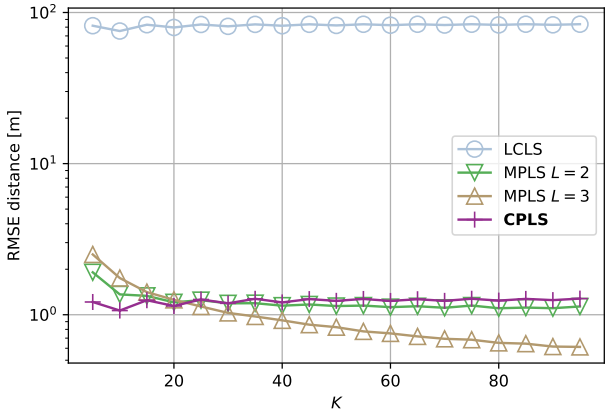
Figure E.1: Clock parameter estimation error using pairwise algorithms time domain and frequency domain; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for SNR = 0dB

Figure E.2 shows the the error of pairwise ranging parameters, namely distance in the first row and velocity in the second row. Figure E.2a shows distance error for varying noise. As

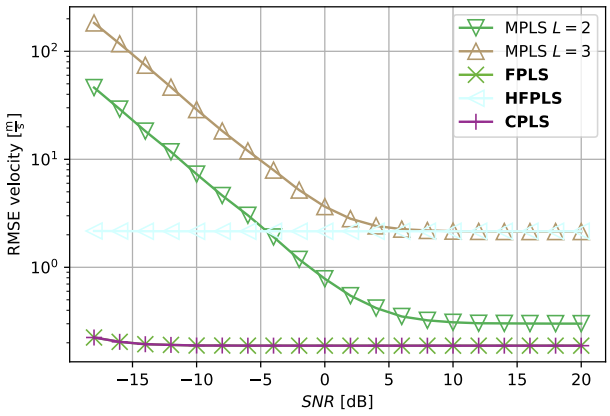
solely frequency-based methods cannot estimate distance, existing time domain algorithms are represented alongside the Combined Pairwise Least Squares. In general, similar behaviour as for the clock offset can be observed for the distance, with the difference that the SNR where mobile time domain methods begin to outperform CPLS at $\approx$ 3dB which again coincides with what can be observed for velocity measurements. For varying number of communications at 0dB SNR — depicted in figure E.2b — Combined Pairwise Least Squares exhibits similar behaviour as could be observed for skew and velocity estimations in frequency domain that this method is based on: Increasing K has no positive effect on the residual error. Conversely, mobile time domain methods decrease in error making them superior to the combined method for $K \gtrapprox 22$.



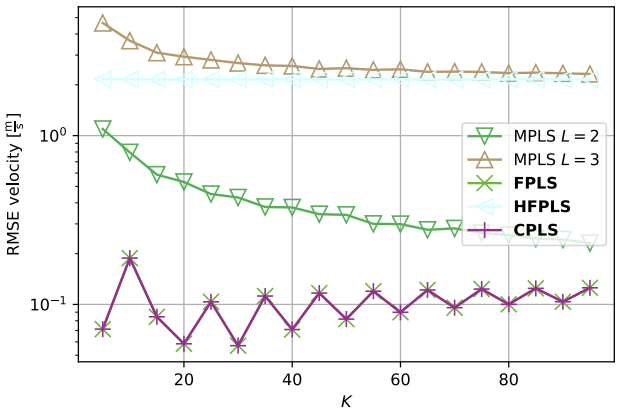
(a) distance



(b) distance



(c) velocity



(d) velocity

Figure E.2: Distance and velocity estimation error using pairwise algorithms in time domain and frequency domain; left column shows RMSE vs. SNR (left) for $K = 10$; right column shows RMSE vs. number of communications K for SNR = 0dB

The estimated velocity for varying noise level is visualized in figure E.2c. Methods operating at up to velocity order (MPLS $L = 2$, FPLS) exhibit similar behaviour in the sense that both reach an RMSE of $2\frac{\mathrm{m}}{\mathrm{s}}$. MPLS starts out at a higher error decreasing linearly on a log scale until around 0dB, whereas FPLS holds the constant error for the whole SNR range. Matching

observations for the clock skew, algorithms operating at or above acceleration order (MPLS $L = 3$, HFPLS) exhibit different performance. They decrease linearly on a log scale — for the whole depicted SNR range in case of MPLS with $L = 3$ and up until around 0dB for HFPLS. In absolute terms HFPLS delivers far superior performance in velocity estimation than all other methods, being only outperformed by MPLS with $L = 3$ for SNR $\geqslant$ 20dB.

Figure E.2d with the 0dB SNR cut for varying K confirms the observations from the SNR sweep and other plots. The frequency domain methods and MPLS with $L = 2$ do not benefit from increased K, whereas MPLS with $L = 3$ does, outperforming MPLS with $L = 2$ and FPLS starting at around $K = 20$. HFPLS outperforms all other methods for the whole depicted K range.