DISSERTATION

Distributed Subspace Tracking in Wireless Sensor Networks

ausgeführt zum Zwecke der Erlangung des akademischen Grades
einer Doktorin der technischen Wissenschaften

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Carolina del Socorro Reyes Membreño
Yppenplatz 5/8
Wien
Österreich

geboren am 05. 05. 1983 in Chontales, Nicaragua
Matrikelnummer: 0756895

Wien, im Dezember 2013

Begutachter:

Univ. Prof. Dr. Christoph Mecklenbräuker
Institut für Nachrichtentechnik und Hochfrequenztechnik
Technische Universität Wien
Österreich

Ao. Univ. Prof. Dr. Thibault Hilaire
Laboratoire d'Informatique de Paris 6 (LIP6)
Université Pierre et Marie Curie France (UPMC)
France

Abstract

WIRELESS Sensor Networks (WSN) emerged within the wireless communication technology as a tool to extend our capability to explore, monitor and control our physical surrounding. They are spatial distributed autonomous sensors with a wireless communications link. These sensors are typically located in inhospitable environments and left completely unattended, meaning that their lifetime in terms of communication and computation resources is limited by the battery capacity. Furthermore, damaged sensors are often difficult to replace, which makes a robust protocol desirable.

Motivated by these limitations, this thesis addresses the more specific topic of decentralized algorithms (i.e. removing the need for a data fusion center) for WSNs which aim to measure and classify a radio wave. The focus is to develop a distributed estimation scheme that is energy efficient and concurrently dispenses high quality data acquisition and estimation, even if only a small party of the WSN contributes. To this end, two distributed subspace-based algorithms are devised, one sharing the observation vector $\underline{\mathbf{y}}$ and the other sharing the sample covariance matrix $\widehat{\mathbf{R}}^{yy}$ between the nodes. The Average Consensus (AC) protocol enforces that all sensors within the network converge towards the mean of all observations.

The performance of the newly developed algorithms is evaluated in terms of their tracking capabilities, the Root Mean Square Error (RMSE) and the Principal Angles between the Subspaces (PABS) for several scenarios. The algorithm sharing the observation vector $\underline{\mathbf{y}}$ outperforms the algorithm sharing the sample covariance matrix $\widehat{\mathbf{R}}^{yy}$ and demonstrates overall good tracking capabilities. Finally, a mathematical proof of convergence analysis for the centralized PAST algorithm based on Singular Value Decomposition (SVD) is presented.

Zusammenfassung

DRAHTLOSE Sensornetzwerke (WSN) haben sich innerhalb der drahtlosen Kommunikationstechnologie als Werkzeuge etabliert, um unsere Umgebung besser erforschen, überwachen und kontrollieren zu können. WSN bestehen aus vielen einzelnen räumlich verteilten autarken Sensoren die typischerweise in unwirtlichen Gebieten betrieben werden. Dies hat zur Folge, dass deren Kommunikation und Rechenleistung durch den notwendigen Energieverbrauch, der normalerweise durch eine Batterie gedeckt werden muss, limitiert wird. Beschädigte Sensoren sind oft schwierig auszutauschen, dies macht ein robustes Protokoll wünschenswert.

Diese Dissertation thematisiert die angeführten Limitationen für den speziellen Fall von drahtlosen Sensornetzwerken, die Radiowellen vermessen. Durch die Entwicklung von dezentralisierten Algorithmen wird Energie eingespart, und die Fehleranfälligkeit gesenkt. Der Fokus liegt darauf, die Signalschätzung auf alle Sensoren im Netwerk energieeffizient und ohne Genauigkeitsverlust zu verteilen. Zu diesem Zweck wurden zwei verteilte Unterraumschätzungsalgorithmen formuliert, die unterschiedlich Information zwischen den Sensoren austauschen: Der eine verteilt unter den Sensoren den Observationsvektor $\underline{\mathbf{y}}$, der andere die Signal-Kovarianzmatrix $\widehat{\mathbf{R}}^{yy}$. Das Average Consensus (AC) Protokoll garantiert, dass das gesamte Netzwerk gegen den Mittelwert alle Observationen konvergiert.

Die Bewertung der neuen Algorithmen erfolgte durch Simulationen für verschiedene Szenarien mittels der Signalverfolgungskapazität, der quadratischen Mittelwertsabweichung (RMSE) und der Differenz der Unterraumwinkeln (PABS).

Es hat sich gezeigt, dass es effizienter ist den Observationsvektor $\underline{\mathbf{y}}$ zu teilen. Dieser Algorithmus liefert gute Ergebnisse und übertrifft jenen mit verteilter Signal-Kovarianzmatrix $\widehat{\mathbf{R}}^{yy}$ in allen simulierten Szenarien.

Abschließend wird mittels einer Konvergenzanalyse basierend auf einer Singulärwertzerlegung die Stabilität des Algorithmus validiert.

This work was funded by the fFORTE-Women in Technology Program at Vienna University of Technology and the Austrian Ministry for Science and Research. Support was also provided by the FWF project under Awards S10611-N13 within the National Research Network SISE.

Acknowledgement

Ayer fui al parque. Había un pasamanos y jugué hasta que me salieron ampollas... y me acordé de vos... mi niña tan esforzada. Nunca te rendiste ante una dificultad antes, ¡No vas a empezar ahora!
CJRM

First and foremost I want to thank my supervisor Christoph Mecklenbräuker for giving me the opportunity to enroll in the doctorate program at the Vienna University of Technology. I am thankful for his trust and the freedom he has given me to find my own path in the immense world of research. For the motivation and support to follow my dreams, I am beholden to him.

My deepest gratitude goes to my second supervisor Thibault Hilaire. His patience, time for discussion and feedback were key for the completion of this work.

I am also indebted to Antoine Souloumiac and Markus Rupp. Without their support, part of this thesis would have not been accomplished.

None of this would have been possible without the amazing working atmosphere at the Institute of Telecommunications. I thank all my colleagues, specially the "Mensa-Walk" people, for their support throughout these years.

To my family at both sides of the globe: I can not imagine anything of this without you backing me up! For your patience, love and support I am very thankful. To you... my better half, words can not express my gratefulness for "just" everything.

Vienna, December 2013

Carolina Reyes Membreño

Phone: (+43) 699-1953-2222
e-mail: creyes@nt.tuwien.ac.at

Contents

Notation and symbols

$\underline{\mathbf{1}}$	vector of ones
$\underline{\mathbf{x}}$	bold, lowercase and underscore variables are defined as vectors
$\mathbf{W}$	bold and uppercase variables are defined as matrices
$.^T$	transpose
$.^H$	hermitian
$\Longleftrightarrow$	if and only if
$\times$	general multiplication
$E\{\cdot\}$	expected value
$\|\| \cdot \|\|_F$	Frobenius norm
$\langle \cdot \rangle$	inner product
$\perp$	orthogohal to
$\bigotimes$	Kronecker product
N	network size
i	sensor index
$\mathcal{H}$	network's graph
$\mathcal{E}$	edge/ connection between sensors
$\mathcal{N}_i$	neighborhood of sensor i
$\mathbf{A}$	adjacency/connectivity matrix
$\mathbf{D}$	degree matrix
$\mathbf{L}$	laplacian matrix
$\vec{p}$	position vector
$\vec{\kappa}$	wave number
$\vec{\xi}$	slowness factor
θ, ϕ	azimuth and elevation
$\Delta_{i,l}$	time propagation delay
r	number of uncorrelated signals
$\underline{\mathbf{x}}$	observation vector
$\boldsymbol{\Upsilon}$	steering matrix
$\underline{\mathbf{s}}$	signal vector
$\underline{\mathbf{n}}$	additive white gaussian noise
$\underline{\boldsymbol{\upsilon}}(\theta)$	array steering vector
$\mathbf{R}^{xx}$	data covariance matrix
σ^2	variance
$\mathbf{U}_s$	signal subspace
$\mathbf{U}_o$	noise subspace
$\boldsymbol{\Lambda}_s$	signal eigenvectors
$\boldsymbol{\Lambda}_o$	noise eigenvectors
$\mathbf{P}$	MUSIC spatial spectrum
$\widehat{\mathbf{R}}$	sample covariance matrix
β	forgetting factor
$\mathbf{W}$	estimated signal subspace
$\underline{\mathbf{y}}$	signal vector variable
$\underline{\mathbf{e}}$	estimation error vector
$\mathbf{G}$	weight matrix

$\mathbf{S}$	selection matrix
$\widetilde{\underline{\mathbf{y}}}$	average consensus of local $\underline{\mathbf{y}}$
$\widetilde{\widetilde{\mathbf{R}}}$	average consensus of local $\underline{\mathbf{y}}$
γ	step-size
$\bar{\sigma}$	singular values related to $\widehat{\mathbf{R}}^{yy}$
σ	singular values related to $\mathbf{W}$
$\mathbf{V}$	matrix containing the signal subspace $\mathbf{W}$

Abbreviations

AC	Average Consensus
DOA	Direction Of Arrival
EVD	Eigen Value Decomposition
MC	Monte Carlo
MEP	Modified Eigen Problem
MUSIC	MUltiple SIgnal Classification
ODE	Ordinary Differential Equation
PABS	Principal Angles Between Subspaces
PAST	Projection Approximation Subspace Tracking
PCA	Principal Component Analysis
RLS	Recursive Least Square
RMSE	Root Mean Square Error
SNR	Signal to Noise Ratio
SVD	Singular Value Decomposition
ULS	Uniform Linear Array
WSN	Wireless Sensor Networks

Chapter 1

Introduction

Wireless Sensor Networks (WSNs) have a multitude of application areas. They can be deployed in remote and inhospitable locations, and serve as "eyes and ears" for safety, research, industry and military concerns [1, 2]. By way of example, avalanche or land slide early detection systems, biological research habitat monitoring, the monitoring of production parameters, and even tracking of conditions inside a nuclear reactor, have all been done with WSNs.

Special focus is given to increasing network robustness, battery operating time, data analysis quality and speed, as well as network scalability. Some of these issues can be improved on a software basis by optimizing the WSNs communication protocol. This thesis addresses the more specific topic of sensors which aim to measure and classify a radio wave. Here, utilizing a distributed detection and estimation approach increases energy efficiency, enhances scalability and provides high quality data acquisition and estimation. In this context, this thesis develops two distributed subspace-based algorithms designed to meet the ever-increasing needs for WSNs.

Subspace tracking methods can be classified according to the way their eigenstructure is calculated: The first methods are known as Modified Eigen Problem (MEP) and they are considered to be adaptive, as they actively calculate the update a sample data matrix by means of Eigen Value Decomposition (EVD) or Singular Value Decomposition (SVD) [3, 4, 5]. The second method is based on adaptive subspace estimation [6, 7, 8, 9], also know as non-MEP. Here, adaptive means that the exact eigenstructure of the sample data block is not calculated every time an update occurs, but rather, the algorithm moves towards that result provided by an EVD or SVD diagonalization.

Out of the short list presented above, the Projection Approximation Subspace Tracking Algorithm (PAST) [8] stands out for its low computational complexity and good convergence properties. However, in its original form, it operates in a

Uniform Linear Array (ULA) where all nodes communicate with a centralized base station that analyzes the signals from all nodes in the network. This setup has three main drawbacks: First, the WSN is limited in size, since the nodes farthest from the base station have high transmission costs if not some form of message hopping can be implemented. Second, the base station is mission-critical for the functioning of the system. Third, a centralized WSN performance speed decreases with the size of the network. This can be a problem for safety-critical applications. This thesis explores two distributed versions of the PAST algorithm based on Average Consensus (AC) [10], as an alternative to tackle this problems.

1.1 Contributions

The goal of this thesis is to obtain a distributed version of the PAST algorithm. This allows decentralized WSNs with no base station, where a local network state estimate is calculated in every node and then it is exchanged within its neighborhood such that a global network agreement is reached. That is to say, the aggregate calculations of all nodes converge, and individual outliers play a lesser role. This favors the implementation of small transmission ranges, thus extending battery life, and allows to quickly react to significant state changes. Therefore, a deeper understanding about how the averaging process of local variables influences the time evolution of the signal subspace is necessary. To this aim, we investigate the following points:

- Evaluate the tracking capabilities of the newly developed distributed algorithms: We study how good the Direction Of Arrival (DOA) estimates are when compared to the true values, and investigate the Root Mean Square Error (RMSE) related to these estimates, for several Signal to Noise Ratio (SNR) values.
- Find the Principal Angles between the estimated and the true subspace, as a tool to learn by how many degrees both subspaces are apart from each other.
- Learn how the different network sizes affect the algorithmic performance.
- Decrease transmission distance between neighboring nodes, and thus improve the WSN life-span. We consider a jittered grid topology and explore different types of network connectivity. The purpose is to find a network structure that exchanges as less messages as possible, but still allows for accurate estimates.
- A mathematical proof that shows the convergence properties of our algorithmic approaches.

The following articles have been published by the author of this thesis and are the basis of the present work:

C. Reyes, T. Hilaire, and C. F. Mecklenbräuker. Distributed projection approximation subspace tracking based on consensus propagation. In *The 3rd International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Aruba, December 2009.

C. Reyes, T. Hilaire, S. Paul, and C. F. Mecklenbräuker. Evaluation of the root mean square error performance of the PAST-consensus algorithm. In *International ITG Workshop on Smart Antennas (WSA 2010)*, Germany, February 2010.

M. Rupp and C. Reyes. Robust versions of the PAST algorithm. *Proc. of EUSIPCO*, Bucharest, Rumania. Sep. 2012.

C. Reyes, R. Dallinger, and M. Rupp. Convergence analysis of distributed PAST based on consensus propagation. *Signals, Systems and Computers (ASILOMAR)*, pages 271-275, November 2012.

1.2 Outline

The current document is structured in the following way:

- Chapter 2 presents background information about WSNs, Graph theory, the wave propagation model used in this work and provides a survey on subspace estimation algorithms.
- Chapter 3 proposes two distributed versions of the PAST algorithm. In order to achieve global estimates, certain data needs to be shared among the nodes. Two different scenarios were tested: In the first approach, the signal vector variable $\underline{\mathbf{y}}$ is exchanged using the Average Consensus algorithm. In the second, the sample covariance matrix $\widehat{\mathbf{R}}$ is selected as the variable to be distributed.
- Chapter 4 presents and analyzes the results of the simulation experiments for both algorithmic variants, and compares their performance with that from the original PAST algorithm using identical simulation settings. Since PAST is a recursive algorithm, the influence of the so-called forgetting factor on the performance is analyzed. The Direction Of Arrival, Principal Angles Between Subspaces and the Root Mean Square error are depicted using different topologies and different SNR values.
- Chapter 5 examines the convergence properties of the PAST algorithm.
- Chapter 6 summarizes the most important contributions of this work, provides some final remarks and renders an insight regarding future research.

Chapter 2

Context and Foundations

THIS chapter is intended to offer basic knowledge that will allow a better understanding of this thesis from beginning to end. We start defining the concept of a wireless sensor network and discuss some drawbacks and applications that motivate this work. Section 2.2 and Section 2.3 state the relationship between wireless sensor networks and array processing as well as introduces the narrow band signal model: we regard a wireless sensor network as a space evolving in time, aiming to track only a small dimension of it. Finally, Section 2.4 provides details regarding the Projection Approximation Subspace Tracking (PAST), which is one of the bases of this work.

2.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of a group of nodes which sense some aspects of its environment. Each node has the ability to communicate these observations, traditionally to a centralized destination unit, which is in charge of collecting and processing the data available in the network. A typical example of a WSN would be a number of sensor nodes set up throughout a large facility in order to monitor pollution levels [11].

Sensor nodes are suitable for deployment in harsh environments and over large geographical areas. However, these advantages also impose some limitations [12, 13]. Since individual nodes are usually battery powered, the energy used for operating is a severely limited resource. Furthermore, most of the state-of-the-art sensor network architectures are centralized, where all nodes either report directly or through multihops to a centralized computer. Wireless message transmission in a network is the most power consuming process. Long-range communication with a fusion center

is especially power intensive. In physics, it is well-known that the intensity [1] (p) radiated from a source is proportional to the inverse square of the distance (d), *i.e.,* $p \propto \frac{1}{d^2}$ [14]. In this work, d is interpreted as the distance between the communicating nodes. Therefore, it is of great interest to find alternative solutions to this problem.

A further matter to consider is, that in many applications, aggregate functions of the sensor data are more important than individual node data. In other words, when the wireless sensor network consists of a significant number of sensor nodes, the average value collected by the whole network is more important than the value collected by a single node, as stated in[15, 16]. For some data processing techniques, such as signal compression, detection, classification and localization, the aggregate statistics are even prerequisites.

In our efforts to address the issues above, we concentrate on developing a distributed version of the Projection Approximation Subspace Tracking (PAST) algorithm [8] with a low amount of message passing among sensor nodes. This algorithm offers robust and computationally efficient signal estimation, but is structured in a centralized way. We aim to distribute it for sensor network applications, so that each node can calculate aggregate values with the signals of all sensor nodes within a small transmission range.

2.1.1 Applications

The first WSN was built in the beginning of the 1950s, with the creation of the Sound Surveillance System (SOSUS) [17, 18], developed by the United States military in order to detect and track Soviet submarines by means of hydrophone arrays. Another early application of these networks was environment monitoring in the heavy industry, *i.e.*, power distribution, waste-water treatment and factory automation. In 1980, a Distributed Sensor Network (DSN) project was founded [18, 19] where both academia and industry could participate and investigate the challenges and applications of these networks. Nowadays, sensor networks have wide ranging uses, and some of the most common applications are presented according to their area of operation:

- Military: WSNs are quickly deployed in battle zones, and can, besides creating a communication network for deployed units [20], detect gunfire [21] or chemical agents [22].

- Security and surveillance: the low cost of these networks make them a very attractive alternative to traditionally wired networks, such as burglar alarm systems. Home owners and Industry often prefer easy to install surveillance systems with a sporadic battery change to wired systems with high installation costs [23, 24].

[1] Power per unit area $\left(\frac{W}{m^2}\right)$

- Environmental monitoring: uses include pollution control [2, 11], natural disaster monitoring such as avalanches and landslides [25, 26], the surveillance of wildlife habitats or chemicals in hydrology, as well as the observation of environmental parameters to help in agriculture [27].

- Health: already used for monitoring physiological parameters like S_pO_2 [2], Pulse and Skin Temperature. Some ECGs [3] are taken using WSNs, and they are also used to give real-time information about diabetes and impending asthma attacks [28, 29].

2.1.2 Hardware

A WSN consists of specialized transducers forming a communications infrastructure intended to monitor and analyze conditions at locations where the installation of power and data lines is impractical.

Each individual part making up the WSN is a detection station called a sensor node. Sensor nodes are usually identical, low-cost, light-weight, battery powered autonomous and portable. Due to the large variety of parameters being monitored, sensors are classified into groups:

- Microelectromechanical system (MEMS) incorporate gyroscopes, accelerometers, magnetometers, pressure and acoustic sensors.

- Complementary metal - oxide - semiconductor (CMOS) based sensors measure temperature, humidity and chemical composition.

- Light-emitting diode (LED) sensors are used for ambient light sensing, proximity sensing and to determine chemical composition.

Each sensor node has the capabilities to extract relevant data from its surroundings, process data, and communicate with its neighboring nodes. The basic sensor node has at least four components (see Figure 2.1):

- Sensing Unit: the nodes primary function of gathering information about the physical world is housed in the sensing unit. The sensor collects an analogue signal, which is digitized and passed on to the processing unit.

- Processing Unit: basically a microcomputer, it consists of a microprocessor, memory and I/O units. Its programming controls not only the behavior of the sensor nodes different components, but also the WSN as a whole.

- Communication Unit: provides the wireless interface for the communication between sensor nodes. The most common choice is Ultra Wide Band (UWB)

[2] Oxygen Saturation
[3] Electrocardiograms

radio waves, although WSNs with other communication media, such as optical communication (laser) or infrared exist.

- Power Unit: usually a high energy-density battery pack, it is the sole power source of the sensor node. Since all activity ceases with its depletion, the batteries need to be either changed or charged regularly. A power generator, such as photovoltaic cells, can be used to extend the battery lifetime.

- Location System: some applications need to factor in the position of the individual nodes, so a GPS or an equivalent device can be installed.

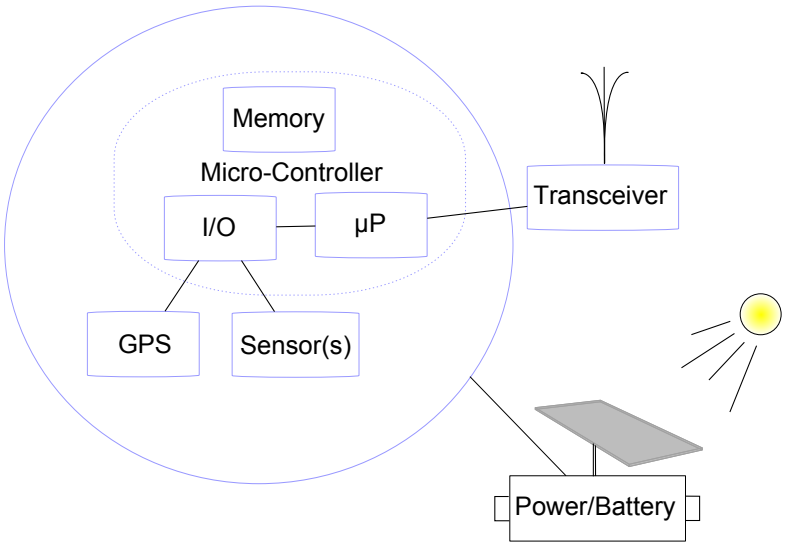

Figure 2.1: Scheme depicting a sensor's architecture.

2.2 Graph theory

The study of networks consisting of interconnections among objects in their most basic form is known as graph theory. In the following section, some basic concepts from the graph theory are introduced, as they will be recurrently used in this thesis.

2.2.1 Foundations

- Graph: let $\mathcal{P}$ be a finite set of N sensor elements indexed by $i \in \{1, 2, \ldots, N\}$, where i is known as vertex. $\mathcal{H} = \{\mathcal{P}, \mathcal{E}\}$ is the sensor network's graph, where $\mathcal{E}$ is a set of elements (i, j), with $i, j \in \mathcal{P}$. A major characteristic of undirected graphs is the symmetry among its connections: An undirected graph exists when $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}$, meaning that sensor i is connected to sensor j and vice versa. This type of graph is addressed throughout this thesis.

- Neighborhood: since this concept is frequently used all over this text, a clear definition is needed: The neighborhood $\mathcal{N}_i$ of any node i is the set of all other nodes connected with i by its edges, *i.e.*, for any node $i \in \mathcal{P}$, there is a set $\mathcal{N}_i \triangleq \{j|(i,j) \in \mathcal{E}\}$ known as the neighborhood of i. We stress the fact throughout this thesis, the node i considers itself as a neighbor. This means that if $\mathcal{N}_i = 9$, it means that sensor i has 8 neighboring nodes plus one, itself.

- Adjacency Matrix: also known as the connectivity matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. It is symmetric only when the graph $\mathcal{H}$ is undirected. It is structured according to which node in the graph $\mathcal{H}$ connects to which other node, namely

$$\mathbf{A}_{i,j} = \begin{cases} 1, & \text{iff } (i,j) \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

- Degree Matrix: is a diagonal matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$, which provides information about the number of edges related to each node i. That is to say

$$\mathbf{D}_{i,j} = \begin{cases} \sum_k \mathbf{A}_{i,k}, & \text{iff } i = j \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

- Laplacian Matrix: it is defined by $\mathbf{L} \triangleq \mathbf{D} - \mathbf{A}$. As the Laplacian merges both, the information about the network connectivity from (2.1) and the exact network node degree (2.2), it becomes evident that this matrix provides the information necessary to represent the graph $\mathcal{H}$. Let us clarify the meaning of the Laplacian by calculating the following example: Consider an undirected graph associated with a wireless sensor network with $N = 4$ elements and $i \in \{1, \ldots, N\}$. As depicted in Figure 2.2, the sensor $i = 1$ communicates with its neighborhood $\mathcal{N}_1 = \{2, 4\}$. Those interconnected elements in the first row of the adjacency matrix $\mathbf{A}_{i,N}$ are set to 1. The non-neighbors are set to 0. The degree matrix in a wireless sensor network is seen as the diagonal matrix whose $\mathbf{D}_{i,i}$ contains the total number of connections related to sensor i. Following the example and still considering $i = 1$ with its two connected neighbors, the node degree $\mathbf{D}_{1,1} = 2$.

$$\underbrace{\begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 1 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}}_{\mathbf{L}} = \underbrace{\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}}_{\mathbf{D}} - \underbrace{\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}}_{\mathbf{A}}$$

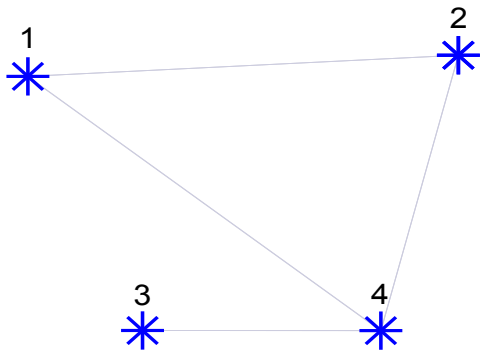

Figure 2.2: Undirected graph with $N = 4$ sensor elements

As the adjacency matrix is symmetric and the degree of each node is placed as a diagonal matrix, the Laplacian of an undirected graph is by construction always symmetric.

2.2.2 Network Topology

The Figures 2.3 and 2.4 depict various graph topologies with $N = 12$ and $N = 36$ elements that work in a centralized way. At Figure a.) a fully connected network, also known as full mesh topology, is presented. All elements of the network connect directly with each other and with a central processing unit (marked in red). This kind of topology is very expensive, since there are many redundant connections. A beneficial side effect of the multiple message transmissions is a high degree of reliability. Figures b.) and c.) correspond to a dense and a sparse representation of a partial mesh topology. Here, the sensors in the system interact only with a few other nodes. Figure d.) and e.) show networks where each node i has a regular neighborhood size $\mathcal{N}_i = 5$ or $\mathcal{N}_i = 9$. Figure e.) is only used on the topology with $N = 36$ elements.

Besides the energy efficiency issues, we are also interested in studying the impact of the network topology on the data estimation performance. In addition, we question how the topology impacts the tracking capabilities and the convergence properties of the distributed algorithmic variants proposed in Chapter 3.

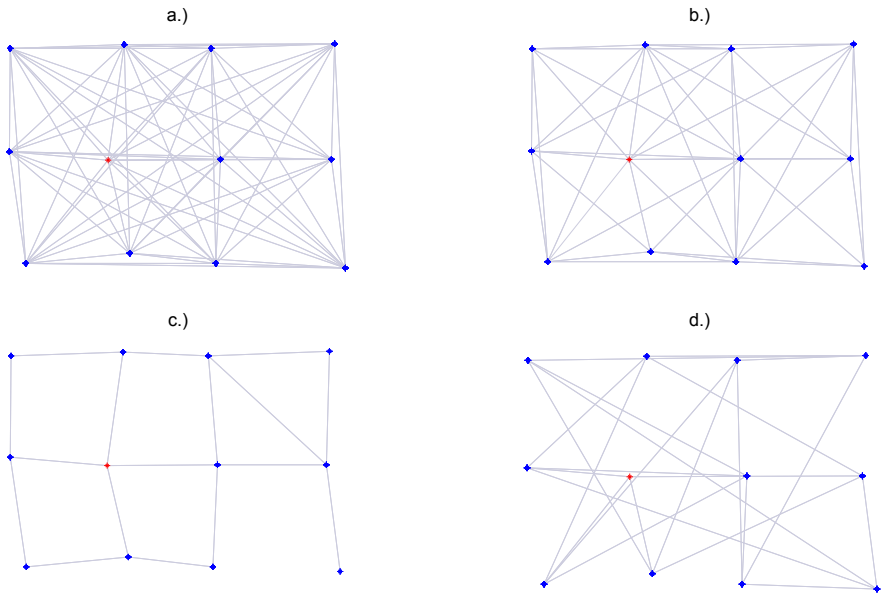

Figure 2.3: Undirected graphs with $N = 12$. The following cases are depicted: a.) Fully connected scenario, b.) Dense connected, c.) Sparse connected and d.) regular topology $\mathcal{N}_i = 5$

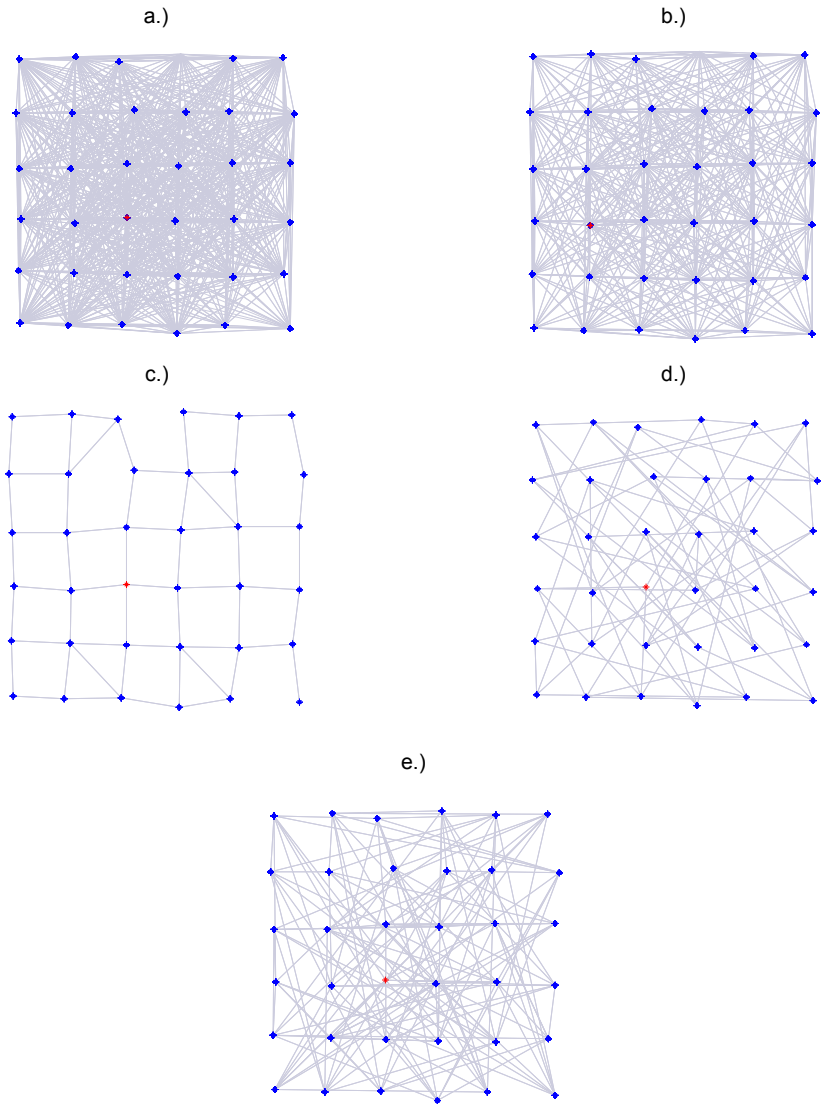

Figure 2.4: Undirected graphs with $N = 36$. The following cases are depicted: a.) Fully connected scenario, b.) Dense connected, c.) Sparse connected, d.) regular topology $\mathcal{N}_i = 5$ and e.) regular topology $\mathcal{N}_i = 9$.

2.3 Wave propagation

A propagating field originated by a *mechanical* [4] or *electromagnetic* wave [5] is often evaluated by means of a sensor array. These arrays consist of transducers (sensors) deployed in a specific geometric configuration, capable of detecting and converting the waves into an electric signal. Array signal processing applications encompass radar, sonar, seismic event prediction and wireless communication systems [30].

Several methods developed for estimating the origin of a source are based on calculating the signal's direction of arrival (DOA). This concept is introduced in Subsection (2.3.1), where also the relationship among the antenna array elements used to create a spatial representation of the signals impinging the sensors is regarded. In addition to this, Subsection (2.3.2) introduces a model that approximates the individual sensor response to an incoming wave.

2.3.1 Space-time representation

Propagating waves are a function of both space and time. In physics, they are expressed as the solutions to the wave equation for the medium of interest [31, 32], which derive from Maxwell's equation.

$$\nabla^2 \vec{E} = \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2}. \tag{2.3}$$

Here, $\vec{E}$ can be an electric field in electromagnetics or acoustic pressure in acoustic waves, $c = 3 \times 10^8$ m/s is the propagation speed in the medium, and ∇^2 is the Laplacian operator

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}. \tag{2.4}$$

Let us consider a general scalar field $s(t, \vec{p})$ occurring at time t and position vector $\vec{p} = [x, y, z]^T$. Hence, the wave equation in (2.4) turns into

$$\frac{\partial^2 s(t,\vec{p})}{\partial x^2} + \frac{\partial^2 s(t,\vec{p})}{\partial y^2} + \frac{\partial^2 s(t,\vec{p})}{\partial z^2} = \frac{1}{c^2} \frac{\partial^2 s(t,\vec{p})}{\partial t^2}. \tag{2.5}$$

The solution for (2.5) is often represented in a complex exponential form, *i.e.*;

$$s(t,\vec{p}) = s(t,x,y,z) = A \exp\{j(\omega t - \kappa_x x - \kappa_y y - \kappa_z z)\} \tag{2.6}$$

$$= A \exp\{j(\omega t - \vec{\kappa}^T \cdot \vec{p})\}, \tag{2.7}$$

where (2.7) is a vectorial and inner product representation. Here, the temporal fre-

[4] Seismic waves: tsunamis or earthquakes. Acoustic waves: vibrating strings of an instrument, vibrating tines of a tuning fork, a microphone or a sonar array.

[5] Wireless communications.

quency is given as $\omega = 2\pi f$ in units of radians per seconds and $\vec{\kappa} = [\kappa_x, \kappa_y, \kappa_z]^T$ is the wave number vector in units of radians per meter, also known as the spatial frequency of the mono-chromatic plane wave. A mono-chromatic wave has a single frequency, constant amplitude and phase. Recall that the frequency content of a signal is defined by its Fourier transform: if the amplitude or phase is time dependent, the Fourier transform will be nonzero at several frequencies. Hence, the signal is not monochromatic. The scalar field in (2.7) may be characterized in terms of plane waves of different frequencies [32] as

$$s(t, \vec{p}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp\{j \left(\omega t - \vec{\kappa}^T \cdot \vec{p}\right)\} d\omega, \tag{2.8}$$

which is basically the space-time representation of the propagation waves, where $S(\omega)$ is the Fourier transform of $s(\cdot)$. Substituting (2.6) into (2.5) yields

$$||\vec{\kappa}||^2 = \kappa_x^2 + \kappa_y^2 + \kappa_z^2 = \frac{\omega^2}{c^2}, \tag{2.9}$$

an expression relating the temporal and the spatial frequency. If this holds true, the exponential in (2.6) is a solution to the wave equation. By replacing $c = \lambda f$ in (2.9), the magnitude of the wave number vector leads to $|\vec{\kappa}| = \frac{2\pi}{\lambda}$. Here, λ represents the wavelength defined as the distance propagated during one temporal period T. Another space-time representation of the basic propagating wave in (2.7) has the form

$$s(t - \vec{\xi}^T \cdot \vec{p}) = A \exp\{j\omega(t - \vec{\xi}^T \cdot \vec{p})\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp\{j\omega(t - \vec{\xi}^T \cdot \vec{p})\} d\omega, \tag{2.10}$$

where the vector $\vec{\xi} = \frac{\vec{\kappa}}{\omega}$, known in the literature as the slowness vector; points in the same direction as $\vec{\kappa}$ and has magnitude $|\vec{\xi}| = \frac{1}{c}$. Let the sensor array in Figure (2.5) be placed close to the coordinate system, such that $\vec{\xi}$ is expressed in terms of its spatial coordinates as

$$\vec{\xi} = -\frac{1}{c} \begin{pmatrix} \sin\phi \cos\theta \\ \sin\phi \sin\theta \\ \cos\phi \end{pmatrix}, \tag{2.11}$$

where θ is the azimuth angle and ϕ stands for the angle of elevation [33]. These angles are known as the Direction of Arrival (DOA) and represent the direction of the propagating wave. In general, they are time dependent, but when the *far-field assumption* is taken into account, the angles ϕ and θ are considered to vary so slowly, that they become nearly constant.

Far-field assumption: For source signals whose propagation distance to a sensor array is significantly larger than the aperture of the array itself, the DOA value at

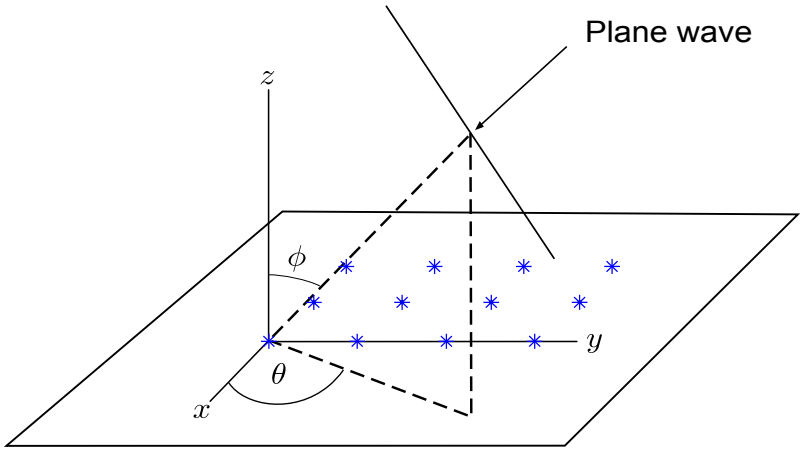

Figure 2.5: Plane wave impinging on a sensor array with $N = 12$ elements.

each sensor will be almost the same. According to (2.7), the wave front of constant phase at time instant t is a plane perpendicular to the propagating direction given by $\vec{\kappa} \cdot \vec{p} =$ constant. This is also known as plane wave.

2.3.2 System model

Let a planar array of sensors at the position $\vec{p}_i$ for all $i = 1, \ldots, N$ observe some signals originated by r sources. Allow the first sensor to be located at the origin of the coordinate system and let $s_l(t)$, for all $l = 1, \ldots, r$ describe the signal wave impinging on the first node. The observation at node i results from the sum of a time delayed version of the original signals hidden in additive noise $n_i(t)$, which is modeled as

$$x_i(t) = \sum_{l=1}^{r} s_l(t - \Delta_{i,l}) + n_i(t), \tag{2.12}$$

where $\Delta_{i,l}$ represents the time propagation delay from the signal source to the i sensor position. The delay is defined by $\Delta_{i,l} = \vec{\xi}_l \cdot \vec{p}_i$ and is associated with the DOA through $\vec{\xi}_l = \frac{\vec{\kappa}_l}{\omega}$, as shown in (2.11). Applying the Fourier transform to the total data array output $\underline{\mathbf{x}}(t) = [x_1(t), \ldots, x_N(t)]^T$, the time delay commutes into phase shift $\exp\{-j\omega\Delta_{il}\}$ and the array output in the Fourier domain is denoted as

$$\underline{\mathbf{x}}(\omega) = \begin{pmatrix} x_1(\omega) \\ \vdots \\ x_N(\omega) \end{pmatrix} = \sum_{l=1}^{r} \underline{\boldsymbol{v}}(\phi_l, \theta_l) s_l(\omega) + \underline{\mathbf{n}}(\omega) = \boldsymbol{\Upsilon}(\omega, \boldsymbol{\phi}, \boldsymbol{\theta}) \underline{\mathbf{s}}(\omega) + \underline{\mathbf{n}}(\omega), \tag{2.13}$$

where $\underline{\boldsymbol{\upsilon}}(\phi_l, \theta_l)) = [\exp\{-j\vec{\kappa}_l \cdot \vec{p}_1\}, \ldots, \exp\{-j\vec{\kappa}_l \cdot \vec{p}_N\}]^T$ is the array manifold (steering vector) depending on the l^{th} incoming wave. The steering matrix is defined as $\boldsymbol{\Upsilon}(\omega, \boldsymbol{\phi}, \boldsymbol{\theta}) = [\underline{\boldsymbol{\upsilon}}(\omega, \phi_l, \theta_l), \ldots, \underline{\boldsymbol{\upsilon}}(\omega\phi_l, \theta_l)]$. The expression (2.13) is widely used when considering broadband signals. In the following, the system model for narrow band signals is derived.

2.3.2.1 Narrow band signals

In wireless communications, the source signals are modulated before being transmitted by the carrier frequency $\omega_c = 2\pi f_c$. At the receiver side, the signals are demodulated such that they again become baseband. The narrow band assumption exists if $B_s \Delta T << 1$. Namely, if the signal $\underline{\mathbf{s}}(t)$ is band limited with bandwidth B_s and the ΔT denotes the maximal travel time between two sensors of the array, the complex demodulated signal wave is practically the same across the array elements.

Redefining (2.13) in terms of narrow band signals, the observed data vector yields

$$\underline{\mathbf{x}}(t) = \boldsymbol{\Upsilon}(\boldsymbol{\phi}, \boldsymbol{\theta})\underline{\mathbf{s}}(t) + \underline{\mathbf{n}}(t). \tag{2.14}$$

Here the signal mixing matrix $\boldsymbol{\Upsilon}(\boldsymbol{\phi}, \boldsymbol{\theta}) \in \mathbb{C}^{N \times r}$ is calculated at the carrier frequency ω_c. The incident signal vector $\underline{\mathbf{s}}(t)$ and noise vector $\underline{\mathbf{n}}(t)$ are assumed to be realizations of a white Gaussian [6] random process (zero-mean and variance σ_N^2), with zero correlation between the different signals and between the noise and the signals. Since the important information is centered at ω_c, the frequency dependence in (2.13) is neglected. Moreover, as the DOA estimation is generally based on samples of (2.14), we allow the index k to denote the discrete samples of $\underline{\mathbf{x}}(t)$. That is to say,

$$\underline{\mathbf{x}}(k) = \boldsymbol{\Upsilon}(\boldsymbol{\phi}, \boldsymbol{\theta})\underline{\mathbf{s}}(k) + \underline{\mathbf{n}}(k). \tag{2.15}$$

In this thesis, a two dimensional sensor array is considered, meaning that the elevation component $\boldsymbol{\phi}$ in Figure 2.5 is neglected. As a result, (2.11) becomes

$$\vec{\xi} = -\frac{1}{c}\begin{pmatrix}\cos\boldsymbol{\theta} \\ \sin\boldsymbol{\theta}\end{pmatrix}. \tag{2.16}$$

[6] Noise is a very undesirable component in electric circuits, as it generally negatively impacts the quality of any transmitted message. One type of noise that is inherent to all electronic devices is the thermal noise, which refers to the random variations in current or voltage caused by the random movement of electrons. The definition of white noise is regularly explained as a random signal whose power spectral density is uniformly distributed in the entire frequency domain, meaning that the power of the signal stays the same in the whole frequency domain with a regular bandwidth.

Rewriting (2.15) only in terms of the angle θ in (2.16) yields

$$\begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{pmatrix} = \begin{pmatrix} 1 & \dots & 1 \\ e^{-j\frac{\omega_c}{c}(x_1\cos\theta_1 + y_1\sin\theta_1)} & \dots & e^{-j\frac{\omega_c}{c}(x_1\cos\theta_r + y_1\sin\theta_r)} \\ \vdots & \vdots & \vdots \\ e^{-j\frac{\omega_c}{c}(x_N\cos\theta_1 + y_N\sin\theta_1)} & \dots & e^{-j\frac{\omega_c}{c}(x_N\cos\theta_r + y_N\sin\theta_r)} \end{pmatrix} \times \begin{pmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_r(k) \end{pmatrix} + \begin{pmatrix} n_1(k) \\ n_2(k) \\ \vdots \\ n_N(k) \end{pmatrix}. \tag{2.17}$$

This is the same as

$$\begin{pmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{pmatrix} = \Big(\underline{\boldsymbol{\upsilon}}(\theta_1), \dots, \underline{\boldsymbol{\upsilon}}(\theta_r)\Big) \begin{pmatrix} s_1(k) \\ s_2(k) \\ \vdots \\ s_r(k) \end{pmatrix} \begin{pmatrix} n_1(k) \\ n_2(k) \\ \vdots \\ n_N(k) \end{pmatrix}, \tag{2.18}$$

where each $\underline{\boldsymbol{\upsilon}}(\theta_l)$ is the array steering vector corresponding to the direction of arrival of the l^{th} signal. Finally, the narrow band signal model to be used is given as

$$\underline{\mathbf{x}}(k) = \boldsymbol{\Upsilon}(\boldsymbol{\theta})\underline{\mathbf{s}}(k) + \underline{\mathbf{n}}(k). \tag{2.19}$$

In (2.19), the received vector $\underline{\mathbf{x}}(k)$ and the steering vectors $\upsilon(\theta_l)$ can be geometrically visualized as vectors lying in the N dimensional space. It is evident that the observation vector is a linear combination of the array steering matrix $\boldsymbol{\Upsilon}(\boldsymbol{\theta})$ together with the signal vector $\underline{\mathbf{s}}(k)$ acting as the coefficients of the combination. Since the first sensor is located at the origin, $(x_1 = y_1 = 0)$ the first row of the signal mixing matrix $\boldsymbol{\Upsilon}(\boldsymbol{\theta})$ becomes one. One of the main goals throughout this thesis is to track the time evolution of this array response matrix, as it is the signal subspace containing the unknown spatial frequencies in terms of the DOA shown in (2.16).

2.4 Subspace methods

Subspace methods are highly useful in the realm of modern signal processing. Some of its applications encompass blind communication channel identification [34], online identification of network anomalies [35, 36], beamforming [37], denoising [38], estimation of direction of arrivals and image compression [23], to mention some.

Many of the first techniques rely on subspace or eigen-based information found by Eigenvalue Decomposition (EVD) of an estimated correlation matrix or Singular Value Decomposition (SVD) of the corresponding data matrix. The main drawback of these methods is that they are not computationally efficient to update when new

data arrives, and therefore are process intensive. This limits their usefulness for real-time and power constrained applications.

Several well-known algorithms have emerged to address the computational complexity issue by various means [8, 37, 39, 7, 40, 9], and two of those are used in this dissertation: The MUltiple SIgnal Classification (MUSIC) [40], and Yang's Projection Approximation Subspace Tracking (PAST) algorithm [8].

The extraction of unknown sources from a set of given signals as in (2.19) is relevant for several engineering applications. Usually, signals are estimated when an incoming wave is detected by two or more array sensors. The higher the number of sensors, the easier it is to estimate the signal. This scenario is known as an underdetermined system constrained to $r < N$. Several methods also address the problem of overdetermined systems, since $r \gg N$ is a plausible scenario for engineering applications [41, 42]. Overall, this thesis studies underdetermined systems where the number of sensors N is much larger than the number of source locations r.

2.4.1 True array covariance matrix

Subspace tracking relies on updating a low dimensional signal subspace instead of the whole space spanned by the covariance matrix. The spatial covariance matrix $\mathbf{R}^{xx}$ contains the noisy measurements $\{\underline{\mathbf{x}}(k), k = 1, \ldots, K\}$ from (2.19) across the sensor array, and it is defined as

$$\mathbf{R}^{xx} = E\{\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\} = \boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H + \sigma^2\mathbf{I} \quad \in \mathbb{C}^{N\times N}, \tag{2.20}$$

where $E\{\cdot\}$ denotes the statistical expectation operator and the covariance matrix of the signal sources $\underline{\mathbf{s}}(k)$ is given by $\mathbf{R}^{ss} = E\{\underline{\mathbf{s}}(k)\underline{\mathbf{s}}^H(k)\} \in \mathbb{C}^{r\times r}$. The eigenvalue decomposition of (2.20) yields

$$\mathbf{R}^{xx} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H = \begin{pmatrix}\mathbf{U}_r & \mathbf{U}_o\end{pmatrix}\left[\begin{pmatrix}\boldsymbol{\Lambda}_r & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Lambda}_o\end{pmatrix} + \sigma_N^2\mathbf{I}_N\right]\begin{pmatrix}\mathbf{U}_r^H\\ \mathbf{U}_o^H\end{pmatrix}, \tag{2.21}$$

where $\boldsymbol{\Lambda}_r = diag\,(\lambda_1, \ldots, \lambda_r)$ correspond to the first r dominant eigenvalues and $\boldsymbol{\Lambda}_o = diag\,(\lambda_{r+1}, \ldots, \lambda_N = \sigma^2)$ to the noise eigenvalues. The column spans of the signal eigenvectors, *i.e.* $\mathbf{U}_r = [\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_r]$ is interpreted as the signal subspace and the noise eigenvectors spanning $\mathbf{U}_o = \left[\underline{\mathbf{u}}_{r+1}, \ldots, \underline{\mathbf{u}}_N\right]$ are characterized as the noise subspace.

2.4.2 MUltiple SIgnal Classification (MUSIC)

The eigenvalues of $\mathbf{R}^{xx}$ given by $\{\lambda_1, \ldots, \lambda_N\}$ are built in such way that the determinant

$$|\mathbf{R}^{xx} - \lambda_i\mathbf{I}| = \mathbf{0}. \tag{2.22}$$

Substituting (2.20) in (2.22) leads to

$$|\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H + \sigma_N^2\mathbf{I} - \lambda_i\mathbf{I}| = |\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H - (\lambda_i - \sigma_N^2)\mathbf{I}| = \mathbf{0}, \tag{2.23}$$

where the eigenvalues η_i of the term $\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H$ are given by

$$\eta_i = \lambda_i - \sigma_N^2. \tag{2.24}$$

As $\boldsymbol{\Upsilon}$ is composed of linearly independent vectors, it has full column rank. Moreover, the signal covariance matrix $\mathbf{R}^{ss}$ is non-singular as the impinging waves are assumed to be low correlated. These two facts are important as they ensure that, when the number of incident signals r is smaller than the size of the sensor array, the matrix $\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H$ is positive semi definite with rank r. This suggests that there exist $N - r$ eigenvalues η_i of $\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H$ equal to zero. Going back to (2.24), it is observable that $N - r$ of the eigenvalues of $\mathbf{R}^{xx}$ are equal to the noise variance σ_N^2. Hence, the eigenvalues corresponding to $\mathbf{R}^{xx}$ may be sorted from the largest to the smallest, as shown in (2.21). Now we search for the eigenvector $\underline{\mathbf{u}}_i$ corresponding to λ_i, such that

$$(\mathbf{R}^{xx} - \lambda_i\mathbf{I})\,\underline{\mathbf{u}}_i = \mathbf{0}. \tag{2.25}$$

Evaluating the eigenvectors associated to the less dominant eigenvalues yields

$$(\mathbf{R}^{xx} - \sigma_N^2\mathbf{I})\underline{\mathbf{u}}_i = \boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H\underline{\mathbf{u}}_i + \sigma_N^2\mathbf{I}\underline{\mathbf{u}}_i - \sigma_N^2\mathbf{I}\underline{\mathbf{u}}_i = \mathbf{0}, \tag{2.26}$$

and further simplification results in

$$\boldsymbol{\Upsilon}\mathbf{R}^{ss}\boldsymbol{\Upsilon}^H\underline{\mathbf{u}}_i = \mathbf{0}. \tag{2.27}$$

As $\boldsymbol{\Upsilon}$ has full rank and $\mathbf{R}^{ss}$ is non-singular, this implies that $\boldsymbol{\Upsilon}^H\underline{\mathbf{u}}_i = \mathbf{0}$. Namely,

$$\begin{pmatrix} \underline{\boldsymbol{\upsilon}}(\theta_1)\underline{\mathbf{u}}_i \\ \vdots \\ \underline{\boldsymbol{\upsilon}}(\theta_r)\underline{\mathbf{u}}_i \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2.28}$$

This suggests that the eigenvectors corresponding to the $N - r$ smallest eigenvalues, known as the noise eigenvectors $\mathbf{U}_o$, are orthogonal to the r steering vectors that span $\boldsymbol{\Upsilon}$. Thus, it becomes obvious that the eigenvectors associated to the r largest eigenvalues, the signal eigenvectors $\mathbf{U}_r$, span the same subspace as the steering matrix $\boldsymbol{\Upsilon}$. This is to say, $\mathbf{Span}(\mathbf{U}_o) \perp \mathbf{Span}(\boldsymbol{\Upsilon})$ and $\mathbf{Span}(\mathbf{U}_r) = \mathbf{Span}(\boldsymbol{\Upsilon})$.

The MUSIC algorithm [40] takes the aforementioned ideas and states that the steering vectors associated with the incident signals may be estimated by recognizing the steering vectors which are as orthogonal as possible to those eigenvectors

associated with the eigenvalues from the sample covariance matrix $\mathbf{R}^{xx}$.

Nevertheless, this algorithm does not directly provide the DOA of the incident signals. In order to calculate them, it is necessary to calculate the MUSIC spatial spectrum on the extent of the parameters space (all possible DOA), i.e.;

$$P = \frac{\underline{\boldsymbol{v}}^H(\theta)\underline{\boldsymbol{v}}(\theta)}{\underline{\boldsymbol{v}}^H(\theta)\mathbf{U}_o\mathbf{U}_o^H\underline{\boldsymbol{v}}(\theta)}. \tag{2.29}$$

Where the DOA of the signals impinging on a node is estimated by locating the peaks in the spectrum, as showh in Figure 2.6 Here, the largest peaks show that

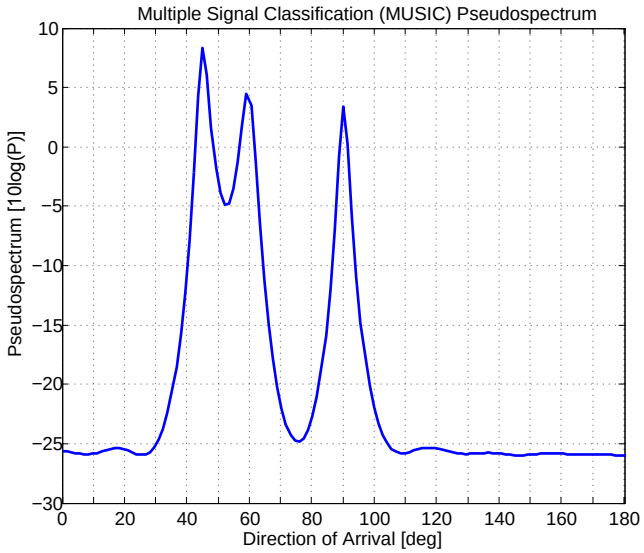

Figure 2.6: Scheme depicting a sensor's architecture.

there are $r = 3$ signals of interest and the estimated DOA are given by the highest points.

Some advantages and drawbacks of the MUSIC algorithm are:

- It works for several array shapes, but knowledge of the sensor positions is needed [43].
- Very sensitive to sensor position, gain, and phase errors. Careful calibration is necessary to make it work well.
- Searching through all θ may limit the performance in terms of speed and computational resources.

Several variants of MUSIC [44, 45] have been proposed to reduce complexity, increase performance and resolution power. Nevertheless, we leave these issues out of the scope of this thesis.

2.4.3 Estimation of the array covariance matrix

The algorithms based on subspace estimation follow the time evolution of the true data covariance matrix $\mathbf{R}^{xx}$. In order to calculate this matrix, we need to have access to the whole random process $\underline{\mathbf{x}}(k)$, which is impossible due to its infinite length. Nevertheless, if we let the process to be ergodic[7] the ensemble averaging covariance matrix $\mathbf{R}^{xx}$ may be approximated by a time average $\widetilde{\mathbf{R}}^{xx}(K)$ over past samples. Additionally, the ergodic process has to be stationary. By supposing $\underline{\mathbf{x}}(k)$ to vary slowly within the effective observation window length, we may assume it as nearly stationary.

A well known-method for estimating the data correlation matrix is given by

$$\acute{\mathbf{R}}^{xx}(K) = \frac{1}{K}\sum_{k=1}^{K}\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^{H}(k) = \frac{1}{K}\mathbf{X}\mathbf{X}^{H} \approx \mathbf{R}^{xx} \qquad \in \mathbb{C}^{N\times N}, \tag{2.30}$$

where $\mathbf{X}$ denotes the noisy data matrix composed of K snapshots $\underline{\mathbf{x}}_k$ in the columns and N sensor elements in the rows, *i.e.,*

$$\begin{aligned}\mathbf{X} &= \begin{pmatrix}\underline{\mathbf{x}}(1) & \underline{\mathbf{x}}(2) & \dots & \underline{\mathbf{x}}(K)\end{pmatrix} \\ &= \boldsymbol{\Upsilon}(\boldsymbol{\theta})\begin{pmatrix}\underline{\mathbf{s}}(1) & \underline{\mathbf{s}}(2) & \dots & \underline{\mathbf{s}}(K)\end{pmatrix} + \begin{pmatrix}\underline{\mathbf{v}}(1) & \underline{\mathbf{v}}(2) & \dots & \underline{\mathbf{v}}(K)\end{pmatrix} \\ &= \boldsymbol{\Upsilon}(\boldsymbol{\theta})\mathbf{S} + \mathbf{V} \qquad \in \mathbb{C}^{N\times K}.\end{aligned} \tag{2.31}$$

The time averaged sample correlation matrix in (2.30) and (2.32) are an estimator for the true covariance matrix in (2.20), and it converges with probability one to $\mathbf{R}^{xx}$ as the sample size increases. In a practical scenario, the received signals at the sensor array do change their statistical information over time. Hence, updating the estimate of the covariance matrix periodically such that older samples are downdated, leads to another way of calculating this matrix, namely

$$\widehat{\mathbf{R}}^{xx}(k) = \beta\widehat{\mathbf{R}}^{xx}(k-1) + (1-\beta)\,\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^{H}(k) \approx \mathbf{R}^{xx} \qquad \in \mathbb{C}^{N\times N}. \tag{2.32}$$

This is known as the exponentially weighted covariance matrix, where β is the forgetting factor with values $0 < \beta \leq 1$. The smaller the selected β value is, the smaller the contribution of previous samples.

[7] Each realization eventually acquires the same statistical properties (mean and variance) while they are sufficiently long in duration.

2.4.4 Projection Approximation Subspace Tracking (PAST)

Among the usual approaches for adaptive subspace-based methods, we find the PAST algorithm [8], which does not require a sample covariance estimate every time a new data sample arrives. It estimates the signal subspace at time k recursively, depending on the previous subspace estimate at time $k-1$ and the new observation $x(k)$.

Let $\underline{\mathbf{x}}$ be the observation vector of a random process with correlation matrix $\mathbf{R}^{xx}$. The respective eigenvalues and eigenvectors are given by (2.21), with $\mathbf{U}^H\mathbf{U} = \mathbf{I}$. In [8, 46, 47, 48], Yang defines

$$\mathcal{S}\{b_1, \ldots, b_r\} = \{\mathbf{W} \in \mathbb{R}^{N\times r} |\ \mathbf{W} = \left[\underline{\mathbf{u}}_{b_1}, \ldots, \underline{\mathbf{u}}_{b_r}\right] \mathbf{Q},\ \mathbf{Q}^H\mathbf{Q} = \mathbf{I}\}, \tag{2.33}$$

as a subset of the sample covariance space $(1, \ldots, N)$, where $\mathbf{Q}$ is a unitary matrix. The expression (2.33) represents the set containing all possible combinations of the $N \times r$ matrices, whose columns span an orthonormal basis of the subspace generated by the eigenvectors $\left[\underline{\mathbf{u}}_{b_1}, \ldots, \underline{\mathbf{u}}_{b_r}\right]$. The true signal subspace is described by $\mathcal{S}\{1, \ldots, r\}$, with eigenvectors $[\underline{\mathbf{u}}_1, \ldots, \underline{\mathbf{u}}_r]$. Now, the target is to find the subspace $\mathbf{W}$, subject to $\mathbf{Span}(\mathbf{W}) \approx \mathbf{Span}(\mathbf{U}_r)$. Consider the following cost function

$$\mathcal{J}(\mathbf{W}) = E\{||\underline{\mathbf{x}}(k) - \mathbf{W}\mathbf{W}^H\underline{\mathbf{x}}(k)||\}^2. \tag{2.34}$$

When $\mathbf{W} \in \mathbb{R}^{N\times r}$ has full rank r, with $N > r$, two theorems are derived [48]:

- *Theorem 1: each stationary point of $J(\mathbf{W})$ is characterized by $\mathbf{W} \in \mathcal{S}(b_1, \ldots, b_r)$. To be more specific, $\frac{d\mathcal{J}(\mathbf{W})}{d\mathbf{W}} = 0$ implies $\mathbf{W}^H\mathbf{W} = \mathbf{I}$ and $(\mathbf{I} - \mathbf{W}\mathbf{W}^H)\mathbf{R}^{xx}\mathbf{W} = 0$. The last equation is satisfied if and only if $\mathbf{W} \in \mathcal{S}\{b_1, \ldots, b_r\}$, where $\{b_1, \ldots, b_r\}$ is an arbitrary subset of $\{1, \ldots, N\}$.*
- *Theorem 2: All stationary points not belonging to $\mathcal{S}(1, \ldots, r)$ are saddle points. If $\lambda_1 \geq \ldots \geq \lambda_r > \lambda_{r+1} \geq \ldots \geq \lambda_N$, $\mathcal{J}(\mathbf{W}) > \mathcal{J}(\mathbf{W}_*) = \sum_{i=r+1}^{N} \lambda_i$ for $\mathbf{W} \notin \mathcal{S}(1, \ldots, r)$ are saddle points.*

The later theorem proposes that when $\mathcal{J}(\mathbf{W})$ is minimized, it reaches the global minimum when the columns of $\mathbf{W}$ are an orthonormal basis of the signal subspace. Naturally, as in real life scenarios there are only $\underline{\mathbf{x}}(k)$ samples available, Yang [8] approximates the cost function to minimize

$$\mathcal{J}(\mathbf{W}(k)) = \sum_{l=1}^{k} \beta^{k-l} \left\|\underline{\mathbf{x}}(l) - \mathbf{W}(k)\mathbf{W}^H(k)\underline{\mathbf{x}}(l)\right\|^2 \tag{2.35}$$

by

$$\mathcal{J}'(\mathbf{W}(k)) = \sum_{l=1}^{k} \beta^{k-l} \left\|\underline{\mathbf{x}}(l) - \mathbf{W}(k)\underline{\mathbf{y}}(l)\right\|^2, \tag{2.36}$$

with

$$\underline{\mathbf{y}}(l) = \mathbf{W}^H(l-1)\underline{\mathbf{x}}(l) \ . \tag{2.37}$$

where $\underline{\mathbf{y}}(l)$ constitutes the approximation vector originated by projecting the vector $\underline{\mathbf{x}}(l)$ onto the column space of $\mathbf{W}(k)$ at each k instant. Consequently, all the sample vectors accessible in the time interval $1 \leq l \leq k$ are entangled in estimating the signal subspace at the current time k. The modified cost function from equation (2.36) is minimized if

$$\mathbf{W}(k) = \widehat{\mathbf{R}}_k^{xy} \left(\widehat{\mathbf{R}}_k^{yy}\right)^{-1}, \tag{2.38}$$

where

$$\widehat{\mathbf{R}}_k^{xy} = \sum_{l=1}^{k} \beta^{k-l} \underline{\mathbf{x}}(l)\underline{\mathbf{y}}^H(l) = \beta \widehat{\mathbf{R}}_{k-1}^{xy} + \underline{\mathbf{x}}(k)\underline{\mathbf{y}}^H(k) \tag{2.39}$$

and

$$\widehat{\mathbf{R}}_k^{yy} = \sum_{l=1}^{k} \beta^{k-l} \underline{\mathbf{y}}(l)\underline{\mathbf{y}}^H(l) = \beta \widehat{\mathbf{R}}_{k-1}^{yy} + \underline{\mathbf{y}}(k)\underline{\mathbf{y}}^H(k). \tag{2.40}$$

We use the Matrix Inversion Lemma to solve (2.38), and arrive at a Recursive Least Square (RLS) algorithm-based solution [49]. The computational steps shown in Algorithm 1 are derived for tracking the signal subspace matrix $\mathbf{W}(k)$.

Algorithm 1: PAST algorithm by Yang [8]

Input: β, $\mathbf{R}(0)$, $\mathbf{W}(0)$
for $k := 1, 2, ...$ **do**
  Input: $\underline{\mathbf{x}}(k)$
  $\underline{\mathbf{y}}(k) = \mathbf{W}_{k-1}^H \underline{\mathbf{x}}(k)$
  $\underline{\mathbf{h}}(k) = \widehat{\mathbf{R}}_{k-1}^{yy} \underline{\mathbf{y}}(k)$
  $\underline{\mathbf{g}}(k) = \underline{\mathbf{h}}(k) / [\beta + \underline{\mathbf{y}}^H(k)\underline{\mathbf{h}}(k)]$
  $\widehat{\mathbf{R}}^{yy}(k) = \frac{1}{\beta}[\widehat{\mathbf{R}}_{k-1}^{yy} - \underline{\mathbf{g}}(k)\underline{\mathbf{h}}^H(k)]$
  $\underline{\mathbf{e}}(k) = \underline{\mathbf{x}}(k) - \mathbf{W}_{k-1}\underline{\mathbf{y}}(k)$
  $\mathbf{W}(k) = \mathbf{W}_{k-1} + \underline{\mathbf{e}}(k)\underline{\mathbf{g}}^H(k)$
  Output: $\mathbf{W}(k)$
end for

In the first step, $\underline{\mathbf{y}}(k)$ stores the modified data vector resulting from the multiplication between the old signal subspace $\mathbf{W}^H(k-1)$ and the new data vector $\underline{\mathbf{x}}(k)$ observed at the sensor array. Note that $\underline{\mathbf{h}}(k)$ and the gain vector $\underline{\mathbf{g}}(k)$ are intermediate steps of the algorithm while $\underline{\mathbf{e}}(k)$ is the estimation error vector. A tracking algorithm estimates the matrix $\mathbf{W}(k)$ as a function of the previously calculated $\mathbf{W}(k-1)$ and the new observation $\underline{\mathbf{x}}(k)$ alone, as shown in the last step.

Chapter 3

Distributed subspace tracking

IN this chapter we provide a short overview of the state-of-the-art subspace tracking algorithms that work in a decentralized way. In addition to this, some basics regarding the Average Consensus (AC) algorithm are provided, as well as general information concerning the types of weights that may be used in wireless sensor networks to guarantee a consensus. At last, we introduce two algorithmic variants based on PAST, which are the scope of study of this thesis in the chapters to come.

3.1 Background and state of the art

Subspace estimation, both in spatial and temporal spectral analysis, has been successfully ported from a centralized to a distributed setting [50, 23, 51, 52]. Measurement data from the individual sensor nodes are locally shared and combined to aggregate values. Several widely used methods exist for in-network signal passing, where each node broadcasts relevant information to its nearest neighbors. Some advantages of distribution are high resilience to link and node failures and minimal, if any, need of network topology knowledge.

Various prominent approaches exist for subspace estimation: A classic is the Principal Component Analysis (PCA) used for top-k approximation of a high-dimensional data set by way of computationally costly Singular Value Decomposition (SVD). This approach has recently been improved by various authors such as Candes et al. [53], Wright et al. [54] and Hage et al. [55]. The improved algorithms, jointly called Robust PCA, are less valuable to outliers, and perform low-rank and sparse decomposition instead of the more costly SVD.

An alternative method is a geometric approach, which estimates subspaces as elements of the complex Grassmannian manifold [56]. To improve estimation results,

researchers have applied a Bayesian method to define a posterior density function on the complex Grassmannian manifold probabilities, and a Hilbert-Schmidt norm to quantify estimation errors.

An application regarding distributed subspace tracking is distributed image processing as shown by Song et al. [23]. The authors describe a decentralized camera network for large area surveillance, where neighboring cameras with pan-tilt-zoom capabilities share data during target tracking. Independent from each cameras own estimation of the targets position, a network consensus using various cameras overlapping fields of view can be reached using the Kalman-consensus distributed tracking algorithm. The convergence results of the related distributed Kalman Filtering algorithm have been analyzed by Soummya Kar et al. in [57], who have found that a stable asymptotic estimation error can be reached at each sensor node, assuming network connectivity and a centrally observable signal model.

3.2 Averaging algorithms

In recent work, distributed adaptive algorithms have been proposed to address the issue of estimation over distributed networks. These new algorithms outperform the classical centralized solution, but are based on specific network topologies which lead to scalability constrains. For example, [58] considers the design of distributed architectures based on randomized graph models.

Different strategies are used for distributed fusion and average of the information. Some algorithms are based on interaction between a node and all its adjacent nodes in the network, so as to reach a *consensus*, namely the Consensus algorithms, introduced by Olfati-Saber and Murray in [10, 59].

Another strategy is based on pairwise communications: the gossip-based algorithms such as the push-sum protocol introduced by Kempe in [15] are good alternatives when low communication overhead is desired.

Likewise, the consensus protocol is widely used for obtaining averages over a network [60, 59, 10]. The work in [61, 62] has introduced a distributed algorithm based on Consensus Propagation for frequency estimation over a wireless sensor network.

Due to its simplicity and low computational complexity $O(Nr)$, we believe that PAST is a suitable algorithm to be investigated and hence being distributed for wireless sensor networks applications. This Subsection presents our two algorithmic approaches for distributing PAST based on the average consensus algorithm, a work previously introduced in [61, 63].

3.2.1 Average consensus propagation

Consensus algorithms are iterative protocols where self-governing nodes communicate with other nodes to achieve an agreement about a specific parameter of interest, without the need of a central unit to process this information. At each time iteration, sensor i exchanges information with its neighborhood $\mathcal{N}_i$, such that a common values is asymptotically reached. In this thesis we adopt the Average Consensus (AC), a version of consensus algorithms whose main task is to compute the average of a set of measurements [64, 65]. AC is used as a tool to develop two distributed subspace tracking algorithmic variants for wireless sensor network applications [61, 62]. In the following, we provide a brief explanation about AC and state necessary conditions to be satisfied in order to guarantee convergence to the average value in the network.

Let each node i from graph $\mathcal{H}$ have an associated scalar value (or vector) x_i defined as the state of node i. To initialize this state, the environmental data monitored by node i is used. To update it, the state information x_j from $\mathcal{N}_i$ is taken into account. Reaching a consensus means that $x_i = x_j$. In this work, we restrict to the synchronous case even though asynchronous average consensus (see e.g. [66] and [67]) has been widely addressed in the context of WSN.

The time invariant model: each node i has an initial scalar value $x_i(0) \in \mathbb{R}$, and the initial state vector containing each scalar is given by $\underline{\mathbf{x}}(0) = [x_1(0), x_2(0), \ldots, x_N(0)]^T$. The task of the algorithm is to compute the average $\frac{1}{N}\sum_{i=1}^{N} x_i(0)$. The local state at each node i and time k is updated using a linear combination of its previous value and the information received from its neighbors as

$$x_i(k) = \mathbf{G}_{ii}x_i(k-1) + \sum_{j\in\mathcal{N}_i} \mathbf{G}_{ij}x_j(k-1). \tag{3.1}$$

Here $\mathbf{G}_{ii}$ corresponds to the weight of i and $\mathbf{G}_{ij}$ is the $(ij)^{th}$ entry of the weight matrix associated to the edge $\{ij\}$, which is non-zero if and only if $\{i,j\} \in \mathcal{E}$. These time invariant weights are selected such that every $x_i(k)$ iteratively converges to the average of their initial values $x_i(0)$. Let us introduce the state vector $\underline{\mathbf{x}}(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^T$ that contains all scalar observations in the network. Then, (3.1) is rewritten as the weighted average of the neighbors' data, i.e.,

$$\underline{\mathbf{x}}(k) = \mathbf{G}\underline{\mathbf{x}}(k-1). \tag{3.2}$$

An undirected graph $\mathcal{H}$ assumption indicates that $\mathbf{G}_{ij} = \mathbf{G}_{ji}$ and that the sum of all weights equals one. Therefore, the $\mathbf{G}$ matrix has the same sparsity as the network graph and it is doubly stochastic, meaning that $\underline{\mathbf{1}}_{N\times 1} = [1, \ldots, 1]^T$ is a left and right eigenvector of $\mathbf{G}$. To achieve asymptotic average consensus, the matrix $\mathbf{G}$ must satisfy $\lim_{k\to\infty} \mathbf{G}^k = \frac{1}{N}\underline{\mathbf{1}}\underline{\mathbf{1}}^T = \mathbf{J}$, or similarly, the spectral radius satisfies

$\rho(\mathbf{G}-\mathbf{J})<1$. As a result we obtain

$$\underline{\mathbf{x}}(k)=\mathbf{G}^k\underline{\mathbf{x}}(0)\xrightarrow{k\to\infty}\frac{1}{N}\sum_{i=1}^{N}x_i(0)\cdot\underline{\mathbf{1}}_{N\times 1} \tag{3.3}$$

All local states $x_i(k)$ approach the averaged value of $\underline{\mathbf{x}}(0)$ as k increases.

3.2.2 Design of the weight matrix

A requirement to design a weight matrix is to have knowledge of the network topology in addition to the information available at each node. Some weighting models assume that the nodes have global information available. For instance, the maximum degree weights demand information about the maximum node degree in the whole network of a static topology [64]. This must be previously determined and broadcast across the network before using any consensus algorithm. Similarly, the pair-wise/geographic gossip introduced in [16] for a time varying topology needs some information about the global network. Here, the algorithm selects randomly (with probability $\frac{1}{N}$) a transmitting node i to establish a bidirectional communication with a randomly selected node j. While pair-wise gossip communicates only with its neighboring $\mathcal{N}_i$, the geographic gossiping may connect with any $j\in P$.

Other models are implemented using local information gathered directly by the nodes or obtained through cooperation, i.e.: The local degree weights, where each node must know the out-degrees of its neighborhood [64]. A well known method for assigning weights in a time varying topology graph is the nearest neighbor rule [68, 69]. Nevertheless, it does not preserve the average because $\underline{\mathbf{1}}^T\mathbf{G}(k)\neq\underline{\mathbf{1}}^T$ ($\underline{\mathbf{1}}$ is not a left eigenvector of $\mathbf{G}(k)$). This approach is therefore unsuitable for calculating the average consensus.

Metropolis weights [70] are very simple to calculate, and they preserve the average and guarantee asymptotic convergence. Although these weights are time dependent, a static network scenario (number of edges does not vary over time) is addressed at the full length of this thesis. The Metropolis weights are given as

$$\mathbf{G}_{i,j}\in\mathbb{R}^{N\times N}=\begin{cases}\frac{1}{(1+\max\{d_i,d_j\})}, & \{i,j\}\in\mathcal{E}\\ 1-\sum_{\nu\in\mathcal{N}_i}\mathbf{G}_{i\nu}, & i=j\\ 0, & \text{otherwise.}\end{cases} \tag{3.4}$$

It is easy to observe that each node requires knowledge of the degree of each neighboring node $j\in\mathcal{N}_i$ in order to calculate the weights. Since the edges are time varying, transmitting this information together with the observation data (state of node) at each iteration guarantees a real time update of the weight matrix. In (3.4) the weight at each edge depends on the largest degree at its two incident nodes. The self weight from node i is selected such that its total weight sum equals 1.

3.3 Distributed algorithmic variants based on PAST

The main goal of this thesis is to develop and investigate two distributed algorithmic approaches for signal subspace tracking applied in a wireless sensor network, without the need for a fusion center. To this aim, we start from Projection Approximation Subspace Tracking (PAST), which is a well-investigated algorithm suitable for implementation in a centralized network. We arrive at a distributed approximation of Algorithm 1 by letting each sensor i broadcast some local observation variables to its neighborhood $\mathcal{N}_i$. Vice versa, the received messages at the sensor nodes from its neighborhood are fused by employing average consensus propagation. On next, we provide a more detailed explanation regarding the variables exchanged in the network, where the averaging process occurs and finally introduce the distributed versions of Algorithm 1.

3.3.1 Distribution of the signal vector variable y

To decentralize the PAST algorithm, we regard the Algorithm 1 containing the primary set of equations locally running at each node i (in parallel). This generates the local variables $\underline{\mathbf{x}}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times 1}, \underline{\mathbf{y}}_i(k) \in \mathbb{C}^{r\times 1}, \underline{\mathbf{h}}_i(k) \in \mathbb{C}^{r\times 1}, \underline{\mathbf{g}}_i(k) \in \mathbb{C}^{r\times 1}, \widehat{\mathbf{R}}_i^{yy}(k) \in \mathbb{C}^{r\times r}$, $\underline{\mathbf{e}}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times 1}$ and $\mathbf{W}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times r}$, where the variable $\mathcal{N}_i$ refers to the neighborhood (node degree) of i from Section 2.2. In order to facilitate our later calculations, we extend the definition of neighborhood to be composed of the j neighboring nodes connected with i by its edges and also including itself as own neighbor.

We consider that every node i broadcasts its own scalar environmental observation $\{x_i(k)\}$ and at the same time receives $\{x_j(k)\}_{\substack{j\in\mathcal{N}_i,\\ j\neq i}}$ from its j neighbors. With this available information, every node i aggregates the local observation vector $\underline{\mathbf{x}}_i(k)$, i.e.,

$$\underline{\mathbf{x}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k), \tag{3.5}$$

where $\underline{\mathbf{x}}_k$ is the data vector observed in the whole network. We introduce the selection matrix $\mathbf{S}_i \in \mathbb{R}^{N\times|\mathcal{N}_i|}$, a truncated version of the adjacency matrix in (2.1), that chooses $\mathcal{N}_i$ observations out of $\underline{\mathbf{x}}(k)$ corresponding to node i. The selection matrix is defined as

$$(\mathbf{S}_i)_{l,j} = \begin{cases} 1 & \text{if } j = l\text{th node} \in \mathcal{N}_i \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

Recall the average consensus algorithm [64, 65] and let us propose the following distributed scheme: Node i sends $\{\underline{\mathbf{y}}_i(k)\}$ to the neighborhood $\mathcal{N}_i$ and receives

$\{\underline{\mathbf{y}}_j(k)\}_{\substack{j\in\mathcal{N}_i,\\ j\neq i}}$. Hence, the averaging process occurs when

$$\underline{\mathbf{y}}_i(k,t) = \mathbf{G}_{ii}\underline{\mathbf{y}}_i(k,t-1) + \sum_{\substack{j\in\mathcal{N}_i,\\ j\neq i}} \mathbf{G}_{ij}\underline{\mathbf{y}}_j(k,t-1). \tag{3.7}$$

Here, $\mathbf{G}_{ii}$ stands for the weight of node i and $\mathbf{G}_{ij}$ is the weight associated to j from $\mathcal{N}_i$. The two time index represent that for each time k the algorithm is updated, there are t consensus rounds taking place. This means that $t = 1, \ldots, t_{\max}$[1] is the averaging time window used for each node in order to reach a global consensus, e.g.

$$\underline{\widetilde{\mathbf{y}}}_i(k) = \underline{\mathbf{y}}_i(k, t_{\max}). \tag{3.8}$$

This information exchange permits that all nodes involved in the communication process have knowledge of the neighboring observations. We propose to locally average the vector $\underline{\mathbf{y}}_i(k)$ in node i by fusing information aggregated in its associated neighborhood $\mathcal{N}_i$. The local averaging is described in (3.7), where we calculate the internal weighted $\underline{\mathbf{y}}_i(k)$ together with the weighted $\underline{\mathbf{y}}_j(k-1)$. This is subsequently iterated over several consensus run, which directly depend on the size of the sensor network (the larger the network, the more consensus runs are needed such that i achieves the average $\underline{\widetilde{\mathbf{y}}}_i(k)$ from the whole network). The reason why we decided to distribute this particular variable is because according to (5.32), this vector contains data from the previous updated signal subspace $\mathbf{W}_i(k-1)$ as well as new arriving observation data $\underline{\mathbf{x}}_i(k)$. As a result, every sensor in the system has indirect knowledge of these parameters.

After defining in which part of the algorithm the distribution takes place, we introduce Algorithm 2. The average consensus $\underline{\widetilde{\mathbf{y}}}_i(k)$ is estimated at every node of the system and the following calculations comply with the original equations from Algorithm 1.

At the end of step $k-1$, every node i sends to its adjacent nodes $\mathcal{N}_i$ its own estimation of $\underline{\mathbf{y}}_i(k-1)$. Then, at the beginning of step k, every node i receives $\{x_j, \underline{\mathbf{y}}_j(k-1), w_i\}_{j\in\mathcal{N}_i}$ from its neighbors and compute a new average, to be sent at the end of step k.

3.3.2 Distribution of the correlation matrix $\widehat{\mathbf{R}}$

In the second approach, node i sends the correlation matrix $\{\widehat{\mathbf{R}}_i^{yy}(k)\}$ to $\mathcal{N}_i$ and receives $\{\widehat{\mathbf{R}}_j^{yy}(k)\}_{\substack{j\in\mathcal{N}_i,\\ j\neq i}}$. The averaging occurs as long as

$$\widehat{\mathbf{R}}_i^{yy}(k,t) = g_{ii}\widehat{\mathbf{R}}_i^{yy}(k,t-1) + \sum_{\substack{j\in\mathcal{N}_i,\\ j\neq i}} g_{ij}\widehat{\mathbf{R}}_j^{yy}(k,t-1), \tag{3.9}$$

[1] $t_{\max}$: time when the consensus is reached

Algorithm 2: Local description of distributed PAST with $\underline{\mathbf{y}}_i$ averaging

Input: $\beta, \widehat{\mathbf{R}}_i^{yy}(0), \ldots, \widehat{\mathbf{R}}_N^{yy}(0), \mathbf{W}_i(0), \ldots, \mathbf{W}_N(0)$
for $k := 1, 2, \ldots$ **do**
  for $i := 1, 2, \ldots, N$ **do**
    Input: $x_i(k)$ *scalar environmental observation*
    $\underline{\mathbf{x}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k)$
    $\underline{\mathbf{y}}_i(k) = \mathbf{W}_i^H(k-1)\underline{\mathbf{x}}_i(k)$
    Apply (3.7) *and* (3.8) *to obtain the average consensus* $\widetilde{\underline{\mathbf{y}}}_i(k)$
    $\underline{\mathbf{h}}_i(k) = \widehat{\mathbf{R}}_i^{yy}(k-1)\widetilde{\underline{\mathbf{y}}}_i(k)$
    $\underline{\mathbf{g}}_i(k) = \underline{\mathbf{h}}_i(k)/[\beta + \widetilde{\underline{\mathbf{y}}}_i(k)^H \underline{\mathbf{h}}_i(k)]$
    $\widehat{\mathbf{R}}_i^{yy}(k) = \frac{1}{\beta}[\widehat{\mathbf{R}}_i^{yy}(k-1) - \underline{\mathbf{g}}_i(k)\underline{\mathbf{h}}_i^H(k)]$
    $\underline{\mathbf{e}}_i(k) = \underline{\mathbf{x}}_i(k) - \mathbf{W}_i(k-1)\widetilde{\underline{\mathbf{y}}}_i(k)$
    $\mathbf{W}_i(k) = \mathbf{W}_i(k-1) + \underline{\mathbf{e}}_i(k)\underline{\mathbf{g}}_i^H(k)$
  end for
end for

and the global consensus is achieved when

$$\widetilde{\mathbf{R}}_i^{yy}(k) = \widehat{\mathbf{R}}_i^{yy}(k, t_{\max}). \tag{3.10}$$

Algorithm 3: Local description of distributed PAST with $\widehat{\mathbf{R}}_i^{yy}$ averaging

Input: $\beta, \widehat{\mathbf{R}}_i^{yy}(0), \ldots, \widehat{\mathbf{R}}_N^{yy}(0), \widetilde{\mathbf{R}}_i^{yy}(0), \ldots, \widetilde{\mathbf{R}}_N^{yy}(0, \mathbf{W}_i(0), \ldots, \mathbf{W}_N(0)$
for $k := 1, 2, \ldots$ **do**
  for $i := 1, 2, \ldots, N$ **do**
    Input: $x_i(k)$ *scalar environmental observation*
    $\underline{\mathbf{x}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k)$
    $\underline{\mathbf{y}}_i(k) = \mathbf{W}_i^H(k-1)\underline{\mathbf{x}}_i(k)$
    $\underline{\mathbf{h}}_i(k) = \widetilde{\mathbf{R}}_i^{yy}(k-1)\underline{\mathbf{y}}_i(k)$
    $\underline{\mathbf{g}}_i(k) = \underline{\mathbf{h}}_i(k)/[\beta + \underline{\mathbf{y}}_i(k)^H \underline{\mathbf{h}}_i(k)]$
    $\widehat{\mathbf{R}}_i^{yy}(k) = \frac{1}{\beta}[\widetilde{\mathbf{R}}_i^{yy}(k-1) - \underline{\mathbf{g}}_i(k)\underline{\mathbf{h}}_i^H(k)]$
    Apply (3.9) *and* (3.10) *to obtain the average consensus* $\widetilde{\mathbf{R}}_i^{yy}(k)$
    $\underline{\mathbf{e}}_i(k) = \underline{\mathbf{x}}_i(k) - \mathbf{W}_i(k-1)\underline{\mathbf{y}}_i(k)$
    $\mathbf{W}_i(k) = \mathbf{W}_i(k-1) + \underline{\mathbf{e}}_i(k)\underline{\mathbf{g}}_i^H(k)$
  end for
end for

3.4 Summary

This chapter presented two algorithmic variants based on PAST. In a first approach, the Algorithm 2 is introduced. Here, every sensor i locally tracks the signal subspace and its own internal state vector $\underline{\mathbf{y}}_i(k)$. Furthermore, a consensus of the local state

and the state vectors from its neighborhood $\mathcal{N}_i$ is reached using Metropolis-weighted $\underline{\mathbf{y}}_j(k)$. To this end, every sensor node i broadcasts its local scalar observation $x_i(k)$ and its locally filtered r-dimensional vector $\underline{\mathbf{y}}_i(k)$ to its neighborhood $\mathcal{N}_i$ of sensor nodes. Thus, every node broadcasts $2(r+1)+1$ real-valued variables per time step. In a second approach, the correlation matrix $\widehat{\mathbf{R}}_i^{yy}(k)$ is distributed among the nodes.

Due to the simplicity of these algorithms, we believe that a well performing low-cost implementation of a distributed PAST algorithm with low communication overhead is a feasible goal. In the next chapters, we show performance results of both approaches, analyze their stability, and discuss their drawbacks.

Chapter 4

Performance analysis of algorithmic approaches

THE goal of this chapter is to evaluate the performance of both Algorithm 2 and Algorithm 3. To this end, some simulation results, showing the tracking capabilities of two signals, are introduced in Section 4.1. The angle difference between the true $\boldsymbol{\Upsilon}$ and the estimated $\mathbf{W}$ subspaces, as well as the root mean square error for several signal to noise ratios are presented in Section 4.2 and Section 4.3, respectively. These common benchmarks allow to investigate the algorithm's behavior for different network sizes and topologies. Last, a summary and some conclusions are provided in Section 4.4.

4.1 Direction of Arrival estimation

As previously introduced in Chapter 2, the Direction-of-Arrival (DOA) is a function of smart antenna arrays, useful to define the direction from where a received electromagnetic wave originated.

These antenna arrays allow to ameliorate the resolution [1] of DOA estimation, when compared to a single antenna. An initial approach for direction finding was based on the Fourier transform (Delay and Sum algorithm), which later came to be known as the conventional beamforming. Nevertheless, their inability to resolve for multiple signal components and accurately estimate their Direction-of-Arrival gave rise to the so-called high resolution methods. There is a large number of applications regarding direction finding. In mobile communications, it allows to reduce the interference and improve the communication quality. It is used in radar, radio, astronomy, sonar and navigation. In rescue, it is possible to identify the origin

[1] Ability to measure the angle of arrival of a radio signal

of an emergency phone call, such that a rescue team is deployed towards the exact location.

MUSIC [40] is a spectral estimation-based method, which task is to extract the desired signal parameters out of the received data provided by a sensor array. These parameters could be the number of one or multiple signal sources, the signal frequencies or the Direction Of Arrival. In the experiments to follow, the MUSIC algorithm introduced in Chapter 2, is used to find the DOA of $r = 2$ incoming waves. As depicted in the signal flow chart in Figure 4.1, the MUSIC algorithm is used every time the update of the $\mathbf{W}(k)$ signal subspace matrix takes place.

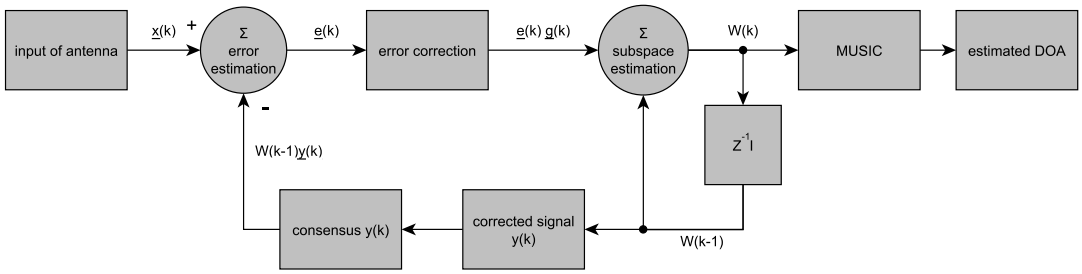

Figure 4.1: Flow chart representation of Algorithm 2. The Average Consensus $\widetilde{\underline{\mathbf{y}}}(k)$ is used to calculate the local errors and posteriorly update the $\mathbf{W}(k)$ signal subspace. The later matrix is influenced from the previous averaging process and has indirect knowledge of the global state in the network. In a last step, the MUSIC algorithm is used for DOA finding.

Furthermore, this section examines both distributed Average-Consensus based algorithms from Chapter 3. Although this thesis focuses on studying the subspace $\mathbf{W}(k)$, analyzing the algorithmic behavior in the presence of several signals sources is still a relevant tool, especially when these methods operate in practice.

4.1.1 Simulation experiments

To start, consider a wireless sensor network composed of N nodes placed in a 2-D jittered cartesian grid. Namely, we evaluate those topologies from Figures 2.3 and 2.4. These sensors observe a total of $K = 120$ snapshots and track the Direction-of-Arrival of $r = 2$ impinging waves. It is assumed that each sensor in the neighborhood $\mathcal{N}_i$ correctly receives the broadcasted messages from node i, with probability 1. Recall the signal model from Section 2.3.2, e.g.,

$$\underline{\mathbf{x}}(k) = \boldsymbol{\Upsilon}(\boldsymbol{\theta}(k))\underline{\mathbf{s}}(k) + \underline{\mathbf{n}}(k), \tag{2.19}$$

and allow $\boldsymbol{\theta}(k)$ to be slowly time-varying, i.e. $\boldsymbol{\theta}(k) \approx \boldsymbol{\theta}$. To display the DOA estimation vs. time, the spectral MUSIC algorithm [40] is applied for extracting r components out of the signal subspace matrix $\mathbf{W}(k)$. All simulation experiments are carried out using 100 different noise realizations (Monte Carlo runs), and therefore all results to be presented are first averaged over all MC runs and then over all N sensor elements [2].

In a first experiment, the network size is set to $N = 36$. The tracking capabilities of two signals $r = 2$ operating at the same frequency $f = 2.4$ GHz are analyzed. Both signals have a constant azimuth at $\theta_1 = [144°]$ and $\theta_2 = [90°]$, represented by a dashed line in red and blue, respectively. At iteration $k = 60$ they undergo a "step" change and become constant again at $\theta_1 = [-108°]$ and $\theta_2 = [-72°]$.

Since PAST has its roots in the Recursive Least-Squares algorithm, it is also governed by the so-called forgetting factor β. As a result, this parameter is investigated as it can be tuned to find a compromise between the tracking capabilities, convergence rate and stability. Whereas the first two parameters are studied in Chapter 5, this Subsection explores how the selection of β impacts the algorithm's performance. For the following simulations, the selected forgetting factors are:

- $\beta = 0.9$, represented by a solid line
- $\beta = 0.8$, represented by a dash-dot line
- $\beta = 0.6$, represented by a dotted line,

Figure 4.2 depicts the performance of the PAST algorithm proposed by Yang in [8]. The estimated direction of arrivals $\widehat{\theta}_1$ and $\widehat{\theta}_2$ are described by the magenta and green colors, respectively. In 1), the tracking accuracy of the algorithm is tested at SNR= -5. Observe that the higher the β value, the more pronounced the lag in azimuth transition. This is due to the fact, that older samples are highly weighted and the algorithm needs more time to integrate the new samples indicating the azimuth change. A smaller forgetting factor adapts very fast to the step change. Nevertheless, it is also more sensitive to noise. Subfigure 2.) displays the performance for the same β values, but now for SNR= 5. As expected, the higher signal to noise ratio leads to a better performance and the behavior among the β's is equivalent to that observed in a.). These results describe the "centralized" solution where no information exchange takes place, and are used as a reference to evaluate how good the outcome of the distributed approaches are.

Let us study the behavior of the proposed Algorithm 2 from Chapter 3. The Figure 4.3 shows the performance when $\underline{\mathbf{y}}_i$ is exchanged in the scenario f.) from Figure 2.4. That is to say, the neighborhood size for each i is set to $\frac{1}{4}N$, i.e.,

[2] Even though the performance from node to node varies due to their different neighborhood size, we believe that in a scenario where so many parameters are studied, the study of the average network behavior is the main approach.

$|\mathcal{N}_i| = 9$. Results in 1.) and 2.) display a satisfactory behavior for both SNR values. This topology meets a fair compromise between tracking accuracy and the amount of information exchanged inside the network.

In Figure 4.4 the results obtained from the evaluation of Algorithm 3 are rather disappointing. The algorithm tracks θ_1 correctly, but right after the step it keeps locked at the same value, where a meaningless step change is observed at $k = 60$. Even for higher SNR= 5 in 2.), there is no improvement. A worse situation is seen in the tracking of θ_2, where the estimates do not manage to track the signal at all. This results actually show the worst possible scenario. In the appendix B it is possible to observe that the algorithmic performance provides is good for fully and dense connected networks.

After comparing both algorithmic versions, it becomes evident that Algorithm 2 provides the best possible tracking capabilities at the lowest message passing rate. Recall that the message exchange is what consumes most of the power resources in a sensor. Therefore, a weak connected network such as e) from Figure 2.4 seems a good compromise between tracking accuracy and the use of the available resources, especially when the performance is comparable to the centralized solution in Figure 4.2. The reader is referred to Appendix A, where the tracking capabilities for the full, dense, sparse and regular neighborhood $|\mathcal{N}_i| = 5$ topologies from Figure 2.4 are presented.

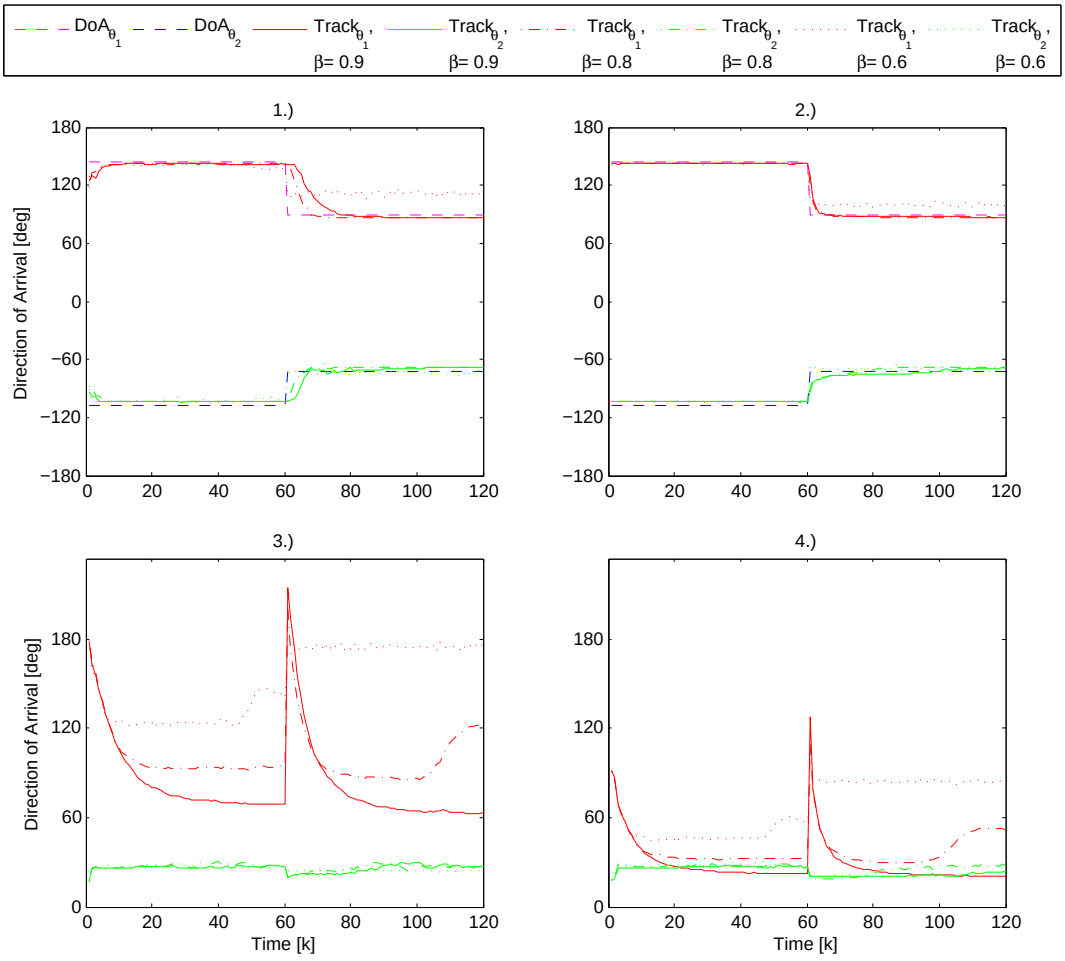

Figure 4.2: Subfigure 1.) and 2.) display the tracking capabilities of the centralized PAST algorithm for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

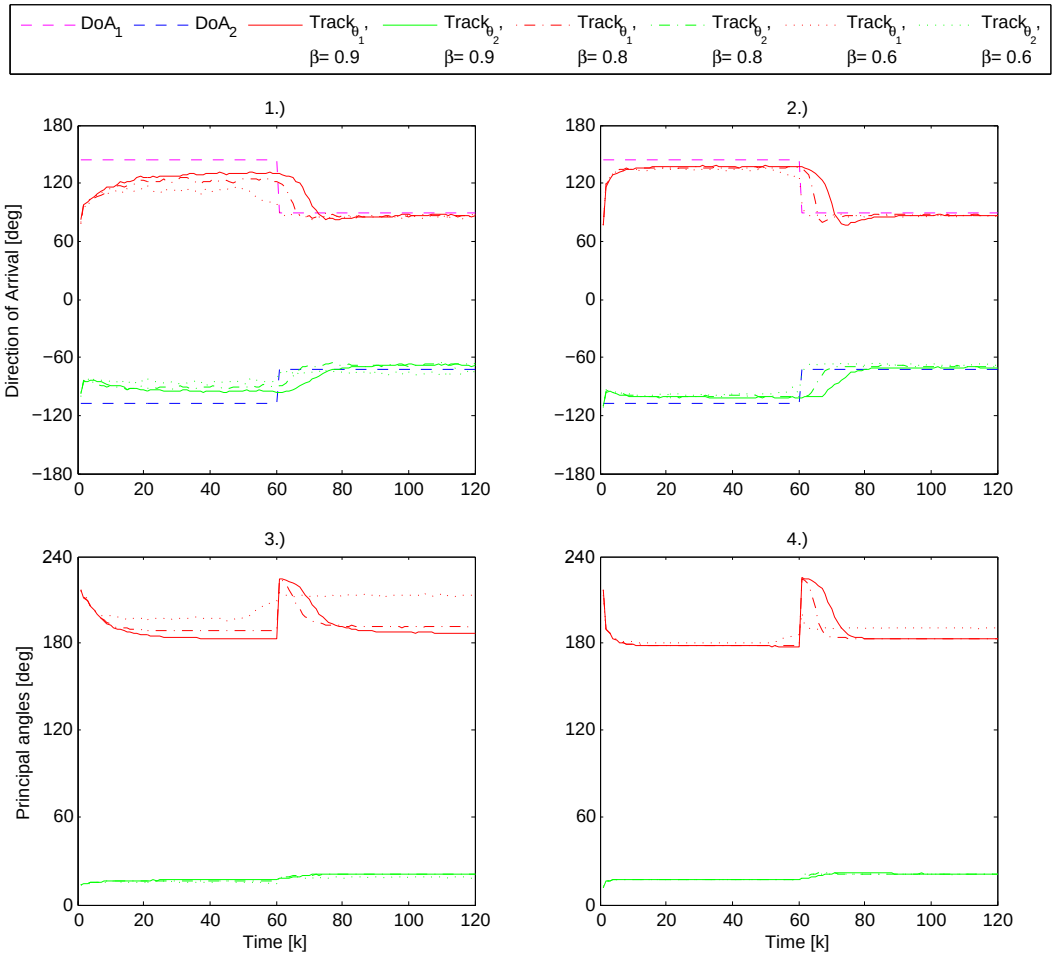

Figure 4.3: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $|\mathcal{N}_i| = 9$. This is the scenario e) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

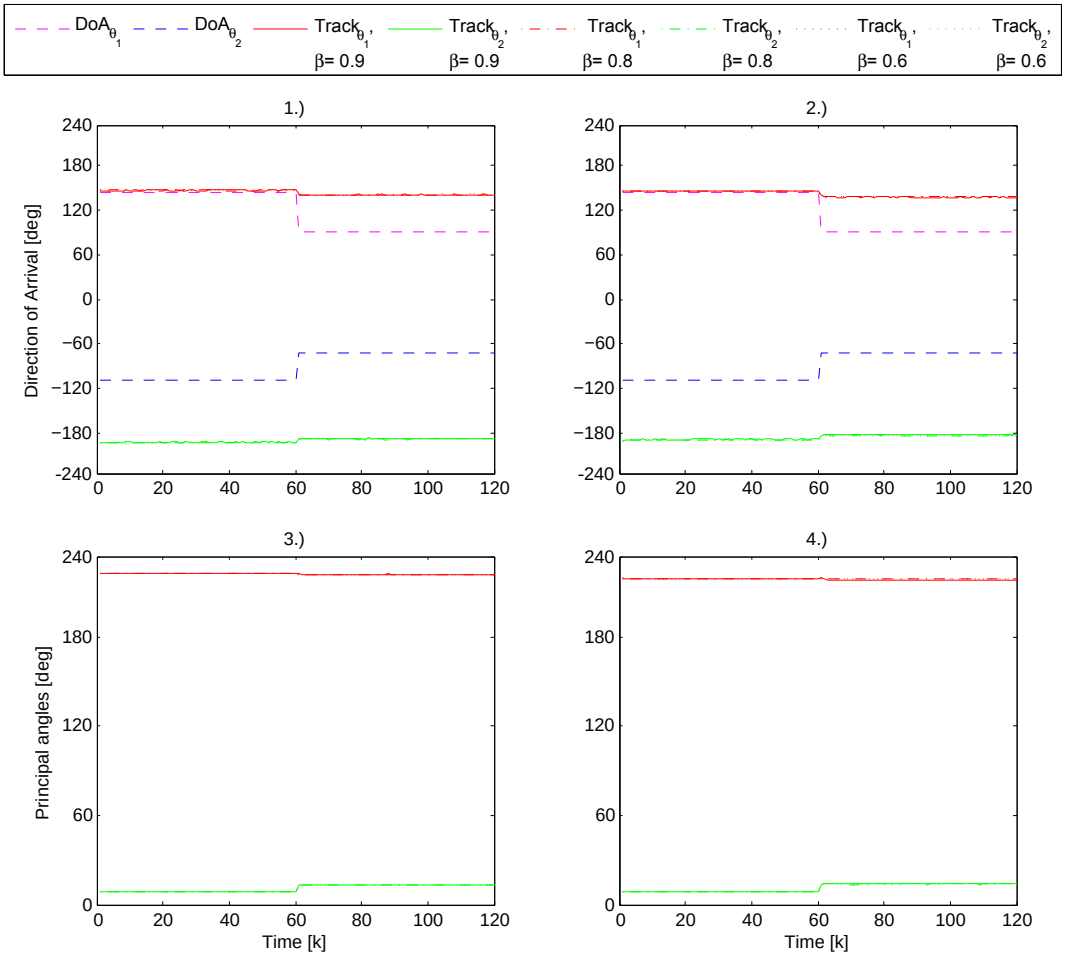

Figure 4.4: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\mathbf{R}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 9$. This is the scenario e) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

4.2 Principal Angles Between Subspaces (PABS)

The interest regarding the PABS originated at Jordan [71] in 1875. In the last fifty years, this research area has gained more popularity as it is used in statistics, data mining [72], information retrieval [73] and even pattern recognition [74][75].

The most common approaches for comparing subspaces are based on the singular value decomposition. Golub and Van Loan provide a survey on these methods in [76]. On a first note, the orthogonal Procrustes problem solved by subspace rotations [77] is addressed. This problem explores how likely it is to rotate $\mathbf{W}_k \in \mathbb{C}^{N\times r}$ into $\boldsymbol{\Upsilon}_k \in \mathbb{C}^{N\times r}$ by means of $\mathbf{Q}_k \in \mathbb{C}^{r\times r}$, such that the distance between both subspaces is minimized, i.e.,

$$\min ||\boldsymbol{\Upsilon} - \mathbf{W}(k)\mathbf{Q}(k)||_F \tag{4.1}$$

subject to $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Another method, the one used in this thesis, refers to the angle between subspaces [78], which is a generalization of the angles between two vectors.

Let the estimated signal subspace matrix $\mathbf{W}(k)$ and the true subspace given by the steering matrix $\boldsymbol{\Upsilon}$ be subspaces in $\mathbb{C}^{N\times r}$, both with the same dimension. The first principal angle between these subspaces is defined as

$$\alpha_1 := \min\left\{\arccos\left(\frac{\langle \underline{\mathbf{u}}, \underline{\mathbf{v}}\rangle}{||\underline{\mathbf{u}}||||\underline{\mathbf{v}}||}\right) |\underline{\mathbf{u}} \in \mathbf{W}(k), \underline{\mathbf{v}} \in \boldsymbol{\Upsilon}\right\} = \angle(\underline{\mathbf{u}}_1, \underline{\mathbf{v}}_1). \tag{4.2}$$

In order to calculate the first (and smallest) principal angle α_1, it is necessary to choose two unit vectors $\underline{\mathbf{u}}_1 \in \mathbf{W}(k)$ and $\underline{\mathbf{v}}_1 \in \boldsymbol{\Upsilon}$, such that the angle between them is minimized. The vectors $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{v}}_1$ are known as the first principal vectors. For finding α_2, select the unit vector $\underline{\mathbf{u}}_2 \in \mathbf{W}(k)$ orthogonal to $\underline{\mathbf{u}}_1$ and $\underline{\mathbf{v}}_2 \in \boldsymbol{\Upsilon}$ which is orthogonal to $\underline{\mathbf{u}}_1$. Again, the angle α_2 is minimized. This searching procedure occurs until each angle $l = 1, \ldots, r$ has been calculated.

$$\alpha_l := \min\left\{\arccos\left(\frac{\langle \underline{\mathbf{u}}, \underline{\mathbf{v}}\rangle}{||\underline{\mathbf{u}}||||\underline{\mathbf{v}}||}\right) |\underline{\mathbf{u}} \in \mathbf{W}(k), \underline{\mathbf{v}} \in \boldsymbol{\Upsilon}, \underline{\mathbf{u}} \perp \underline{\mathbf{u}}_m, \underline{\mathbf{v}} \perp \underline{\mathbf{v}}_m, m = 1, 2, \ldots, l-1\right\}. \tag{4.3}$$

The computation of the principal angles in these simulations use the function subspacea.m created by Knyazev [79], which is more robust against round-off errors.

The Figure 4.2 presents in 3.) and 4.) the principal angles between the subspace spanned by the columns of the estimated signal subspace $\mathbf{W}(k)$ and the original signal subspace spanned by the columns of the matrix $\boldsymbol{\Upsilon}(\boldsymbol{\theta})$.

From the theory introduced in Section 4.2, we know that the first principal angle always equals zero if the subspaces are equal. In this case, the first principal angle

is provided by θ_2. This value is the closest to zero and does not improve much for higher SNR values. Note that different β values have no impact on the angle difference estimation. On the other hand, the second principal angle is provided by θ_1. In 3.) we observe that the impact of the step at $k = 60$ is more pronounced. Likewise, the angle difference decreases for higher β values, and drops even further for higher SNR values. Subfigure 4.) shows that for $\beta = 90$ both angles are almost equal. An interesting behavior occurs when $\beta = 0.8$, as the angle becomes larger after iteration $k = 100$. It is possible that for this particular value, the subspace needs more time to stabilize.

In Figure 4.4, the first principal angle in 3.) and 4.) shows a similar performance to that in Figure 4.2. Here, the second principal angle yields 180°. This suggests that the second principal vectors originating from this angle point to opposite directions. Figure 4.4 introduces an interesting result: even though the DOA estimation is poor, the first principal angle keeps rather low. Nevertheless, the second principal angle rises over 200°.

The Figure 4.5 introduces the angle difference between the subspaces for the previous selected forgetting factors. These results are averaged over all MC runs and over all time iterations. Observe that at each topology, the right hand side figures related to the first principal angle display very similar results for all SNR values. Nonetheless, those at the left side show a dramatic improvement when the SNR increases. Also note, that the topology results at the right hand side are upside down when compared to those at the left hand side. The orthogonality constraint between the first and the second principal vectors probably originates this shift. Algorithm 3 is evaluated in Figure 4.6. As expected, the principal angles behave similar to those in Figure 4.5, but with a higher angle error.

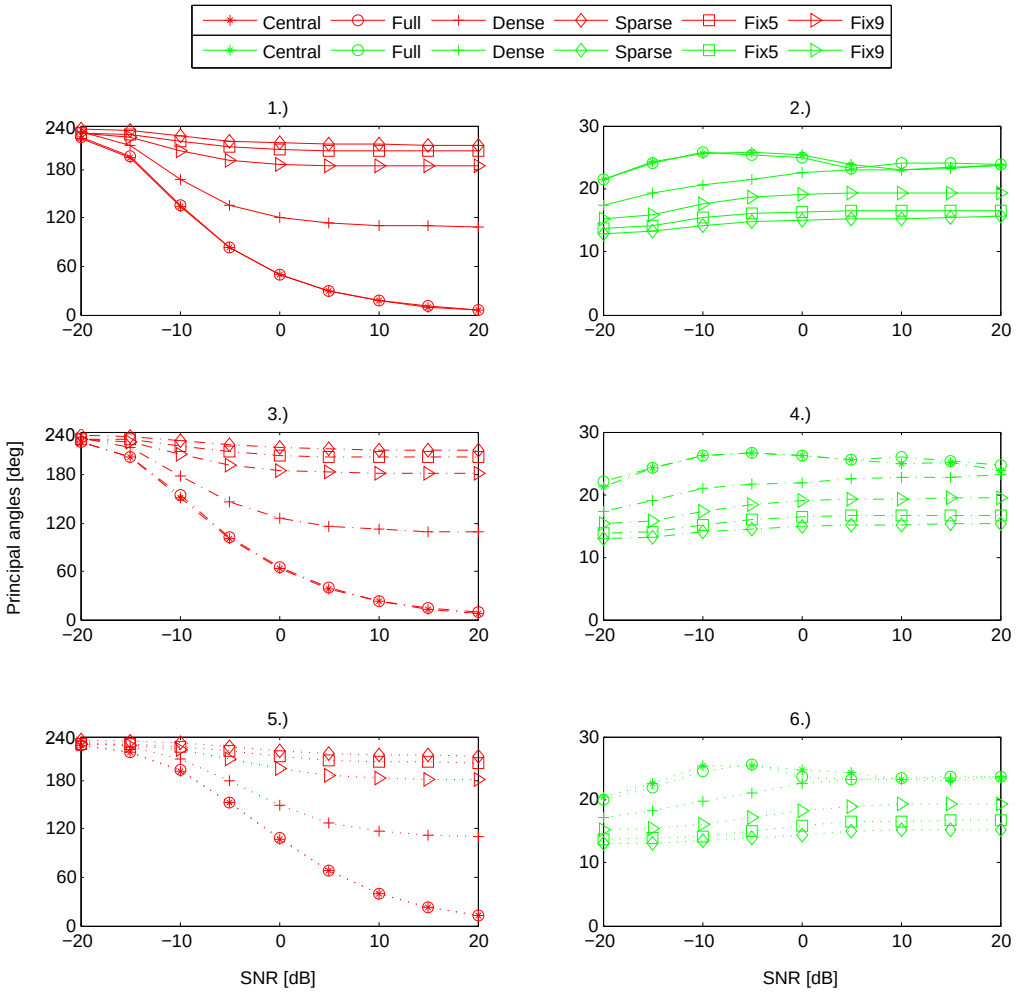

Figure 4.5: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the step signals. The Algorithm 2 is analyzed for the topologies proposed in Figure 2.4. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

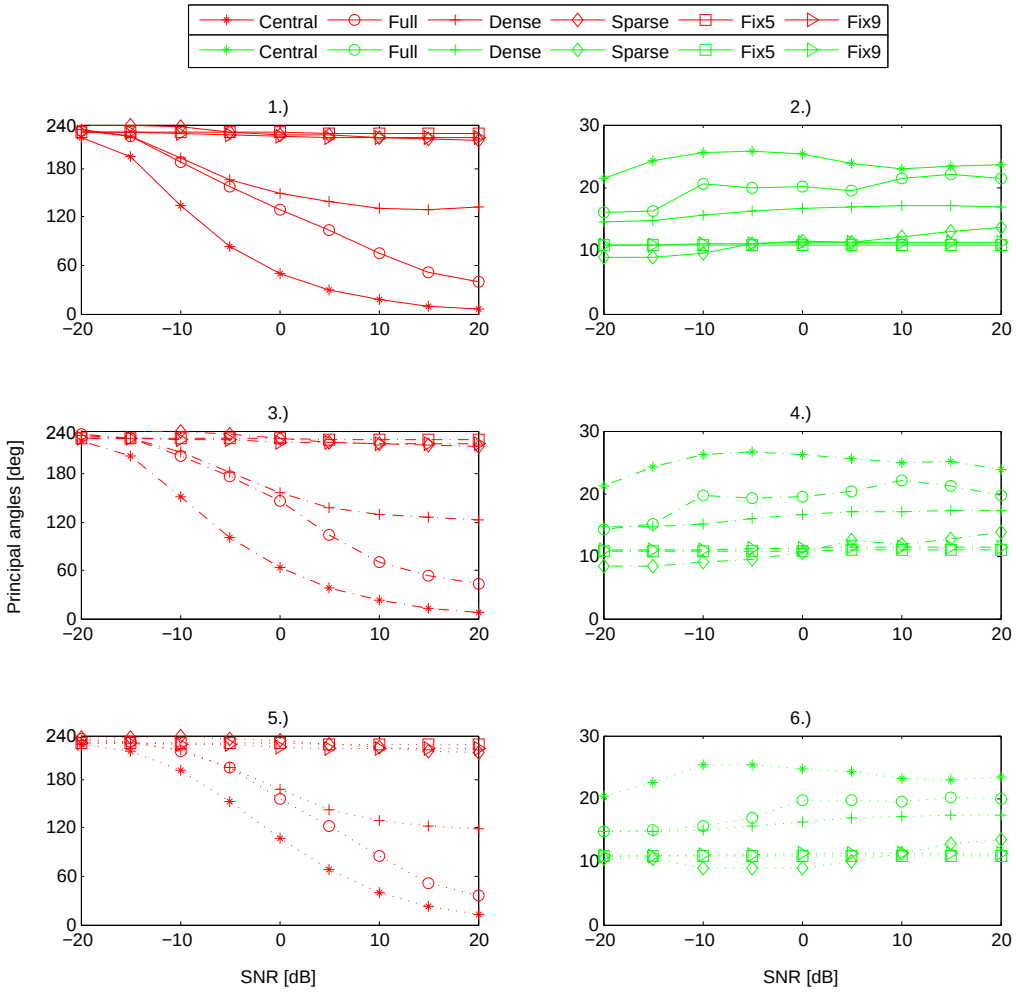

Figure 4.6: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the step signals. The Algorithm 3 is analyzed for the topologies proposed in Figure 2.4. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

4.3 Root mean square error

There are a variety of those so-called "measures of goodness of fit". For instance, the Root Mean Square Error is used to gauge the quality of a model using the differences between the estimated and observed results. Smaller results mean a better reproduction of the observable data by the underlying model. As the quality of the model can be distorted by various factors, i.e. sample size, estimator accuracy, this work considers the RMSE as a quality indicator tool.

The formula used to calculate the RMSE for the first signal is given as

$$\mathrm{RMSE}_{\theta_1} = \sqrt{\frac{1}{K}\frac{1}{N}\frac{1}{MC}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{mc=1}^{MC}|\theta_1(k,i,mc) - \widehat{\theta}_1(k,i,mc)|^2}, \tag{4.4}$$

and for the second

$$\mathrm{RMSE}_{\theta_2} = \sqrt{\frac{1}{K}\frac{1}{N}\frac{1}{MC}\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{mc=1}^{MC}|\theta_2(k,i,mc) - \widehat{\theta}_2(k,i,mc)|^2}. \tag{4.5}$$

Recall the tracking in Figure 4.2 for the centralized PAST. When these results are compared with Figure 4.3 for the Fix9 topology, it becomes evident that even though the later is less densely connected, the tracking is still satisfactory. This motivated to study the performance in terms of the RMSE for both incoming wave signals by means of 4.4 and 4.5.

It is interesting to see in Figure 4.7, that $\widehat{\theta}_1$ goes below $\widehat{\theta}_2$ for the centralized case. In the fix9 case, it works the other way around. This suggests, that the algorithm used for extracting the frequencies out of the subspace (i.e. MUSIC) can be optimized such that both signals are separated more accurately. Naturally, the steepest slope corresponds to the centralized PAST, with $\widehat{\theta}_1$ presenting the lowest errors, as the estimation capabilities are proportional to the amount of information collected and shared. This is better observed in Figure 4.8, where the performance is averaged over both signals as:

$$\mathrm{RMSE}_{\theta_r} = \frac{1}{2}(\mathrm{RMSE}_{\theta_1} + \mathrm{RMSE}_{\theta_2}). \tag{4.6}$$

The behavior is the same: centralized PAST or fully connected are the best achievable values. The worst results are given by those topologies with lower connectivity. Observe that the selection of the forgetting factor does not dramatically change the overall performance.

In effect, the high forgetting factors have better performance for slowly varying or constant signals (as in Figure 4.3). Nevertheless, the local differences in performance (namely how fast the algorithm adapts to the step) are negligible when the signals are averaged over time as well. As a result, the forgetting factor $\beta = 0.6$ in c.)

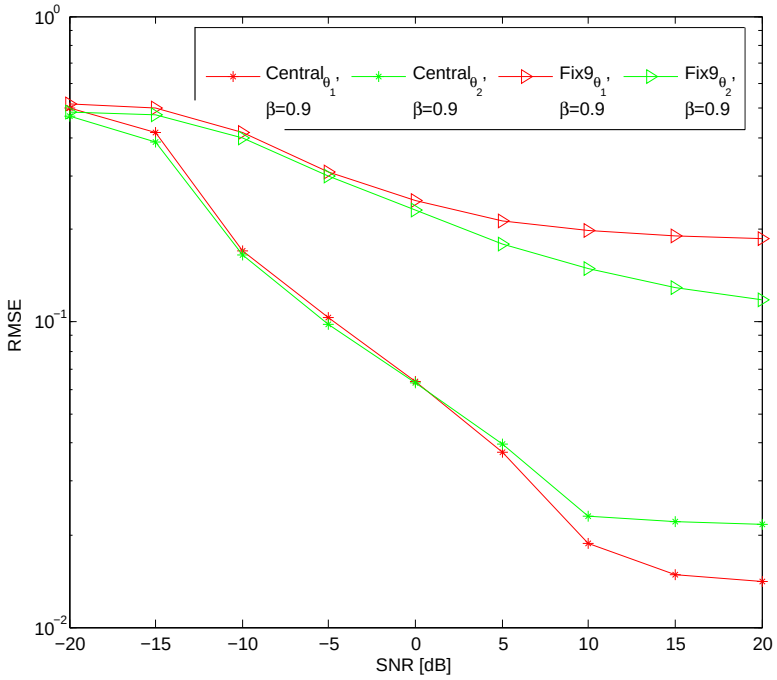

Figure 4.7: Evaluation of the Root Mean Square Error for the centralized PAST and for Algorithm 2 using the scenario e.) introduced in Figure 2.4. This experiment depicts the performance of both algorithms when the signals undergo a step change. The forgetting factor is set to $\beta = 0.9$

shows a similar performance when contrasted to $\beta = 0.9$ in a.) and $\beta = 0.8$ in b.). Comparing these results with those in Figure 4.9, where the Algorithm 3 suggests again that this approach is more sensitive to the information exchange.

Let us repeat the above experiment, but considering slowly varying signals. That is to say, the DOAs are constant at $\theta_1 = [144°]$ and $\theta_1 = [-144°]$, but after iteration 22 they start to approach each other, and become constant again at iteration 98. It is evident, that the tracking capabilities start to decrease with increasing proximity of the signals. This occurs because the search procedure of MUSIC is not able to identify both signals. It searches for the r peaks in its pseudospectrum related to the most dominant components (see Figure 2.6), but when the signals are so close to each other, it is likely that the subspace is reducesd to only one dominant eigenvector

with its corresponding eigenvalue. Closely spaced signals present a high correlation between each other, and as they move closer together their corresponding eigenvectors become more parallel. As a result, the subspace eventually looses a dimension with only a single dominant eigenvalue that makes the correct identification of signal and noise subspace difficult.

Again, the Algorithm 2 presents the best results when compared with the approach when $\widehat{\mathbf{R}}$ is distributed. The angle difference between the subspaces in Figure 4.13 presents the same behavior as for the step-change case. While the first principal angle (at the right side) related to θ_2 stays very close to zero for all SNR values, the angle difference related to θ_1 changes according to the Signal to Noise Ratio. Figure 4.14 only confirms that averaging $\widehat{\mathbf{R}}$ does not allow for any improvement.

Figure 4.15 provides a good insight regarding the RMSE performance. Here, we observe that the first signal is indeed calculated as the strongest one for both centralized and the fix9 topology. The average RMSE from (4.6) is displayed for Algorithm 2 in the Figure 4.16. The best performance shown in a.) with $\beta = 0.9$ corresponds to the dense connected network. For lower $\beta = 0.8, 0.4$, the performance of PAST and fully connected gets worse. This means that for strongly connected networks, a higher forgetting factor is more suitable. In contrast, for weak connected networks a lower β value allows for a better performance.

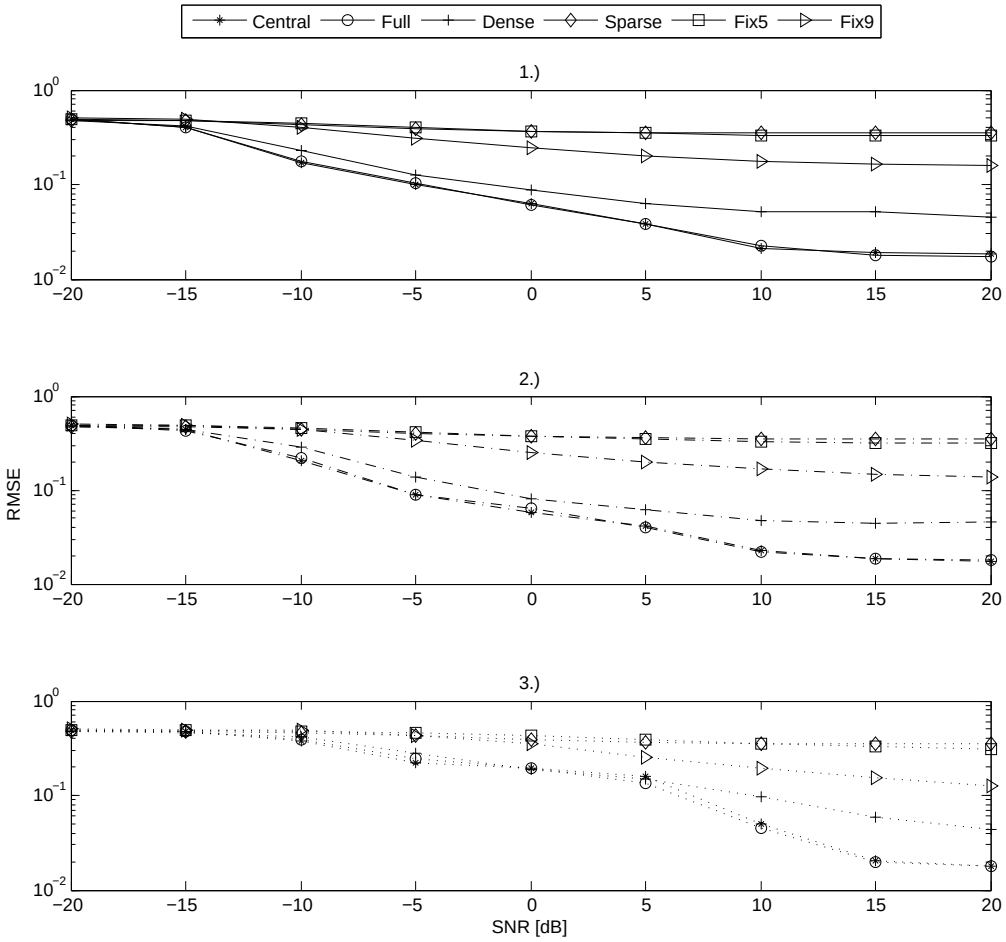

Figure 4.8: Evaluation of the Root Mean Square Error as in 4.6 when Algorithm 2 is evaluated for N=36 sensors and for all possible topologies from Figure 2.4. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

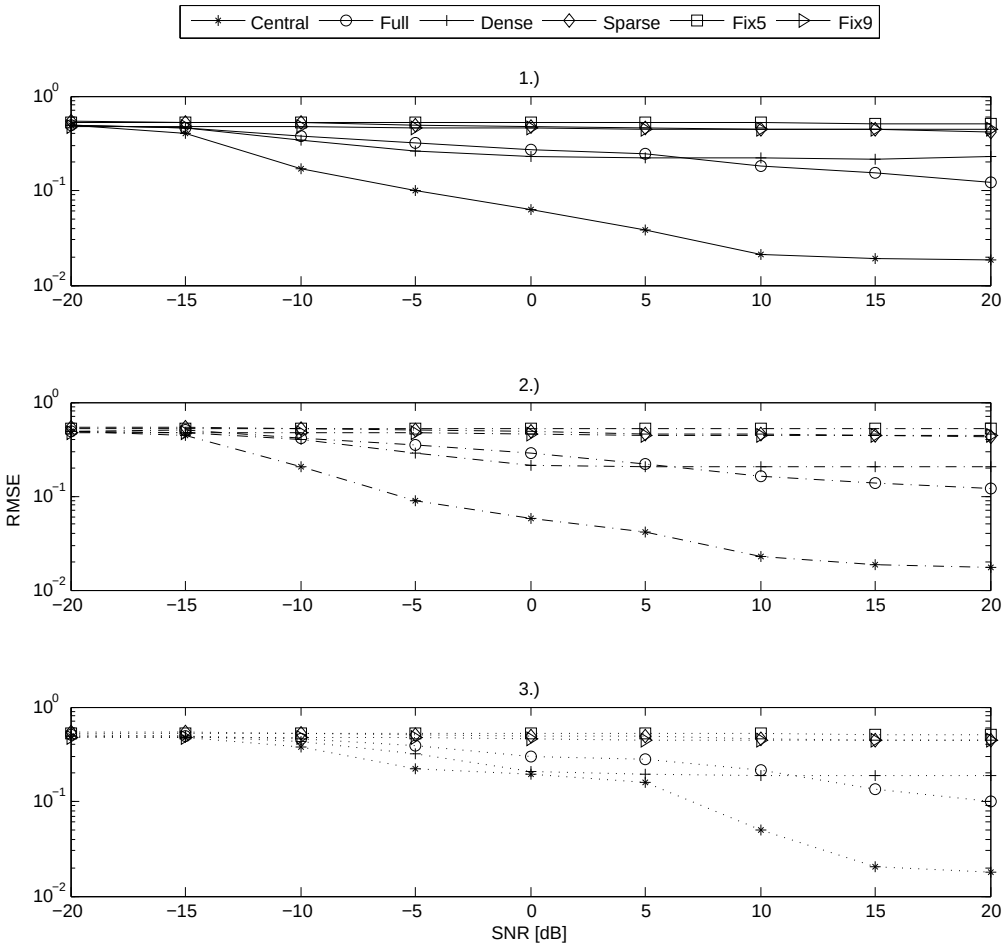

Figure 4.9: Evaluation of the Root Mean Square Error as in 4.6 when Algorithm 3 is evaluated for N=36 sensors and for all possible topologies from Figure 2.4. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

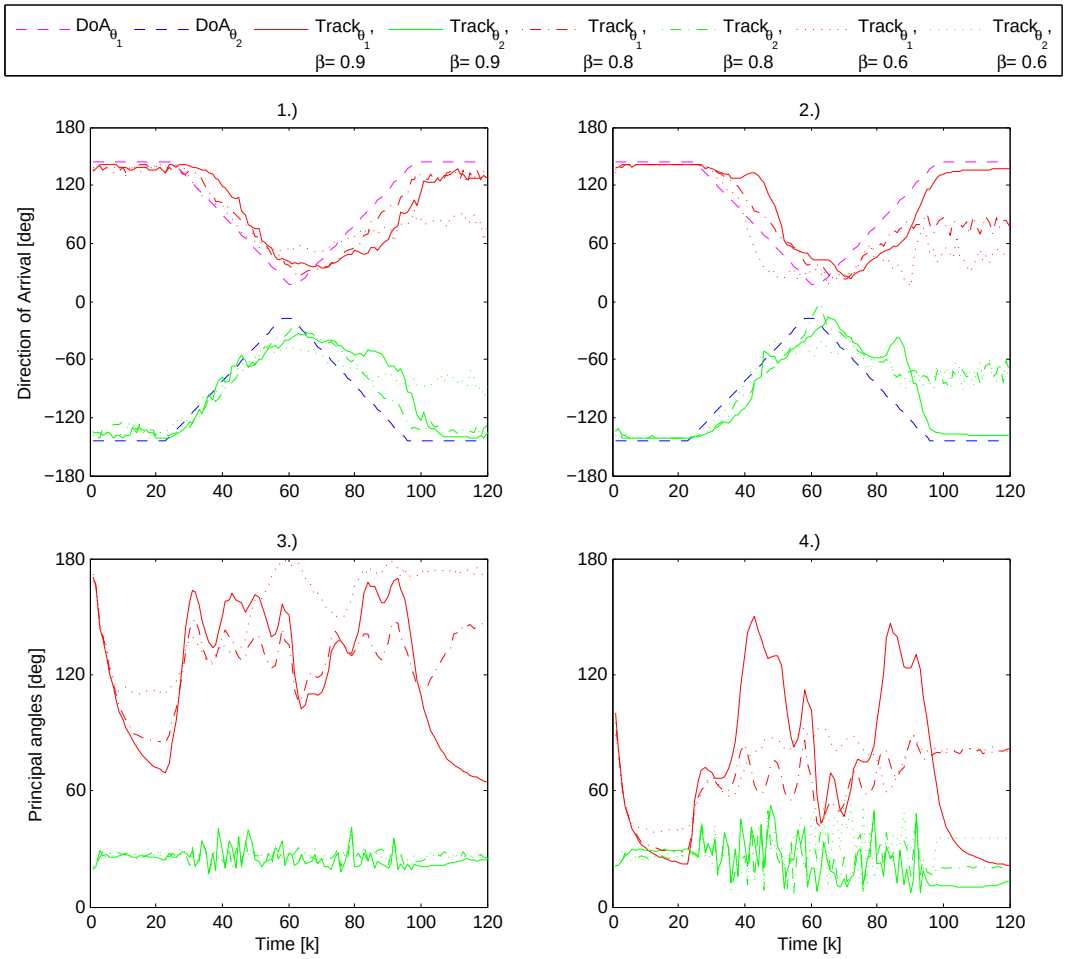

Figure 4.10: Subfigure 1.) and 2.) display the tracking capabilities of PAST. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

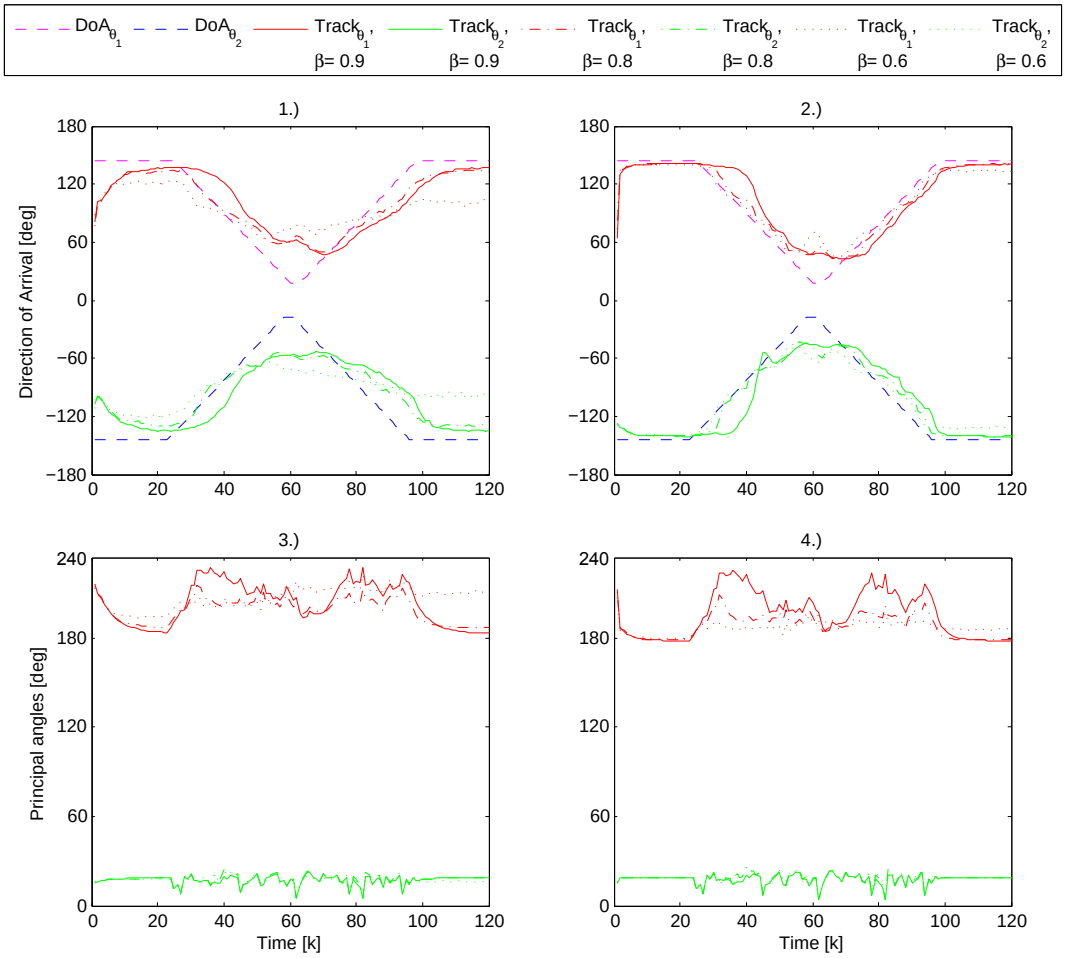

Figure 4.11: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 9$. This is the scenario e) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

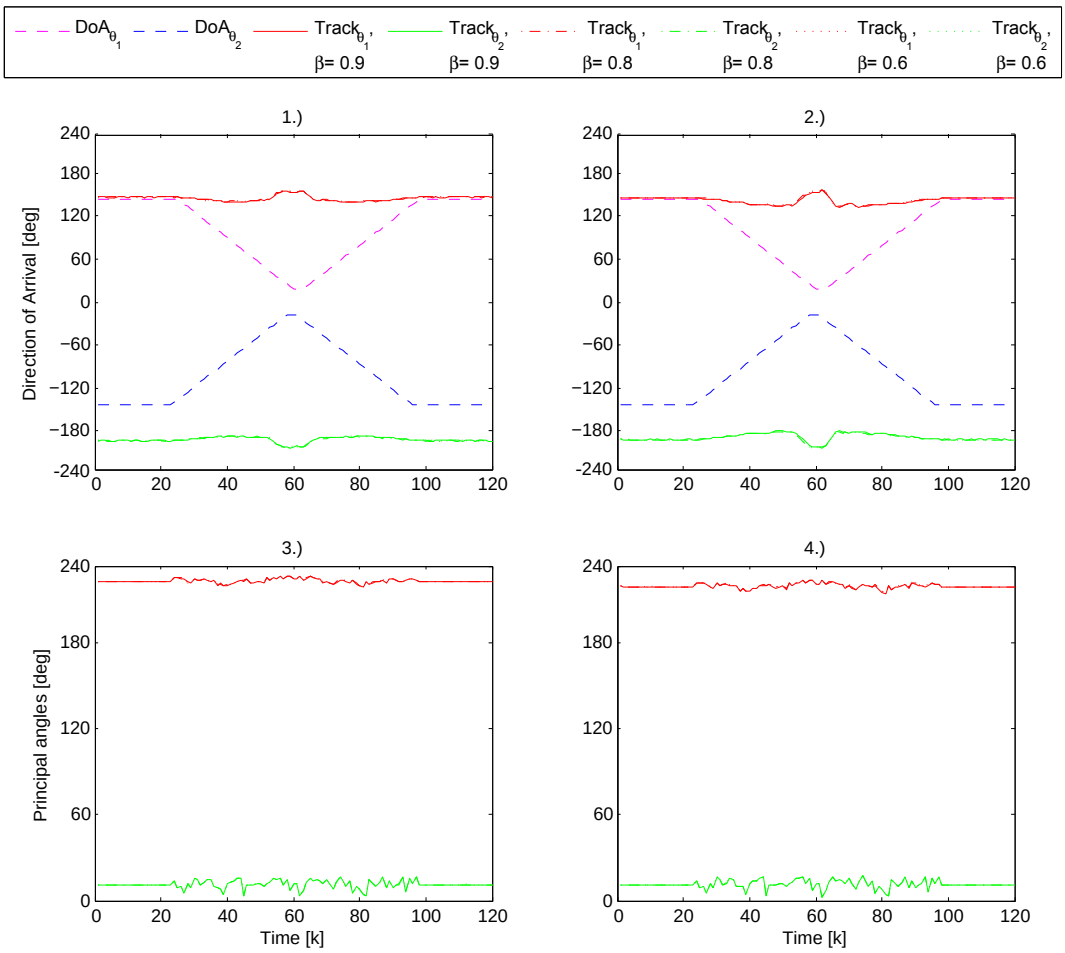

Figure 4.12: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\mathbf{R}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 9$. This is the scenario e) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR= -5dB for those left side subfigures and SNR= 5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

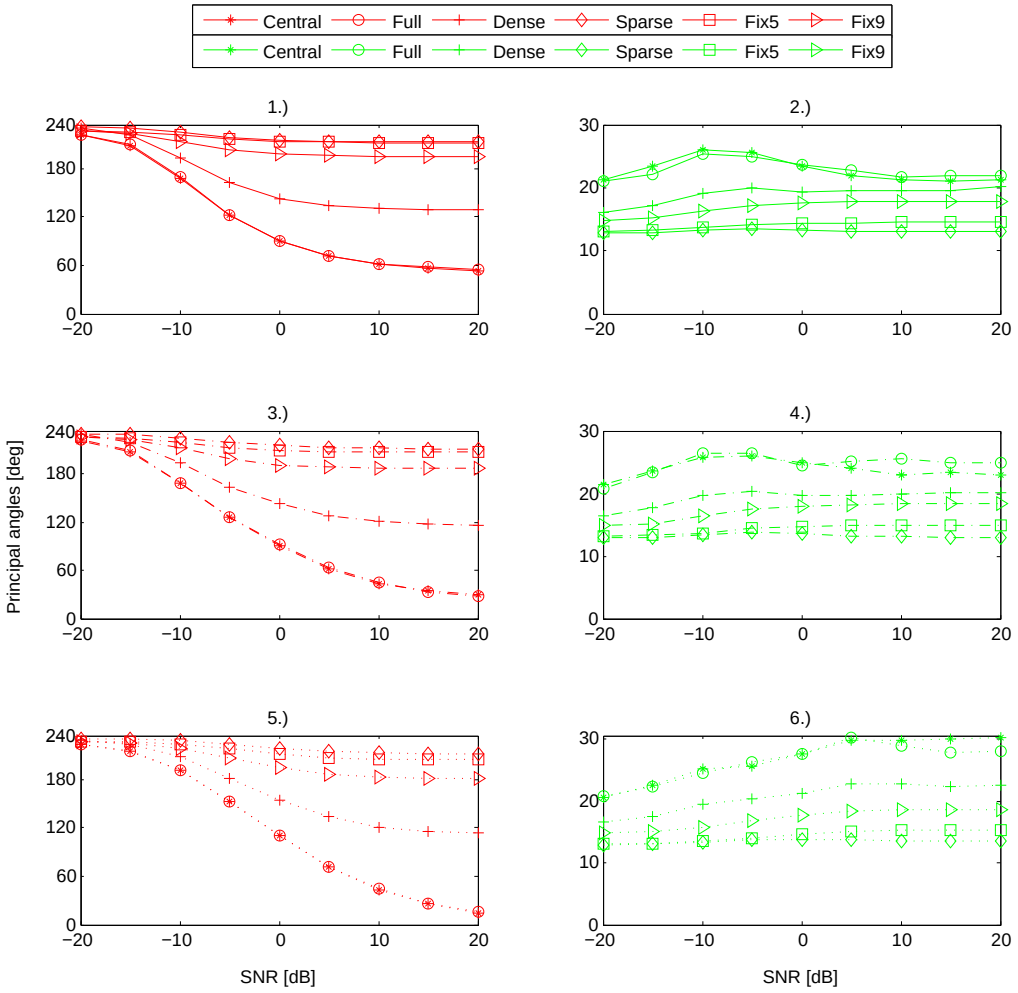

Figure 4.13: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the dynamic signals. The Algorithm 2 is analyzed for the topologies proposed in Figure 2.4. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

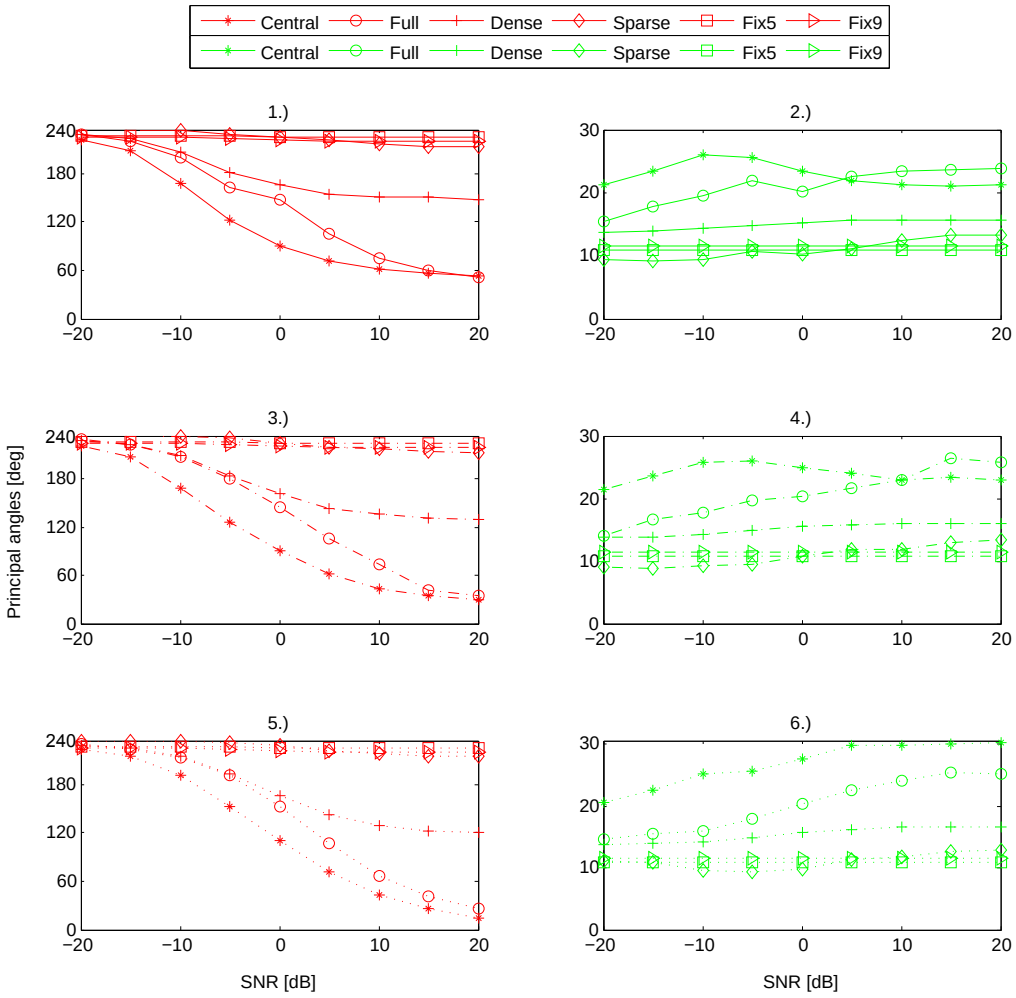

Figure 4.14: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the dynamic signals. The Algorithm 3 is analyzed for the topologies proposed in Figure 2.4. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

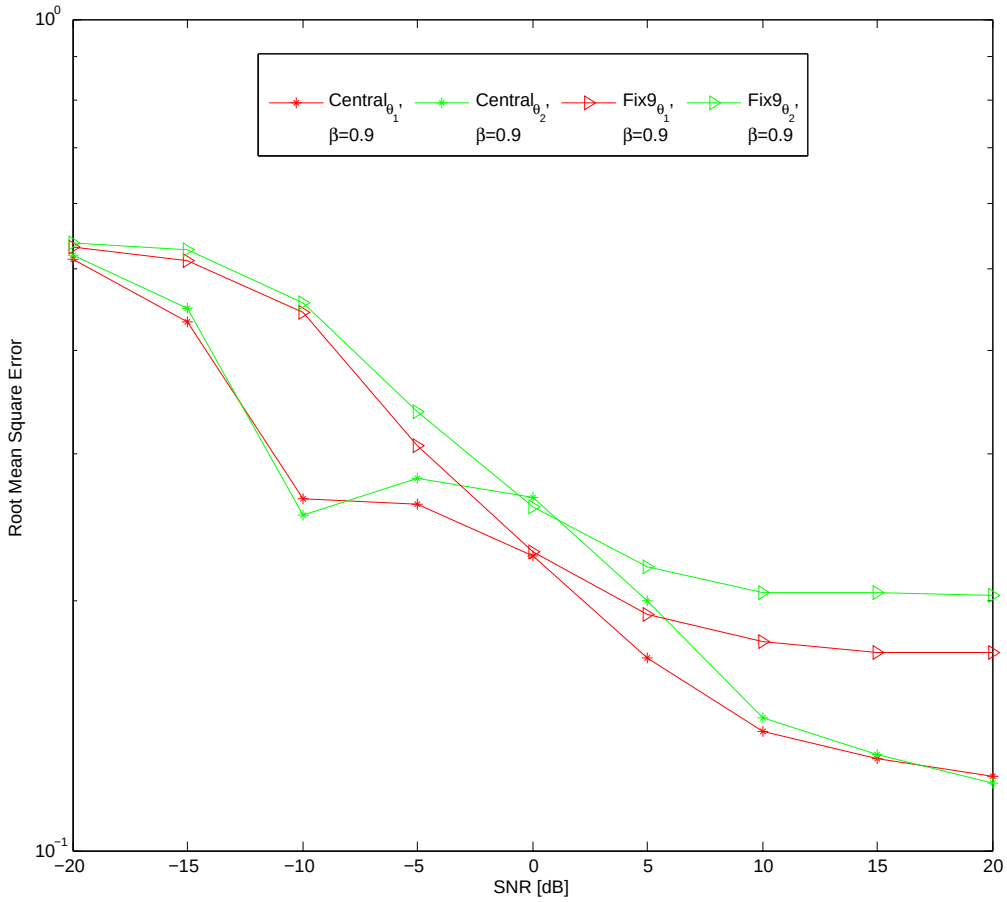

Figure 4.15: Evaluation of the Root Mean Square Error. The red lines correspond to $\widehat{\theta}_1$ and the green ones relate to $\widehat{\theta}_2$ from the crossing signals case. A performance comparison between PAST (depicted by the asterisks) and a scenario with constant neighborhood $|\mathcal{N}_i| = 9$ (triangles) using Algorithm 2 is presented.

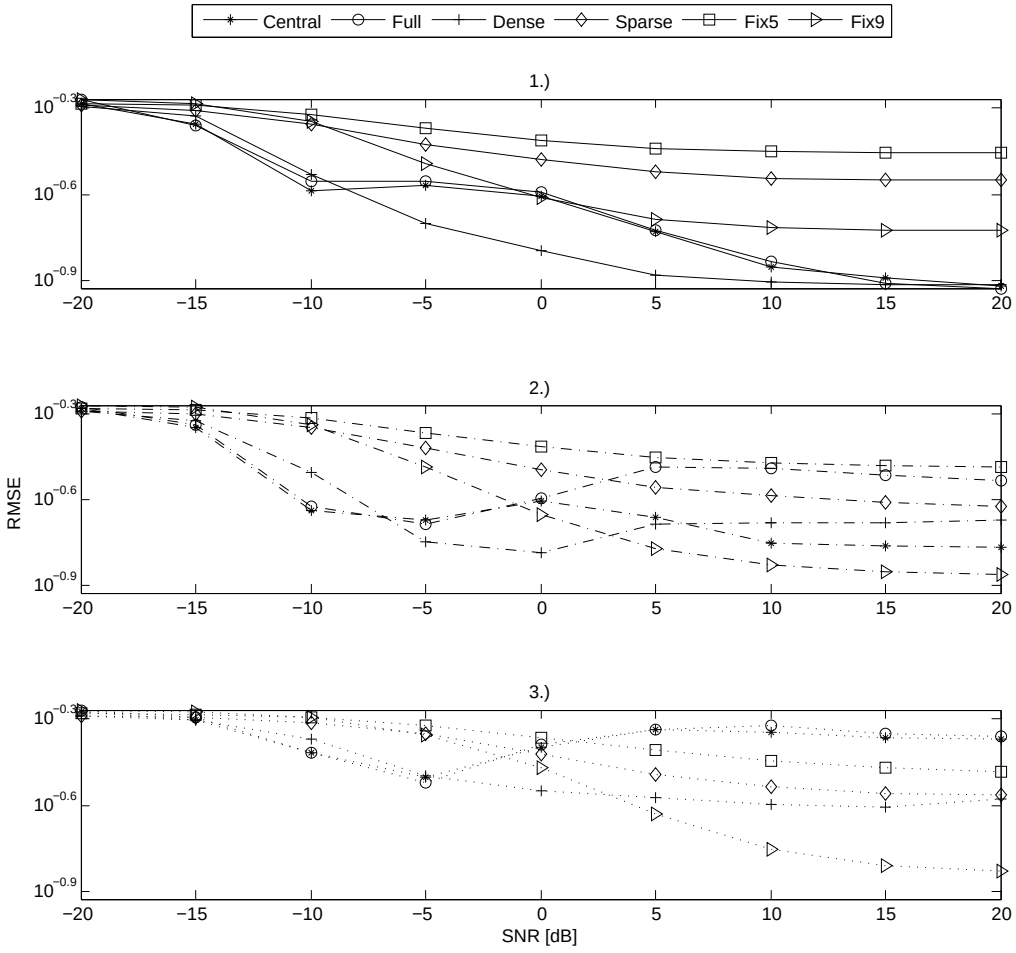

Figure 4.16: Evaluation of the Root Mean Square Error as in 4.6 when Algorithm 2 is evaluated for N=36 sensors and for all possible topologies from Figure 2.4. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

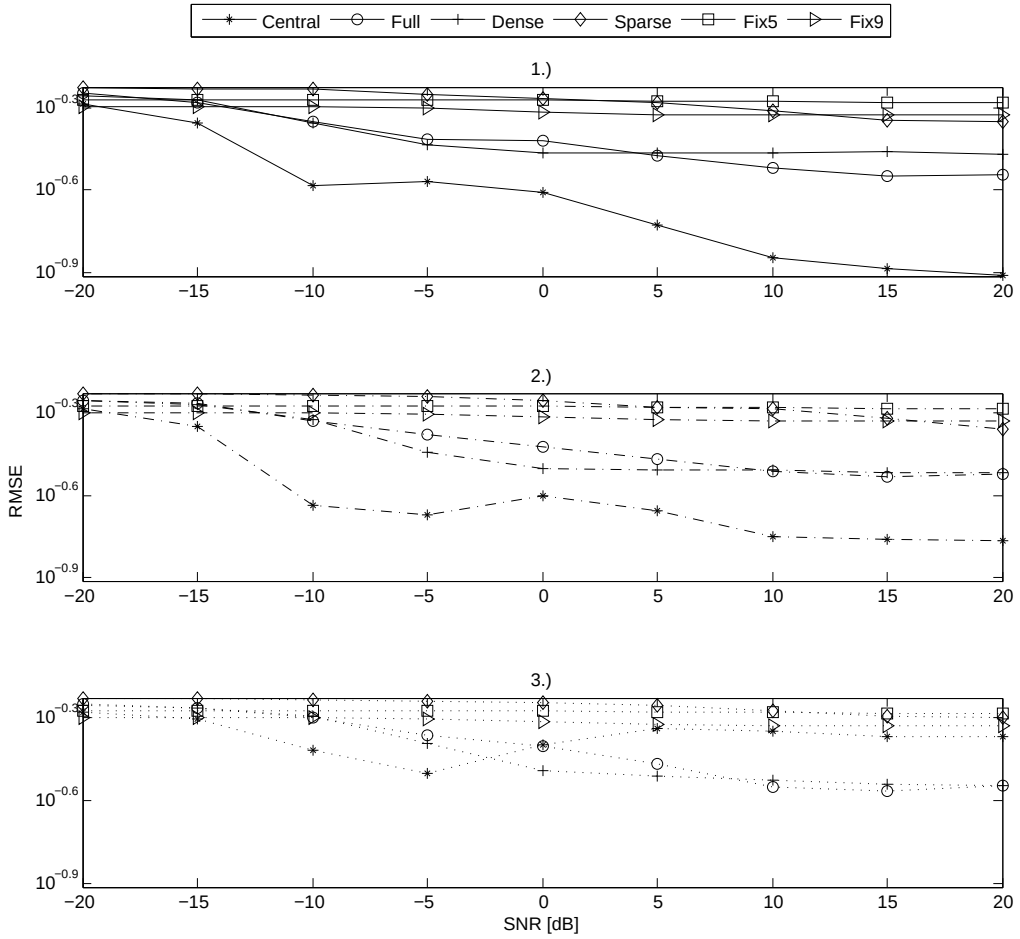

Figure 4.17: Evaluation of the Root Mean Square Error as in 4.6 when Algorithm 3 is evaluated for N=36 sensors and for all possible topologies from Figure 2.4. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

4.4 Remarks

This chapter evaluated the performance of both distributed Algorithm 2 and Algorithm 3 in terms of their DOA tracking capabilities, Principal Angles Between subspaces and Root Mean Square Error.

As expected, the network size and the connectivity influenced the algorithmic performance in either of our proposed algorithms. In a strongly connected scenario, the large amount of information exchanged allows to obtain very accurate estimates. Simulations were first performed in a network of size $N = 36$. The best Direction Of Arrival estimates given by a weak connected network were obtained with a regular topology, i.e. $\mathcal{N}_i = 9$, that employs the Algorithm 2, where the local signal update vector $\underline{\mathbf{y}}$ was averaged within the network. Algorithm 3 does not offer a good output for weakly connected networks. Moreover, the experiment with smaller network size $N = 12$ presents a slightly worse behavior. Throughout the experiments, we observed that the tracking capabilities of Algorithm 2 are by far better than those from Algorithm 3. A possible explanation is that the former approach employs the Average Consensus at the very beginning of the calculations, where no extra nuisance (i.e. thermal noise) is introduced.

MUSIC is a subspace-based DOA detection algorithm that exploits the orthogonality between the signal and noise subspaces in order to find specific parameters of interest. However, one of its major defects lies in the fact that its performance deteriorates when signals are too close to each other. Here, the forgetting factor gains an important role. With large β values, previous estimates have more influence on future estimates (as expected in a recursive algorithm). Although low betas "forget" faster, the noise and the averaging over all nodes affect the tracking performance of the algorithm. A trade-off between adaptation time and accuracy must be considered. For that reason, the correct selection of these forgetting factors are very important in algorithms like PAST, since they work in a recursive subspace-based fashion.

Another issue encountered for the correct RMSE analysis was that even though MUSIC finds the correct direction of the signal sources, it is not capable of distinguishing between each of them. This is not an issue in the tracking, as the signals are only "swapped". However, this introduces a bias when the Root Mean Square Error is calculated, since the the definition is given by the comparison of the "true" value and its estimate, see (4.4). To tackle this problem, a signal identification algorithm is needed in order to separate the signals in a correct way.

Chapter 5

Convergence properties

Additionally, a mathematical convergence analysis of Algorithm 2 and Algorithm 3 is provided by means of Singular Value Decomposition methods. The convergence properties in the mean and mean square sense, as well as step-size bounds guaranteeing the stability of the algorithms are presented in the last part of this chapter. Finally, the convergence behavior of both algorithmic approaches is corroborated through simulation experiments.

5.1 Convergence analysis

Since its introduction by Bin Yang in 1993, the Projection Approximation Subspace Tracking (PAST) algorithm [8, 46] and its many derivatives have become quite popular as relatively simple algorithms to detect subspaces, separate them and even track them. The original analysis of the algorithm's behavior [47, 48] was based on an Ordinary Differential Equation (ODE) approach with all its advantagaes and drawbacks. In the ODE framework iterative approaches are interpreted as differential equations of continuous functions based on some Lyapunov arguments, so it can be deduced whether step-sizes exist for which the algorithm converges.

With such an approach, a relatively large amount of assumptions have to be made in order to make it work. This is one of its larger drawbacks as it often remains unclear whether all such assumptions can be satisfied in practice. The second drawback is that a practical (upper) bound on the step-size is hard to deduce formally, leaving it to experiments to find out which step-sizes work, and which do not. These issues motivated our work in [80], where a different analysis approach of the PAST algorithm, based on Singular Value Decomposition (SVD), is proposed. This method allows to remove most of the assumptions mentioned in

[48] and to find practical step-size bounds for $\gamma(k)$, which yield to a new insight in the algorithmic behavior. In the following pages, we provide an overview of our work introduced in [80].

The PAST algorithm minimizes the cost function in (5.32). Contained in $\underline{\mathbf{x}} \in \mathbb{C}^{N\times 1}$ is the data observed at time k, and $\mathbf{W} \in \mathbb{C}^{N\times r}$ is the signal subspace containing r narrow-band signal waves impinging an array of N number of sensors, hidden among additive noise. Starting with initial values $\widehat{\mathbf{R}}_0^{yy} = \mathbf{I}$ and $\mathbf{W}_0 \in \mathbb{C}^{N\times r} = [\mathbf{I}_r, \mathbf{0}]^T$ a recursive algorithm is applied on a sequence of vectors $\underline{\mathbf{x}}(k)$ in order to minimize the corresponding Least-Squares (LS) cost function of (2.36) on continuously incoming observations $\underline{\mathbf{x}}(k)$:

$$\underline{\mathbf{y}}(k) = \mathbf{W}^H(k-1)\underline{\mathbf{x}}(k) \tag{5.1}$$

$$\underline{\mathbf{e}}(k) = \underline{\mathbf{x}}(k) - \mathbf{W}(k-1)\mathbf{W}^H(k-1)\underline{\mathbf{x}}(k) \tag{5.2}$$

$$\widehat{\mathbf{R}}^{yy}(k) = \widehat{\mathbf{R}}^{yy}(k-1) + \beta(k)\left[\underline{\mathbf{y}}(k)\underline{\mathbf{y}}^H(k) - \widehat{\mathbf{R}}^{yy}(k-1)\right] \tag{5.3}$$

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \gamma(k)\underline{\mathbf{e}}(k)\underline{\mathbf{y}}^H(k)\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}. \tag{5.4}$$

The so-called subspace update equation in (5.4) applies the inverse of the estimated autocorrelation matrix $\widehat{\mathbf{R}}^{yy}(k) = E\left\{\underline{\mathbf{y}}(k)\underline{\mathbf{y}}(k)^H\right\}$. This is typically achieved by the matrix inversion lemma to reduce the computational complexity. Other fast variants are possible to derive and have been proposed [81] but are not the scope of this study. We restrict to a relatively general form of the algorithm with two free parameters $\beta(k)$ and $\gamma(k)$. The primer is bounded by $0 < \beta(k) \leq 1$ and acts as forgetting factor for the decaying memory estimation of the autocorrelation matrix $\widehat{\mathbf{R}}^{yy}(k)$. A value of β close to one gives more emphasis to the most recent rank-one update $\underline{\mathbf{y}}(k)\underline{\mathbf{y}}(k)^H$ and leads to a faster decaying memory. A value close to zero, in turn, results in only slight changes of $\widehat{\mathbf{R}}^{yy}(k)$. While the optimal choice of $\beta(k)$ depends on the tracking problem, the choice of $\gamma(k)$ directly affects the convergence behavior of the algorithm which gives reason to consider it more thoroughly.

5.1.1 First order analysis of PAST

The main goal here is transform the signal subspace matrix $\mathbf{W}(k)$ into a unitary matrix, so that $\lim_{k\to\infty} \mathbf{W}(k)^H\mathbf{W}(k) = \mathbf{I}_r$. In a first order analysis, we study the algorithmic behavior in the mean. We assume the sensor data $\mathbf{x}(k)$ to be of random nature with an autocorrelation matrix $\widehat{\mathbf{R}}^{xx}(k) = E\left\{\mathbf{x}(k)\mathbf{x}(k)^H\right\}$.

Assumptions: Let us examine the estimated autocorrelation matrix $\widehat{\mathbf{R}}^{yy}(k)$. Since this matrix is being averaged over time as shown in (5.3), it is assumed to be ergodic. More specifically, the ensemble average $\mathbf{R}^{yy}(k)$ of the given process is determined from a time average $\widehat{\mathbf{R}}^{yy}(k)$ over past samples. As a consequence of its eigendecomposition $\mathbf{R}^{yy}(k) = E\left\{\widehat{\mathbf{R}}^{yy}(k)\right\} = \mathbf{V}_1\mathbf{\Lambda}^{yy}(k)\mathbf{V}_1^H$, where $\mathbf{\Lambda}^{yy}(k)$

contains the time-variant eigenvalues of $\mathbf{R}^{yy}(k)$. After some substitutions and allowing $\mathbf{W}(k-1)$ and $\mathbf{x}(k)$ be statistically independent, it is possible to find that the eigendecomposition of the time-averaged $\widehat{\mathbf{R}}^{yy}(k)$ is

$$\mathbf{R}^{yy}(k) = E\left\{\widehat{\mathbf{R}}^{yy}(k)\right\} = E\left\{\mathbf{y}(k)\mathbf{y}(k)^H\right\} = E\left\{\mathbf{W}(k-1)^H\mathbf{x}(k)\mathbf{x}(k)^H\mathbf{W}(k-1)\right\} \tag{5.5}$$

$$\begin{aligned} &= E\left\{\mathbf{W}(k-1)^H\mathbf{R}^{xx}(k)\mathbf{W}(k-1)\right\} \\ &= \left(\mathbf{U}_1\bar{\mathbf{\Sigma}}(k-1)\mathbf{V}_1^H\right)^H\left(\mathbf{U}_1\mathbf{\Lambda}^{xx}(k)\mathbf{U}_1^H\right)\left(\mathbf{U}_1\bar{\mathbf{\Sigma}}(k-1)\mathbf{V}_1^H\right) \\ &= \mathbf{V}_1\bar{\mathbf{\Sigma}}(k)\mathbf{\Lambda}^{xx}(k)\bar{\mathbf{\Sigma}}(k)\mathbf{V}_1^H. \end{aligned} \tag{5.6}$$

Here we emphasize a subtle but very important difference: the autocorrelation matrix $\widehat{\mathbf{R}}^{yy}(k)$ in (5.3) is a time averaged matrix whose decomposition leads to likewise temporally-averaged singular values $\bar{\mathbf{\Sigma}}(k)$. These singular values should not be confused with the instantaneous singular values $\mathbf{\Sigma}(k)$ originating from decomposing $E\{\mathbf{W}(k-1)\}$. The mean of (5.4) is thus written as

$$\begin{aligned} E\{\mathbf{W}(k)\} &= E\{\mathbf{W}(k-1)\} \\ &+ \gamma(k)E\left\{(\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}(k-1)^H)\mathbf{x}(k)\mathbf{x}(k)^H\mathbf{W}(k-1)\right\} \\ &\quad \times E\left\{\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}\right\}. \end{aligned} \tag{5.7}$$

Applying the expectation with respect to the random process $\mathbf{x}(k)$ leads to

$$\begin{aligned} E\{\mathbf{W}(k)\} &= E\{\mathbf{W}(k-1)\} + \gamma(k)E\left\{\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}(k-1)^H\right\} \\ &\quad \times \mathbf{R}^{xx}(k)E\{\mathbf{W}(k-1)\}E\left\{\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}\right\}. \end{aligned} \tag{5.8}$$

Since the decomposition of the autocorrelation matrix $\mathbf{R}^{xx}(k) = \mathbf{Q}\mathbf{\Lambda}^{xx}(k)\mathbf{Q}^H$, we recognize that a steady-state solution only exists if $E\{\mathbf{W}(k)\} = [\mathbf{U}_1, \mathbf{U}_2]\,[\mathbf{\Sigma}(k), O]^T\,[\mathbf{V}_1\mathbf{V}_2]^H$ with $\mathbf{U}_1 = \mathbf{Q}$ where we applied a SVD on $\mathbf{W}$ and partitioned it into the significant part $\mathbf{U}_1\mathbf{\Sigma}(k)\mathbf{V}_1^H$ and a zero block. With such findings it is advantageous to rewrite (5.8) into its SVD components. Here it becomes evident, that all terms are comprised of the same left matrix $\mathbf{U}_1$ and the right matrix $\mathbf{V}_1^H$. After some factorization, all terms in the center become diagonal.

$$\begin{aligned} \mathbf{U}_1\mathbf{\Sigma}(k)\mathbf{V}_1^H &= \mathbf{U}_1\mathbf{\Sigma}(k-1)\mathbf{V}_1^H + \gamma(k)\left(\mathbf{I}-\mathbf{U}_1\mathbf{\Sigma}(k-1)\mathbf{V}_1^H\mathbf{V}_1\mathbf{\Sigma}(k-1)\mathbf{U}_1^H\right) \\ &\quad \times \mathbf{U}_1\mathbf{\Lambda}^{xx}(k)\mathbf{U}_1^H\mathbf{U}_1\mathbf{\Sigma}(k-1)\mathbf{V}_1^H\left[\mathbf{V}_1\left(\bar{\mathbf{\Sigma}}(k-1)\mathbf{\Lambda}^{xx}(k)\bar{\mathbf{\Sigma}}(k-1)\right)^{-1}\mathbf{V}_1^H\right] \\ &= \mathbf{U}_1\mathbf{\Sigma}(k-1)\mathbf{V}_1^H + \gamma(k)\left(\mathbf{U}_1 - \mathbf{U}_1\mathbf{\Sigma}(k-1)\mathbf{\Sigma}^H(k-1)\right) \\ &\quad \times \mathbf{\Lambda}^{xx}(k)\mathbf{\Sigma}(k-1)\left(\bar{\mathbf{\Sigma}}(k-1)\mathbf{\Lambda}^{xx}(k)\bar{\mathbf{\Sigma}}(k-1)\right)^{-1}\mathbf{V}_1^H \\ &= \mathbf{U}_1[\mathbf{\Sigma}(k-1) + \gamma(k)\left(\mathbf{I}-\mathbf{\Sigma}(k-1)\mathbf{\Sigma}^H(k-1)\right) \end{aligned} \tag{5.9}$$

$$\times \mathbf{\Lambda}^{xx}(k)\mathbf{\Sigma}(k-1)\left(\bar{\mathbf{\Sigma}}(k-1)\mathbf{\Lambda}^{xx}(k)\bar{\mathbf{\Sigma}}(k-1)\right)^{-1}]\mathbf{V}_1^H$$
$$\mathbf{\Sigma}(k) = \mathbf{\Sigma}(k-1) + \gamma(k)\left(\mathbf{I} - \mathbf{\Sigma}(k-1)\mathbf{\Sigma}(k-1)\right)$$
$$\times \mathbf{\Lambda}^{xx}(k)\mathbf{\Sigma}(k-1)\left(\bar{\mathbf{\Sigma}}(k)\mathbf{\Lambda}^{xx}(k)\bar{\mathbf{\Sigma}}(k)\right)^{-1}. \tag{5.10}$$

As a result, it becomes feasible to express (5.10) only by its diagonal terms as

$$\sigma^l(k) = \sigma^l(k-1) + \gamma(k)\frac{\left(1-\left(\sigma^l(k-1)\right)^2\right)\sigma^l(k-1)}{\left(\bar{\sigma}^l(k)\right)^2}. \qquad l = 1,\ldots,r \tag{5.11}$$

Although being a complicated term in $\sigma^l(k-1)$, we recognize that the actual values $\lambda_i^{xx}(k)$ in the autocorrelation matrix $\mathbf{\Lambda}^{xx}(k)$, namely the power contributions of the observed sensor signals, are irrelevant to the algorithm. Knowing that the steady state solution is given if all signals are perfectly decorrelated, that is when all singular values are one, we aim to have $\sigma^l(k-1) \to 1$. We can rewrite this for $l = 1, 2, ..., r$ as

$$1-\sigma^l(k) = (1-\sigma^l(k-1))\left[1-\gamma(k)\frac{(1+\sigma^l(k-1))\sigma^l(k-1)}{\left(\bar{\sigma}^l(k)\right)^2}\right]. \qquad l = 1,\ldots,r \tag{5.12}$$

As all singular values are positive, we have thus proven the following theorem.

Theorem 1.1. *The PAST algorithm converges in the mean for a sufficiently small step-size $\gamma(k) > 0$, if $\frac{(1+\sigma^l(k-1))\sigma^l(k-1)}{\left(\bar{\sigma}^l(k)\right)^2}$ is bounded.*

As the right hand side term in (5.12) can take on arbitrary values in the range $[0, \infty)$, it is difficult to bound the step-size at this point. We recognize, that in particular small singular values are decisive. Note, that similar forms of (5.8) have been analyzed in [82] in the context of blind source separation and [83] as a means to compute robustly matrix inverses. Furthermore, in his article [8] Bin Yang also proposed a simpler gradient term algorithm, that is simply omitting the matrix inverse of $\widehat{\mathbf{R}}^{yy}(k)$. The analysis method presented here can be applied to such an algorithm, revealing now that the step-size $\gamma(k)$ depends strongly on the eigenvalues $\lambda_i^{xx}(k)$ of the observation process $\mathbf{x}(k)$.

5.1.2 Anaylsis of step-size bounds

Let us return to Equation (5.12). We learn here that

$$0 < \gamma(k) < \frac{2\left(\bar{\sigma}^l(k)\right)^2}{(1+\sigma^l(k-1))\sigma^l(k-1)}; \qquad l = 1, ..., r, \tag{5.13}$$

and we assume that it has an upper bound

$$0 < \gamma(k) < \frac{2\bar{\sigma}^2_{\min}(k)}{(1+\sigma_{\max}(k-1))\sigma_{\max}(k-1)}. \tag{5.14}$$

If we use the relation $0.25 + 2x^2 \geq x + x^2$ for $x \geq 0$ we obtain an even lower bound

$$0 < \gamma(k) < \frac{2\bar{\sigma}^2_{\min}(k)}{0.25 + 2\sigma^2_{\max}(k-1)}. \tag{5.15}$$

The term $\sigma^2_{\max}(k-1)$ is related to $\mathbf{W}(k-1)^H\mathbf{W}(k-1)$. If $\lambda^{xx}(k)$ and $\mathbf{v}(k)$ are an eigenvalue and eigenvector pair of the matrix $\mathbf{W}(k)^H\mathbf{W}(k)$, the following condition holds true:

$$\begin{aligned} |\lambda^{xx}(k)|^m\|\mathbf{v}(k)\| &= \|(\lambda^{xx})^m(k)\mathbf{v}(k)\| \\ &= \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\mathbf{v}(k)\| \\ &\leq \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\| \cdot \|\mathbf{v}(k)\| \\ &\Longrightarrow |\lambda^{xx}(k)| \leq \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\|^{\frac{1}{m}}. \end{aligned} \tag{5.16}$$

Thus, for $m = 1$ we obtain the simplified but practically feasible lower bound

$$0 < \gamma(k) < \frac{2\bar{\sigma}^2_{\min}(k)}{0.25 + 2\|\mathbf{W}(k-1)^H\mathbf{W}(k-1)\|} = \gamma_{\max}(k)\bar{\sigma}^2_{\min}(k). \tag{5.17}$$

The smallest singular value $\bar{\sigma}_{\min}(k)$ is thus decisive for convergence and a step-size $\gamma(k)$ needs to be selected based on its knowledge. In the classic PAST algorithm such knowledge is not present and thus only very small step-sizes can be selected in the hope to have sufficiently large values of $\bar{\sigma}_{\min}(k)$, which in turn results in slow convergence and poor tracking.

As $\bar{\sigma}_{\min}(k)$ is typically not available, we may estimate it. Nowadays, low-complexity methods are available to estimate the smallest singular value [84, 85]. Such estimation technique however, can easily lead to too small step-sizes, resulting in a very slow convergence as well as poor tracking. It is thus of further interest to modify the algorithm in such a way that the dependency on the smallest singular value disappears. In order to prevent the undesired behavior of the classic PAST algorithm of not offering a feasible step-size bound, we are proposing to alter the update equation (5.4) into a generic update

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \gamma(k)\mathbf{e}(k)\mathbf{y}(k)^H\mathbf{B}, \tag{5.18}$$

with the following options:

$$\text{PAST-I:} \qquad \mathbf{B} = \left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}\bar{\sigma}^2_{\min}(k) \tag{5.19}$$

$$\text{PAST-II:} \quad \mathbf{B} = \left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1} \mathbf{R}_{\mathbf{W}}(k) \tag{5.20}$$

$$\text{PAST-III:} \quad \mathbf{B} = [\widehat{\mathbf{R}}^{yy}(k) + \alpha \mathbf{I}_r]^{-1} \tag{5.21}$$

$$\text{PAST-IV:} \quad \mathbf{B} = \mathbf{I} \tag{5.22}$$

for some small but positive $\alpha > 0$. Such regularization in PAST-III is not required for computing the inverse of $\widehat{\mathbf{R}}^{yy}(k)$, but prevents the smallest singular value to have a decisive impact on the stability. PAST-IV is the gradient-type version of the PAST algorithm [8]. In PAST-II an average of $\mathbf{W}(k-1)^H\mathbf{W}(k-1)$ is computed by $\mathbf{R}_{\mathbf{W}}(k) = \mathbf{R}_{\mathbf{W}}(k-1) + \beta(k)\left(\mathbf{W}(k-1)^H\mathbf{W}(k-1) - \mathbf{R}_{\mathbf{W}}(k-1)\right)$ and applied to compensate for the inverse singular values of $\widehat{\mathbf{R}}^{yy}(k)$, as we would find $\mathbf{R}_{\mathbf{W}}(k) = \mathbf{U}\bar{\mathbf{\Sigma}}^2(k)\mathbf{U}^H$. A simpler version of this is PAST-I where we assume knowledge of the smallest singular value only, for example by simple tracking algorithms [85].

Applying the same analysis technique as before provides the next bounds

$$0 < \gamma_{\mathrm{I}}(k) < \gamma_{\max}(k) \leq \min_l \frac{2\left(\bar{\sigma}^l(k)\right)^2}{(1+\sigma^l(k-1))\sigma^l(k-1)\bar{\sigma}^2_{\min}(k)} \tag{5.23}$$

$$0 < \gamma_{\mathrm{II}}(k) < \gamma_{\max}(k) \leq \min_l \frac{2}{(1+\sigma^l(k-1))\,\sigma^l(k-1)} \tag{5.24}$$

$$0 < \gamma_{\mathrm{III}}(k) < \gamma_{\max}(k)\delta_{\min} \leq \min_l \frac{2\left(\left(\bar{\sigma}^l(k)\right)^2 + \delta\right)}{(1+\sigma^l(k-1))\,\sigma^l(k-1)} \tag{5.25}$$

$$0 < \gamma_{\mathrm{IV}}(k) < \frac{\gamma_{\max}(k)}{\mathrm{tr}[\mathbf{R}^{xx}(k)]} \leq \min_l \frac{2}{(1+\sigma^l(k-1))\,\sigma^l(k-1)\lambda^{l,xx}(k)} \tag{5.26}$$

with $\delta = \alpha/\lambda^{l,xx}(k)$ and $\delta_{\min} = \alpha/\lambda^{xx}_{\max}(k)$. Thus, the knowledge of $\delta_{\min}$ is sufficient to provide a conservative step-size bound. Indirectly, the choice of α now also determines the convergence speed; larger values typically offering higher speed. Bounds for (5.24) and (5.26) can also be derived, following the approach explained at the beginning of this section. If in (5.26) $\lambda^{l,xx}(k)$ is not known, it may be feasible to replace it by $\mathrm{tr}(\mathbf{R}^{xx}(k))$. Now it only depends on the matrix norm to compute a safe upper bound of $\gamma(k)$ which is a feasible operation. In practice, it turns out that a matrix one norm provides tight results. The upper bounds define some time-variant maximal value $\gamma_{\max}(k)$ and we select fractions $\beta\gamma_{\max}(k)$, where $\beta \in [0,1]$. Often the PAST algorithm is being run with a step-size $\gamma(k) = 1/k$ or $\gamma(k) = 1/[k+1]$. In these cases, after a few iterations, the step-size satisfies the stability condition and the algorithm converges. However, the price for this convergence is a lack of tracking capability as the step-size becomes so small that the algorithm cannot adjust to a new situation anymore.

5.1.3 Second order analysis of PAST

In particular for adaptive filters, we are also interested in analyzing the second order moment describing the quantitative behavior of the algorithm. We have investigated the evolution of terms of the form $\mathbf{W}(k)^H\mathbf{W}(k)$ from the update equation (5.4) and found out, that the result is very similar to that in (5.12) for the first order moment. The main difference relies on a term that appears equally for all diagonal entries but does not influence the stability of the system. On the other hand, it does influence the values for the step size $\gamma(k)$.

$$\begin{aligned}
E\left\{\mathbf{W}^H(k)\mathbf{W}(k)\right\} = {} & E\left\{\mathbf{W}^H(k-1)\mathbf{W}(k-1)\right\} + 2\gamma(k) \\
& \times E\left\{\mathbf{W}^H(k-1)\left[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}^H(k-1)\right]\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\right\} \\
& \times E\Big\{\mathbf{W}(k-1)\left[\mathbf{W}^H(k-1)\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\mathbf{W}(k-1)\right]^{-1}\Big\} + \gamma^2(k) \\
& \times E\Big\{\left[\mathbf{W}^H(k-1)\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\mathbf{W}(k-1)\right]^{-1}\mathbf{W}^H(k-1)\Big\} \\
& \times E\Big\{\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\left[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}^H(k-1)\right]^2\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\Big\} \\
& \times E\Big\{\mathbf{W}(k-1)\left[\mathbf{W}^H(k-1)\underline{\mathbf{x}}(k)\underline{\mathbf{x}}^H(k)\mathbf{W}(k-1)\right]^{-1}\Big\}. \qquad (5.27)
\end{aligned}$$

Reformulating the update equation in such second order terms results in one particularly difficult term: $E\{\underline{\mathbf{x}}(k)\underline{\mathbf{x}}(k)^H[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}(k-1)^H]^2\underline{\mathbf{x}}(k)\underline{\mathbf{x}}(k)^H\}$. For this term to compute we need to find fourth order moments of the vector process $\underline{\mathbf{x}}_i(k)$. Note however that such fourth order moments only modify the terms in $\gamma_i^2(k)$, thus replacing them by the square of second order moments is a good approximation. For instance, assume $\mathbf{x}_i(k)$ to be of complex Gaussian nature, the term can be given as

$$\begin{aligned}
E\{\underline{\mathbf{x}}(k)\underline{\mathbf{x}}(k)^H[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}(k-1)^H]^2\underline{\mathbf{x}}(k)\underline{\mathbf{x}}(k)^H\} = {} & \mathbf{R}^{xx}(k)[\mathbf{I}-\mathbf{W}(k-1) \\
& \times \mathbf{W}(k-1)^H]^2\mathbf{R}^{xx}(k) + \mathbf{R}^{xx}(k) \\
& \times \mathrm{tr}\left\{\mathbf{R}^{xx}(k)[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}(k-1)^H]^2\right\}.
\end{aligned}$$

and therefore

$$\begin{aligned}
\mathbf{W}^H(k)\mathbf{W}(k) = {} & \mathbf{W}^H(k-1)\mathbf{W}(k-1) + 2\gamma(k)\mathbf{W}^H(k-1)\left[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}^H(k-1)\right] \\
& \times \mathbf{R}^{xx}(k)\mathbf{W}(k-1)\left[\mathbf{W}^H(k-1)\mathbf{R}^{xx}(k)\mathbf{W}(k-1)\right]^{-1} \\
& + \gamma^2(k)\left[\mathbf{W}^H(k-1)\mathbf{R}^{xx}(k)\mathbf{W}(k-1)\right]^{-1}\mathbf{W}^H(k-1)\mathbf{R}^{xx}(k) \\
& \times \left[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}^H(k-1)\right]^2\mathbf{R}^{xx}(k) \\
& + \mathbf{R}^{xx}(k)\mathrm{tr}\left\{\mathbf{R}^{xx}(k)\left[\mathbf{I}-\mathbf{W}(k-1)\mathbf{W}^H(k-1)\right]^2\right\} \\
& \times \mathbf{W}(k-1)\left[\mathbf{W}^H(k-1)\mathbf{R}^{xx}(k)\mathbf{W}(k-1)\right]^{-1}. \qquad (5.28)
\end{aligned}$$

Diagonalizing the later equation yields

$$\begin{aligned}\mathbf{\Sigma}^T(k)\mathbf{\Sigma}(k) = \mathbf{\Sigma}^T(k-1)\mathbf{\Sigma}(k-1) + 2\gamma(k)\mathbf{\Sigma}^T(k-1)\left[\mathbf{I} - \mathbf{\Sigma}(k-1)\mathbf{\Sigma}^T(k-1)\right]\mathbf{\Lambda}_{xx}(k) \\ \times \mathbf{\Sigma}(k-1)\left[\mathbf{\Sigma}^T(k-1)\mathbf{\Lambda}_{xx}(k)\mathbf{\Sigma}(k-1)\right]^{-1} \\ +\gamma^2(k)\left[\mathbf{\Sigma}^T(k-1)\mathbf{\Lambda}_{xx}(k)\mathbf{\Sigma}(k-1)\right]^{-1} \\ \times \mathbf{\Sigma}^T(k-1)\mathbf{\Lambda}_{xx}(k)\left[\mathbf{I}-\mathbf{\Sigma}(k-1)\mathbf{\Sigma}^T(k-1)\right]^2 \\ \times \mathbf{\Lambda}_{xx}(k)\mathbf{\Sigma}(k-1)\left[\bar{\mathbf{\Sigma}}(k)\mathbf{\Lambda}_{xx}(k)\bar{\mathbf{\Sigma}}(k)\right]^{-1}. \qquad (5.29)\end{aligned}$$

The second part will thus not disturb the diagonal nature of the analysis but change the coefficient for γ^2. Continuing with our Gaussian assumption for $\underline{\mathbf{x}}(k)$ we obtain a second order condition.

Theorem 1.2. *Given a complex-valued Gaussian random process* $\mathbf{x}(k)$*, the PAST algorithm converges in the mean-square sense as long as*

$$\left|1 - 2\gamma\frac{\left(\sigma^l(k-1)\right)^2}{\bar{\sigma}^l(k)^2} - \gamma^2\frac{\left(\sigma^l(k-1)\right)^2}{\bar{\sigma}^l(k)^2}\rho^l\right| \;<\; 1 \qquad (5.30)$$

$$\rho^l \;=\; \frac{(1-\left(\sigma^l(k-1)\right)^2)^2\lambda^l + \sum_{m=1}^r |1-(\sigma^m(k-1))^2|^2\lambda^m}{(1-(\sigma^l(k-1))^2\bar{\sigma}^l(k))^2\lambda^l}. \qquad l = 1, ..., r$$

Consider the second term $\sum_{m=1}^r |1-(\sigma^m(k-1))^2|^2\lambda^m$ in the numerator of ρ^l above that applies equally to all term $l = 1, ..., r$. Leaving it out, we obtain the same result as in the first order analysis. We thus have to ask what influence it may have. As the term appear equally for all diagonal entries, it balances the terms a bit, but it has no other influence on the stability then to limit the values of $\gamma(k)$ even further. As it appears only in $\gamma(k)^2$ its influence only shows for larger values of $\gamma(k)$. For small values of $\gamma(k)$ the smallest singular value $\sigma_{\min}$ remains decisive to find the stability bound on the step-size $\gamma(k)$.

Throughout this work, we realized that this method can be further extended to incorporate also distributed versions of PAST, that rely on one or more consensus algorithms for data exchange in wireless sensor networks [61].

5.1.4 Mathematical derivation of two distributed PAST variants

To decentralize the PAST algorithm, it is necessary to regard (5.1)-(5.4) as the primary set of equations locally running at the same time in each node i. This requires to extend all these variables occurring simultaneously by the localization index i, i.e., $\underline{\mathbf{y}}_i(k) \in \mathbb{C}^{r\times 1}, \widehat{\mathbf{R}}_i^{yy}(k) \in \mathbb{C}^{r\times r}$, $\underline{\mathbf{e}}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times 1}, \underline{\mathbf{x}}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times 1}$, $\mathbf{W}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times r}$ as well as $\beta_i(k)$ and $\gamma_i(k)$. Since the communication across the sensors is done synchronously, each node i broadcasts its own observation $\{x_i(k)\}$ and receives

$\{x_{j,k}\}_{\substack{\forall j\in\mathcal{N}_i\\ j\neq i}}$. With this available information, i aggregates a local observation vector $\underline{\mathbf{x}}_i(k)$, which can be expressed in terms of the global observation vector $\underline{\mathbf{x}}(k)$ and the selection matrix $\mathbf{S}_i \in N \times |\mathcal{N}_i|$ from (3.6) according to

$$\underline{\mathbf{x}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k). \tag{5.31}$$

From (5.1) it is straightforward to write

$$\underline{\mathbf{y}}_i(k) = \mathbf{W}_i^H(k-1)\underline{\mathbf{x}}_i(k), \tag{5.32}$$

and substituting (5.31) for $\underline{\mathbf{x}}_i(k)$ originates $\underline{\mathbf{y}}_i(k) = \mathbf{W}_i^H(k-1)\mathbf{S}_i^T\underline{\mathbf{x}}(k)$. For later calculations, it is advisable to introduce the matrix $\mathbf{V}_i^H(k-1) = \mathbf{W}_i^H(k-1)\mathbf{S}_i^T$ and let the original subspace spanned by the columns of $\mathbf{W}_i(k)$ be embedded in $\mathbf{V}_i(k) \in \mathbb{C}^{N\times r}$. Under such considerations, (5.32) turns into

$$\underline{\mathbf{y}}_i(k) = \mathbf{V}_i^H(k-1)\underline{\mathbf{x}}(k). \tag{5.33}$$

We remark the fact that the above equation designates the signal update vector (first step) of both later introduced algorithmic variants. This new expression makes possible to reformulate Algorithm 2 according to the structure in (5.1)-(5.4), yielding

$$\underline{\mathbf{y}}_i(k,t) = g_{ii}\underline{\mathbf{y}}_i(k,t-1) + \sum_{\substack{j\in\mathcal{N}_i,\\ j\neq i}} g_{ij}\underline{\mathbf{y}}_j(k,t-1) \tag{3.7}$$

$$\widetilde{\underline{\mathbf{y}}}_i(k) = \underline{\mathbf{y}}_i(k,t_{\max}) \tag{3.8}$$

$$\underline{\mathbf{e}}_i(k) = \mathbf{S}^T\left(\underline{\mathbf{x}}(k) - \mathbf{V}_i(k-1)\widetilde{\underline{\mathbf{y}}}_i(k)\right) \tag{5.34}$$

$$\widehat{\mathbf{R}}_i^{yy}(k) = \widehat{\mathbf{R}}_i^{yy}(k-1) + \beta_i(k)\left[\widetilde{\underline{\mathbf{y}}}_i(k)\widetilde{\underline{\mathbf{y}}}_i^H(k) - \widehat{\mathbf{R}}_i^{yy}(k-1)\right] \tag{5.35}$$

$$\mathbf{V}_i(k) = \mathbf{V}_i(k-1) + \gamma_i(k)\underline{\mathbf{e}}_i(k)\widetilde{\underline{\mathbf{y}}}_i^H(k)\left(\widehat{\mathbf{R}}_i^{yy}(k)\right)^{-1}. \tag{5.36}$$

Similarly, the new local representation of Algorithm 3 results in

$$\underline{\mathbf{e}}_i(k) = \mathbf{S}^T\left(\underline{\mathbf{x}}(k) - \mathbf{V}_i(k-1)\mathbf{V}^H(k-1)\underline{\mathbf{x}}(k)\right) \tag{5.37}$$

$$\widehat{\mathbf{R}}_i^{yy}(k) = \widehat{\mathbf{R}}_i^{yy}(k-1) + \beta_i(k)\left[\underline{\mathbf{y}}_i(k)\underline{\mathbf{y}}_i^H(k) - \widehat{\mathbf{R}}_i^{yy}(k-1)\right] \tag{5.38}$$

$$\widehat{\mathbf{R}}_i^{yy}(k,t) = g_{ii}\widehat{\mathbf{R}}_i^{yy}(k,t-1) + \sum_{\substack{j\in\mathcal{N}_i,\\ j\neq i}} g_{ij}\widehat{\mathbf{R}}_j^{yy}(k,t-1), \tag{3.9}$$

$$\widetilde{\mathbf{R}}_i^{yy}(k) = \widehat{\mathbf{R}}_i^{yy}(k,t_{\max}) \tag{3.10}$$

$$\mathbf{V}_i(k) = \mathbf{V}_i(k-1) + \gamma_i(k)\underline{\mathbf{e}}_i(k)\underline{\mathbf{y}}_i^H(k)\left(\widetilde{\mathbf{R}}_i^{yy}(k)\right)^{-1}. \tag{5.39}$$

Notice that the main difference between both local representations (3.7)-(5.36) and (5.37)-(5.39) is the update of $\widehat{\mathbf{R}}_i^{yy}(k)$ and $\mathbf{V}_i(k)$, influenced by the average

consensus process.

In the following, we examine the convergence properties of these algorithms based on [80]. To start, it is required to find beforehand a global description for such methods. A block diagonal structure is a suitable approach for representing these global algorithms, since it permits to place each local variable of i in the diagonal of a block matrix and allows for an easy manipulation through Kronecker products.

Algorithm 4: Global description of distributed PAST with $\underline{\mathbf{y}}_i(k)$ averaging

Input: $\beta(k), \gamma(k), \widehat{\mathbf{R}}^{yy}(0), \mathbf{V}(0)$
for $k := 1, 2, ...,$ **do**

$$\mathbf{Y}(k) = \mathbf{V}^H(k-1)\left(\underline{\mathbf{I}}_N \otimes \underline{\mathbf{x}}(k)\right)$$

$$\widetilde{\mathbf{Y}}(k) = \left(\mathbf{G}^{t_{\max}} \otimes \mathbf{I}_r\right)\mathbf{V}^H(k-1)\left(\underline{\mathbf{x}}(k) \otimes \underline{\mathbf{1}}_N\right) \tag{5.40}$$

$$\mathbf{E}(k) = \mathbf{S}^T\left[\left(\mathbf{I}_N \otimes \underline{\mathbf{x}}(k)\right) - \mathbf{V}(k-1)\left(\mathbf{G}^{t_{\max}} \otimes \mathbf{I}_r\right)\mathbf{Y}(k)\right]$$

$$\widehat{\mathbf{R}}^{yy}(k) = \widehat{\mathbf{R}}^{yy}(k-1) + \beta\left(\widetilde{\mathbf{Y}}(k)\widetilde{\mathbf{Y}}^H(k) - \widehat{\mathbf{R}}^{yy}(k-1)\right) \tag{5.41}$$

$$\mathbf{V}(k) = \mathbf{V}(k-1) + \gamma(k)\mathbf{S}\mathbf{E}(k)\,\mathrm{diag}\left(\widetilde{\mathbf{Y}}^H(k)\right)\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1} \tag{5.42}$$

end for

Algorithm 5: Global description of distributed PAST with $\mathbf{R}_i(k)$ averaging

Input: $\beta(k), \gamma(k), \widehat{\mathbf{R}}_0^{yy}, \mathbf{V}_0$
for $k := 1, 2, ...,$ **do**

$$\mathbf{Y}(k) = \mathbf{V}^H(k-1)\left(\underline{\mathbf{I}}_N \otimes \underline{\mathbf{x}}(k)\right)$$

$$\mathbf{E}(k) = \mathbf{S}^T\left(I_{N^2} - \mathbf{V}(k-1)\mathbf{V}^H(k-1)\right)\left(\underline{\mathbf{I}}_N \otimes \underline{\mathbf{x}}(k)\right)$$

$$\widehat{\mathbf{R}}^{yy}(k) = \widehat{\mathbf{R}}^{yy}(k-1) + \beta\left(\mathbf{Y}(k)\mathbf{Y}^H(k) - \widehat{\mathbf{R}}^{yy}(k-1)\right) \tag{5.43}$$

$$\widetilde{\mathbf{R}}^{yy}(k) = \left(\mathbf{G}^{t_{\max}} \otimes \mathbf{I}_r\right)\widehat{\mathbf{R}}^{yy}(k)\left(\mathbf{1}_N \otimes \mathbf{I}_r\right) \tag{5.44}$$

$$\mathbf{V}(k) = \mathbf{V}(k-1) + \gamma(k)\mathbf{S}\mathbf{E}(k)\mathbf{Y}^H(k)\,\mathrm{diag}\left(\widetilde{\mathbf{R}}^{yy}(k)\right)^{-1} \tag{5.45}$$

end for

We acknowledge that in Algorithm 1, the update equation $\widetilde{\mathbf{Y}}(k) = \left[\underline{\mathbf{y}}_i(k), \ldots, \underline{\mathbf{y}}_N(k)\right]$ is indeed containing the averaged signal values from each node i. Note that for updating (5.42), we restructure $\widetilde{\mathbf{Y}}(k)$ in such a way that it becomes a block diagonal structure. The same situation occurs in Algorithm 2, whith $\widetilde{\mathbf{R}}^{yy}(k) = \left[\widetilde{\mathbf{R}}_i^{yy}(k), \ldots, \widetilde{\mathbf{R}}_N^{yy}(k)\right]^T$. Here, an update of (5.45) requires a reshaping of the global correlation matrix $\widetilde{\mathbf{R}}^{yy}(k)$. A more detailed derivation of both algorithms is provided in the Appendix (C). Nevertheless, a similarity between the structure of the update equation (5.42) and (5.45) with (5.4) from the centralized solution in [80] is forthwith recognizable. As a result,

it is straightforward to apply the guidelines from [80] to evaluate the convergence properties of both distributed algorithms.

5.1.5 First order analysis of distributed PAST

This subsection studies the convergence properties of both global distributed subspace tracking variants, i.e., the Algorithm 4 and Algorithm 5. To this aim, we stem on the lines presented for the centralized PAST (Subsection 5.1.1) and extend them to provide an insight regarding convergence in the mean and mean square. Furthermore, we seek for step-sizes that guarantee stability.

We provide a first order analysis for Algorithm 4 and Algorithm 5. As in the previos section, the main goal is to obtain a unitary matrix for the $\mathbf{W}_i(k)$ embedded in $\mathbf{V}(k)$, such that $\lim_{k\to\infty} \mathbf{W}_i(k)^H \mathbf{W}_i(k) = \mathbf{I}_r$. We assume that the data vector $\underline{\mathbf{x}}(k)$ containing all the observations in the network, is of random nature and has an autocorrelation matrix $\mathbf{R}^{xx}(k) = E\left\{ (\underline{\mathbf{x}}(k) \otimes \underline{\mathbf{1}}_N) (\underline{\mathbf{x}}(k) \otimes \underline{\mathbf{1}}_N)^H \right\}$. We are interested in a first order analysis of the signal subspace update of both aforementioned algorithms. Therefore, we evaluate the mean of (5.42) and rewrite it as

$$E\{\mathbf{V}(k)\} = E\{\mathbf{V}(k-1)\} + \gamma(k) E\left\{ (\mathbf{I}_{N^2} - \mathbf{V}(k-1)\mathbf{V}(k-1)^H) \underline{\mathbf{x}}(k) \underline{\mathbf{x}}(k)^H \mathbf{V}(k-1) \right\} \times E\left\{ \left(\widehat{\mathbf{R}}^{yy}(k) \right)^{-1} \right\}. \tag{5.46}$$

Similarly, (5.45) is expressed as

$$E\{\mathbf{V}(k)\} = E\{\mathbf{V}(k-1)\} + \gamma(k) E\left\{ (\mathbf{I}_{N^2} - \mathbf{V}(k-1)\mathbf{V}(k-1)^H) \underline{\mathbf{x}}(k) \underline{\mathbf{x}}(k)^H \mathbf{V}(k-1) \right\} \times E\left\{ \left(\widetilde{\mathbf{R}}^{yy}(k) \right)^{-1} \right\}. \tag{5.47}$$

In order to solve the two above equations, it is very important to consider the iterative approximations $\left(\widetilde{\mathbf{R}}^{yy}(k) \right)^{-1}$ and $\left(\widehat{\mathbf{R}}^{yy}(k) \right)^{-1}$ to be independent of the signal subspace matrix $\mathbf{V}(k)$. We model $\underline{\mathbf{x}}(k)$ as a random, ergodic process and let the calculation of the ensemble average $\mathbf{R}^{yy}(k)$ of the given process be determined from its time average $\widehat{\mathbf{R}}^{yy}(k)$ over past samples as

$$\mathbf{R}^{yy}(k) = E\left\{ \widehat{\mathbf{R}}^{yy}(k) \right\} = E\left\{ \mathbf{Y}(k)\mathbf{Y}(k)^H \right\} = \mathbf{V}^H(k-1)\mathbf{R}^{xx}(k)\mathbf{V}(k-1), \tag{5.48}$$

and its matrix decomposition

$$\mathbf{R}^{yy}(k) = \mathbf{P}_1 \bar{\mathbf{\Sigma}}(k) \mathbf{\Lambda}^{xx}(k) \bar{\mathbf{\Sigma}}(k) \mathbf{P}_1^H. \tag{5.49}$$

Due to this averaging process, it is likely to regard such average values as independent of each other. That is to say, we allow them to be statistically independent [49], and let the decomposition of the autocorrelation matrix

$\mathbf{R}^{xx} = \mathbf{Q}\mathbf{\Lambda}^{xx}\mathbf{Q}^H$. Recall from [80], that a steady-state solution only exists if $E\{\mathbf{V}(k)\} = [\mathbf{U}_1, \mathbf{U}_2][\mathbf{\Sigma}(k), \mathbf{0}]^T[\mathbf{P}_1\mathbf{P}_2]^H$ and $\mathbf{U}_1 = \mathbf{Q}$. We applied an SVD to (5.46) and (5.47) and partitioned it into the significant part $\mathbf{U}_1\mathbf{\Sigma}(k)\mathbf{P}_1^H$ and a zero block. While diagonalizing these equations, we observe that all block matrices encompass the same left block matrix $\mathbf{U}$ and the right block matrix $\mathbf{P}_1^H$. These matrices are simplified such that the final solution accomodates only the center block matrix $\mathbf{\Sigma}(k)$ comprising $\mathbf{\Sigma}_i(k)$ diagonal elements, i.e.,

$$\mathbf{\Sigma}(k) = \mathbf{\Sigma}(k-1) + \gamma(k)(\mathbf{I} - \mathbf{\Sigma}(k-1)\mathbf{\Sigma}(k-1))\mathbf{\Lambda}^{xx}\mathbf{\Sigma}(k-1)(\bar{\mathbf{\Sigma}}(k)\mathbf{\Lambda}^{xx}\bar{\mathbf{\Sigma}}(k))^{-1}. \tag{5.50}$$

The above expression is obtained with (5.46). If (5.47) is used instead, an analogous expression can be found. Furthermore, it is possible to revise (5.50) in terms of the local singular values at each node i. That is to say

$$\sigma_i^l(k) = \sigma_i^l(k-1) + \gamma_i(k)\frac{\left(1 - \left(\sigma_i^l(k-1)\right)^2\right)\sigma_i^l(k-1)}{\left(\bar{\sigma}_i^l(k)\right)^2}. \qquad l = 1, \ldots, r$$

Here, $\bar{\sigma}_i^l(k-1)$ are the singular values calculated by diagonalizing the time average $\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}$. They should not be confused with the instantaneous singular values $\sigma_i^l(k)$ derived from decomposing $\mathbf{V}(k)$. Under the assumption that all signals are perfectly decorrelated, the steady state solution is achieved if, i.e., $\sigma_i^l(k-1) \to 1$. Thus

$$1-\sigma_i^l(k) = (1-\sigma_i^l(k-1))\left[1 - \gamma(k)\frac{(1+\sigma_i^l(k-1))\sigma_i^l(k-1)}{\left(\bar{\sigma}_i^l(k)\right)^2}\right]. \tag{5.51}$$

As all singular values are positive we can again see that the PAST algorithm converges in the mean for a sufficiently small step-size $\gamma_i(k) > 0$ if $\frac{(1+\sigma_i^l(k-1))\sigma_i^l(k-1)}{\left(\bar{\sigma}_i^l(k)\right)^2}$ is bounded. However, since this term can take on arbitrary values in the range $[0, \infty]$, it is difficult to bound the step-size at this point. We recognize that in particular small singular values are decisive.

Note that similar forms of (5.46) and (5.47) have been analyzed in [82] in the context of blind source separation and [83] as a means to compute robustly matrix inverses.

5.1.6 Step - sizes

Let us return to Equation (5.51). We learn here that

$$0 < \gamma(k) < \frac{2\bar{\sigma}_i(k)^2}{(1+\sigma_i(k-1))\sigma_i(k-1)}; i = 1, ..., r, \tag{5.52}$$

and we assume that it has an upper bound

$$0 < \gamma(k) < \frac{2\bar{\sigma}^2_{\min}(k)}{(1+\sigma_{\max}(k-1))\sigma_{\max}(k-1)}. \tag{5.53}$$

If we use the relation $0.25 + 2x^2 \geq x + x^2$ for $x \geq 0$ we obtain an even lower bound

$$0 < \gamma(k) < \frac{2\bar{\sigma}^2_{\min}(k)}{0.25 + 2\sigma^2_{\max}(k-1)}. \tag{5.54}$$

The term $\sigma^2_{\max}(k-1)$ is related to $\mathbf{W}(k-1)^H\mathbf{W}(k-1)$. If $\lambda^{xx}(k)$ and $\mathbf{v}(k)$ are an eigenvalue and eigenvector pair of the matrix $\mathbf{W}(k)^H\mathbf{W}(k)$, the following condition holds true:

$$\begin{aligned} |\lambda^{xx}(k)|^m\|\mathbf{v}(k)\| &= \|(\lambda^{xx})^m(k)\mathbf{v}(k)\| && (5.55)\\ &= \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\mathbf{v}(k)\| \\ &\leq \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\| \cdot \|\mathbf{v}(k)\| && (5.56)\\ \implies |\lambda^{xx}(k)| &\leq \|(\mathbf{W}(k)^H\mathbf{W}(k))^m\|^{\frac{1}{m}}. && (5.57) \end{aligned}$$

In [80], the following more conservative upper bound for $\sigma^l_i(k-1)$ was found

$$0 < \gamma_i(k) < \frac{2\left(\bar{\sigma}^l_i(k)\right)^2}{0.25 + 2\|\mathbf{W}_i(k-1)^H\mathbf{W}_i(k-1)\|^2} = \gamma_{\max_i(k)}\left(\bar{\sigma}^l_{\min_i(k)}\right)^2. \tag{5.58}$$

The knowledge of the smallest singular value $\bar{\sigma}^l_{\min_i(k)}$ is decisive for convergence. Nevertheless, calculating it or even finding an estimate for it would lead to a high computational burden. Therefore, we follow the guidelines from [80] and alter the $\mathbf{V}(k)$ update equation in Algorithm 2 and Algorithm 3 such that calculating the smallest singular value becomes nonessential. Accordingly, (5.42) becomes

$$\mathbf{V}(k) = \mathbf{V}(k-1) + \gamma(k)\mathbf{SE}(k)\ \mathrm{diag}\left(\widetilde{\mathbf{Y}}^H(k)\right)\mathbf{A}, \tag{5.59}$$

and (5.45) yields

$$\mathbf{V}(k) = \mathbf{V}(k-1) + \gamma(k)\mathbf{SE}(k)\mathbf{Y}^H(k)\ \mathbf{B}. \tag{5.60}$$

Recall the different step-sizes originally introduced in [80], which guarantee feasible step-sizes bounds and let them be addressed at each node i as

- PAST-I:

$$\begin{aligned} \mathbf{A} &= \mathrm{diag}\left(\widetilde{\mathbf{Y}}^H(k)\right)\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}\left(\bar{\sigma}^l_{\min_i(k)}\right)^2 \\ \mathbf{B} &= \mathbf{Y}^H(k)\mathrm{diag}\left(\widetilde{\mathbf{R}}^{yy}(k)\right)^{-1}\left(\bar{\sigma}^l_{\min_i(k)}\right)^2 \end{aligned}$$

- PAST-II:

$$\mathbf{A} = diag\left(\widetilde{\mathbf{Y}}^H(k)\right)\left(\widehat{\mathbf{R}}^{yy}(k)\right)^{-1}\mathbf{R}_{\mathbf{V}_i(k)}$$
$$\mathbf{B} = \mathbf{Y}^H(k)diag\left(\widetilde{\mathbf{R}}^{yy}(k)\right)^{-1}\mathbf{R}_{\mathbf{V}_i(k)}$$

$$0 < \gamma_{I_i(k)} = \gamma_{II_{i(k)}} < \gamma_{\max_i(k)} \leq \min_l \frac{2}{\left(1+\sigma_i^l(k-1)\right)\sigma_i^l(k-1)} \tag{5.61}$$

- PAST-III:

$$\mathbf{A} = diag\left(\widetilde{\mathbf{Y}}^H(k)\right)\left[\widehat{\mathbf{R}}^{yy}(k) + \alpha\mathbf{I}_r\right]^{-1}$$
$$\mathbf{B} = \mathbf{Y}^H(k)\left[diag\left(\widetilde{\mathbf{R}}^{yy}(k)\right) + \alpha\mathbf{I}_r\right]^{-1}$$

$$0 < \gamma_{III_i(k)} < \gamma_{\max_i(k)}\delta_{\min_i(k)} \leq \min_l \frac{2\left(\left(\bar{\sigma}_i^l(k-1)\right)^2+\delta\right)}{\left(1+\sigma_i^l(k-1)\right)\sigma_i^l(k-1)} \tag{5.62}$$

- PAST-IV:

$$\mathbf{A} = diag\left(\widetilde{\mathbf{Y}}^H(k)\right)\mathbf{I}_r,$$
$$\mathbf{B} = \mathbf{Y}^H(k)\mathbf{I}_r$$

$$0 < \gamma_{IV_i(k)} < \frac{\gamma_{\max_i(k)}}{\mathrm{tr}\left(\mathbf{R}_i^{xx}(k)\right)} \leq \min_l \frac{2}{\left(1+\sigma_i^l(k-1)\right)\sigma_i^l(k-1)\lambda_i^{l,xx}}. \tag{5.63}$$

For the PAST-I version, we assume knowledge of the smallest singular value only. This can be achieved by tailored tracking algorithms, e.g., [85]. The PAST-II variant calculates an average $\mathbf{R}_{\mathbf{V}_i}(k)$ for the matrix $\mathbf{V}_i(k-1)^H\mathbf{V}_i(k-1)$, given by

$$\mathbf{R}_{\mathbf{V}_i}(k) = \mathbf{R}_{\mathbf{V}_i}(k-1) + \beta(k)(\mathbf{V}_i(k-1)^H\mathbf{V}_i(k-1) - \mathbf{R}_{\mathbf{V}_i}(k-1)), \tag{5.64}$$

and is applied to compensate for the inverse singular values of $\widehat{\mathbf{R}}^{yy}(k)$ or $\widetilde{\mathbf{R}}^{yy}(k)$. The PAST-III scheme introduces a regularization for computing the inverse of this matrix, which prevents numerical issues due to possible ill conditioned matrices. Finally, the PAST-IV version can be considered as the gradient-type version of the original PAST algorithm from [8]. In order to avoid the eigenvalue decomposition of the local data correlation matrix $\mathbf{R}_i^{xx}(k) = E\{\underline{\mathbf{x}}_i(k)\underline{\mathbf{x}}_i(k)\}$ for finding $\lambda_i^{l,xx}$ in (5.63), we consider only its trace. We also let $0 < \alpha \leq 1$, $\delta = \alpha/\lambda_i^{l,xx}$ and $\delta_{\min} = \alpha/\lambda_{\max_i}^{xx}$. Note that the choice of α also influences the convergence speed. The higher the

selected value, the faster the achieved convergence. To summarize, by utilizing the aforementioned step-sizes we evade calculating the singular values in $\widehat{\mathbf{R}}^{yy}(k)$ or $\widetilde{\mathbf{R}}^{yy}(k)$, as would be required in (5.58). Furthermore, the computation of a safe upper bound of $\gamma_i(k)$ relies only on the matrix norm, which is a computationally feasible operation. As shown in [80], the induced matrix 2- norm provides tight results. Finally, with these feasible step-sizes, an evaluation of the algorithmic behavior is presented in the next section.

5.1.7 Simulation results - First order analysis

The following experiments apply both distributed PAST variants to a typical beam steering scenario. Consider a wireless sensor network with $N = 12$ nodes, where each sensor i observes the data $\underline{\mathbf{x}}_i(k)$ at each time k given by (**??**). We use $r = 3$ different angles of arrivals set to $\theta = \{0.01, 0.03, 0.2\}$ radians. The bounds (5.61)-(5.63) define a time-variant maximal value $\gamma_i^{\max}(k)$ and we select fractions $\beta\gamma_i^{\max}(k)$, where $\beta \in (0, 1]$. We further set $\beta(k) = \beta$ in (5.3) by which we reduce one degree of freedom and let $\gamma(k) = \beta\gamma_{\max}(k)$. We choose the maximum number of consensus iterations to be $t_{\max} = 100$ and we average over 100 Monte Carlo (MC) runs. Note that for the centralized versions we expect the algorithms to work for values β from zero to one.

We implement Algorithm 1 in a regular topology, which means that every node i in the network has the same degree $|\mathcal{N}_i| = 4$. Figure 1 depicts the distance measure $\sum_{l=1}^{r}\left(1 - \sigma^l(k)\right)$ being averaged over all nodes in the network. Here, the original update equation from (12), known as Distributed PAST, shows the fastest convergence followed by the PAST-I and PAST-II variants, which need smaller step-sizes to achieve the same steady-state. The PAST-III and PAST-IV schemes perform considerably slower and during the simulations for step-size values of β above 0.9, they both show first signs of instability.

Figure 2 provides more results for the same simulation settings as those used for Figure 1. However, for this experiment we implemented a network topology which is random irregular. The former means that the $\mathcal{N}_i$ selected neighborhood is not necessarily composed of the j sensors closest to i, they are actually selected in a random fashion. By irregular we refer to different node degrees in the network, e.g., $|\mathcal{N}_i| = 3, 5$, and 8. Note that for all MC runs we use the irregular topology that was generated in the first simulation. We move on to Figure 2 and immediately recognize that PAST-III shows a faster convergence, in contrast to all the other variants that show practically the same performance.
Now we evaluate Algorithm 2 where the correlation matrix $\widehat{\mathbf{R}}_i^{yy}(k)$ is exchanged between the nodes. Here, PAST-III and PAST-IV lead to the fastest, practically overlapping results for both regular (Figure 3) and irregular (Figure 4) algorithmic

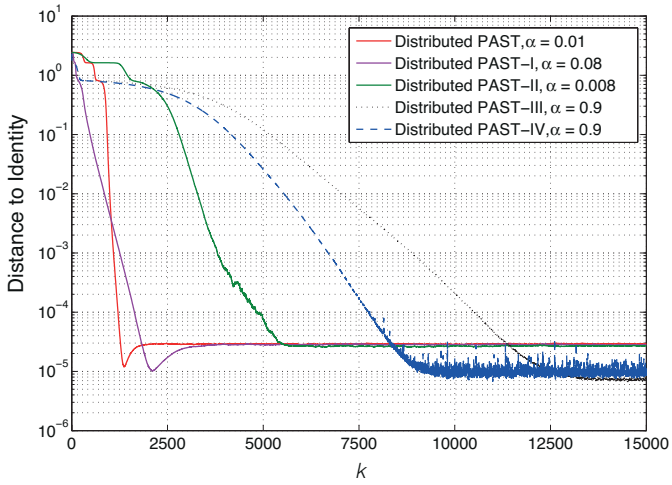

Figure 5.1: Averaged distance to identity for Algorithm 1 with a regular network topology.

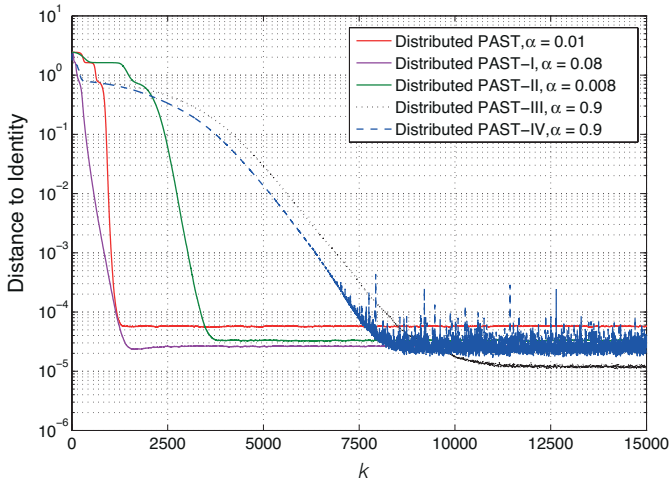

Figure 5.2: Averaged distance to identity for Algorithm 1 with a random irregular network topology.

topologies. Again, the stability limit is found near to step-size $\beta = 0.9$. We likewise observe that for both topologies PAST-I converges very slowly and that stability issues arise for step-sizes larger than 0.3. We do not present any plot for PAST-II, since the simulation results did not lead to a steady-state. Finally, by several experiments with both algorithms we realized how important it is to initialize the local autocorrelation matrix $\widehat{\mathbf{R}}_{i,0}^{yy}$ and the local signal subspace $\mathbf{W}_i(k)$ by appropriate values. For both, Algorithm 1 and Algorithm 2, the initial $\widehat{\mathbf{R}}_i^{yy}(k)$ value was defined as an identity matrix. The signal subspace $\mathbf{W}_i(k)$ in Algorithm 1 was characterized as a truncated identity matrix of size $|\mathcal{N}_i| \times r$, for Algorithm 2 we defined it to be all ones.

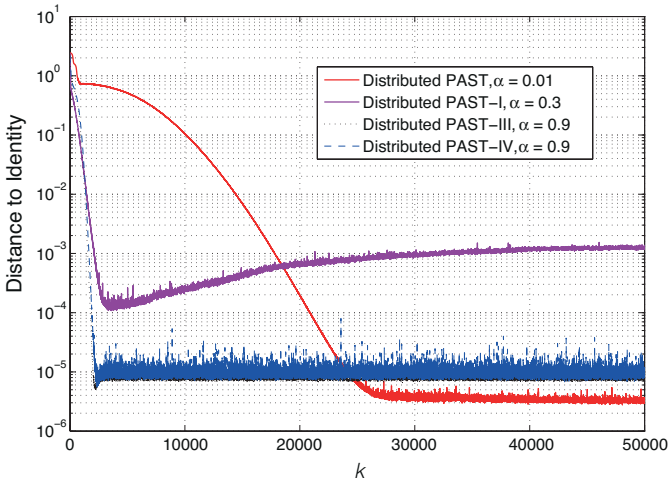

Figure 5.3: Averaged distance to identity for Algorithm 2 with a regular network topology.

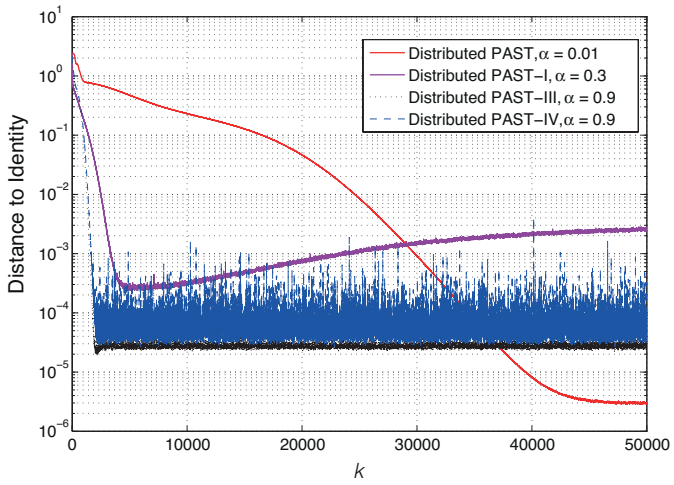

Figure 5.4: Averaged distance to identity for Algorithm 2 with a random irregular network topology.

5.1.8 Second order analysis of distributed PAST

A first order analysis is often considered as leading to relatively weak statements. In particular for adaptive filter algorithms it is known that such analysis describes only the qualitative behavior of an algorithm but only the second order analysis helps evaluating the quantitative behavior. To this end we also investigate the evolution of terms of the form $\mathbf{W}_i(k)^H\mathbf{W}_i(k)$ from the update equation (5.4).

Chapter 6

Conclusions and final remarks

THE goal of this thesis was to solve one of the current major challenges in the field of Wireless Sensor Networks, i.e., the ability to estimate radio waves in a decentralized fashion. To this aim, we employed well established subspace tracking methods [8, 40]. In particular, the roots of our work are based in the Projection Approximation Subspace Tracking algorithm [8], which in its original form operates as a centralized framework.

We proposed two distributed versions of PAST, i.e., Algorithm 2 and Algorithm 3, which exchange the local signal update vector $\underline{\mathbf{y}}_i$ and the signal covariance matrix $\widehat{\mathbf{R}}_i^{yy}$, respectively. To achieve a global state available at every sensor within the network, the Average Consensus protocol was used.

A performance analysis of the newly developed algorithms was introduced in terms of the DOA tracking capabilities. As expected, the results for strongly connected networks were the best in both approaches. Nonetheless, the performance of a network with thirty six sensor elements and regular neighborhood size ($\mathcal{N}_i$) when Algorithm 2 is employed, provides remarkably better results when contrasted to Algorithm 3. A sensor network with a regular neighborhood of the order $\frac{1}{4}$ of the total network size shows a good compromise between the DOA estimation accuracy and the amount of consumed resources is sought.

It has not yet become clear why the Algorithm 3 becomes rigid after initial good tracking results. Compared to Algorithm 2, which exchanges a local signal update vector $\underline{\mathbf{y}}_i$, the more information intensive distribution of the data covariance matrix $\widehat{\mathbf{R}}_i^{yy} \in \mathbb{C}^{N \times r}$ was expected to improve performance. Instead, Algorithm 2 consistently outperforms Algorithm 3.

In this work a convergence proof for the aforementioned algorithms was presented. In [80], we provided a convergence analysis of PAST, which was based on Singular Value Decomposition. This proof is simpler than the original proof de-

scribed in [48], and has the additional benefit of being extendable to the analysis of our new algorithmic approaches.

During the course of these years, we developed an algorithm where the local signal subspaces $\mathbf{W}_i$ were distributed. Our aim was to average those subspace intersections in order to achieve a common global subspace. However, since the basis that spans the local signal subspace $\mathbf{W}_i$ depends on the neighborhood size $\mathcal{N}_i$, this approach could not be realized while maintaining network flexibility.

To conclude, we have seen that the performance and the convergence analysis of the Algorithm 2 yield satisfactory results without adding too much complexity.

Based on the proposed distributed algorithms that track the temporal evolution of the signal subspace, another DOA algorithm can be used to further optimize the DOA estimates. In addition to this, finding optimal strategies for the adequate selection of the forgetting factors is a plausible approach.

Appendix A

Simulation results when $\underline{y}$ is distributed

Let us study the behavior of the Algorithm 2 proposed in Chapter 3 for the network topologies, sensor position, network size and connectivity from Figure 2.4. Naturally, since the main focus is to study the performance of the aforementioned distributed approach, the central processing unit (represented by the red dot) becomes just another node in the network.

The Direction-of-Arrival tracking capabilities of Algorithm 2 are investigated for a fully connected network, i.e., the topology a) from Figure 2.4 for $N = 36$ sensor elements. This scenario can be interpreted as an equivalent of PAST, with the difference that there is recurrent information exchange between all sensors. In Figure A.9 we observe that the DOA estimation does not improve when compared to those results from the centralized solution in 4.2. Meaning that incurring in any sort of information exchange for this specific scenario is not advantageous at all. On the other hand, if the performance is contrasted with all possible topologies (see Figures A.10, A.11 and A.12), this case undoubtedly yields the best estimates, as each node has access to all data available in the sensor network.

During the first 60 iterations, the first component θ_1 in 1.) from Figure A.9 is correctly calculated for all possible forgetting factors. After the step change occurs, the impact of the β's in the tracking capabilities becomes notable. For $\beta = 0.6$, the algorithm stabilizes fast, but presents a noisier behavior and certain offset with respect to the true DOA θ_1 value. This issue is improved in 2.), where a higher SNR=5 helps to reduce this gap. Juxtaposed to $\beta = 0.6$, $\beta = 0.8$ in 1.) shows the second largest lag, followed by $\beta = 0.9$, where both converge to the same results after a few iterations. Note that when the later values are evaluated in 2.), the outcome turns out to be the same. Now, have a closer look at the second component θ_2 in

1.) and recall that its step change is shorter than that of θ_1. A similar behavior for the tracked DOA is observable: as the signal is constant in the first 60 iterations, the estimates for all β's are very good. When the step changes, $\beta = 0.6$ recovers much faster. This suggests that for sudden changes in DOA (such as that in θ_1), the selection of a small β proves a benefit to the tracking capabilities of the algorithm. At last, the estimated $\widehat{\theta}_2$ in 2.) with SNR=5 provides smoother results. In Figure A.10, the dense topology b) from Figure 2.4 is studied. The initial mistracking of θ_1 in 1.) becomes more accentuated and the performance of Algorithm 2 is slightly worse when compared to the fully connected scenario in Figure A.9. The higher the network connectivity is, the more information is available at each node. As a result, the data is better estimated. Now observe in Figure A.11 the behavior of the first distributed variant for a sparsely connected network, i.e., c.) from Figure 2.4. Again, $\beta = 0.8$ in 1.) and 2.) adjust faster to the Direction-of-Arrival change. Surprisingly, 1.) shows that the signal corresponding to θ_2 is perfectly tracked in the first 60 iterations, even at SNR=-5dB However, for θ_1 MUSIC fails to estimate the correct values, until after the step, when estimates seem approach the true DOA's.

A similar scenario to the one described above is analyzed in Figure A.12: the neighborhood size for each sensor i is set to be $|\mathcal{N}_i| = 5$ elements (see d.) in Figure 2.4). This case displays a higher offset for both signals. The estimates $\widehat{\theta}_1$ are particularly worse than $\widehat{\theta}_2$, but after the step change they get in track and are able to identify the correct DOA's.

One of the issues regarding the MUSIC algorithm is the need to consider a large number of sensor elements to work at its best. Note that for $N = 36$, the outputs are correlated to their respective scenarios. Nevertheless, in the sparse connected case from Figure A.11, the lag between the estimated $\widehat{\theta}_2$ in 1.) and the $\widehat{\theta}_2$ in 2.) increases for the three forgetting factors no matter if a higher SNR values is selected. We presume that this is a consequence of the low network connectivity in this case.

In the following, we aim to corroborate these suspicions by studying a smaller sensor network with $N = 12$ elements, as shown in Figure 2.3. The first experiment is again for the fully connected scenario a) in Figure A.13. From the previous experiments it is clear that this are the best achievable results. The same trend among the β's is observable in 1.). Nevertheless, when the Signal to Noise Ratio is increased to 5dB, the lag from each forgetting factor rises unexpectedly. This trend is also exhibited in the other scenarios from Figure A.14, A.15 and A.16.

Notwithstanding the unsatisfactory tracking results for $N = 12$, we proceed the evaluate the angles error between the true signal subspace $\mathbf{\Upsilon}(k)$ and the estimated one $\mathbf{W}(k)$. Linebarger et all. shows in [86] that a good (or bad) MUSIC-based tracking performance does not necessarily imply that the signal subspace estimate is better or worse. There are several improved versions of the MUSIC algorithm for Direction-Of-Arrival estimation, (Reference...)

The principal angle difference is evaluated for Algorithm 2. As expected, 3.) and

4.) in Figure A.9 exhibit again a similar behavior to their analogous in Figure 4.2. For 3.) and 4.) of the dense, sparse, regular $\mathcal{N}_i = 5$ and regular $\mathcal{N}_i = 9$ scenarios corresponding to the Figures A.10, A.11, A.12 and 4.3, the algorithmic behavior is governed by the network connectivity and the SNR. Nonetheless, this seems to be the case only for the first principal component derived from θ_1. The angle error estimate for θ_2 is not greatly altered for higher SNR. Recall the case when $N = 12$ as shown in Figure A.13. If 3.) and 4.) are compared with those results from Figure A.9, it becomes evident that the angle error for the principal component associated to θ_1 are very similar between them. Besides this, the trend regarding the error floor for the principal angle associated to θ_2 stays valid.

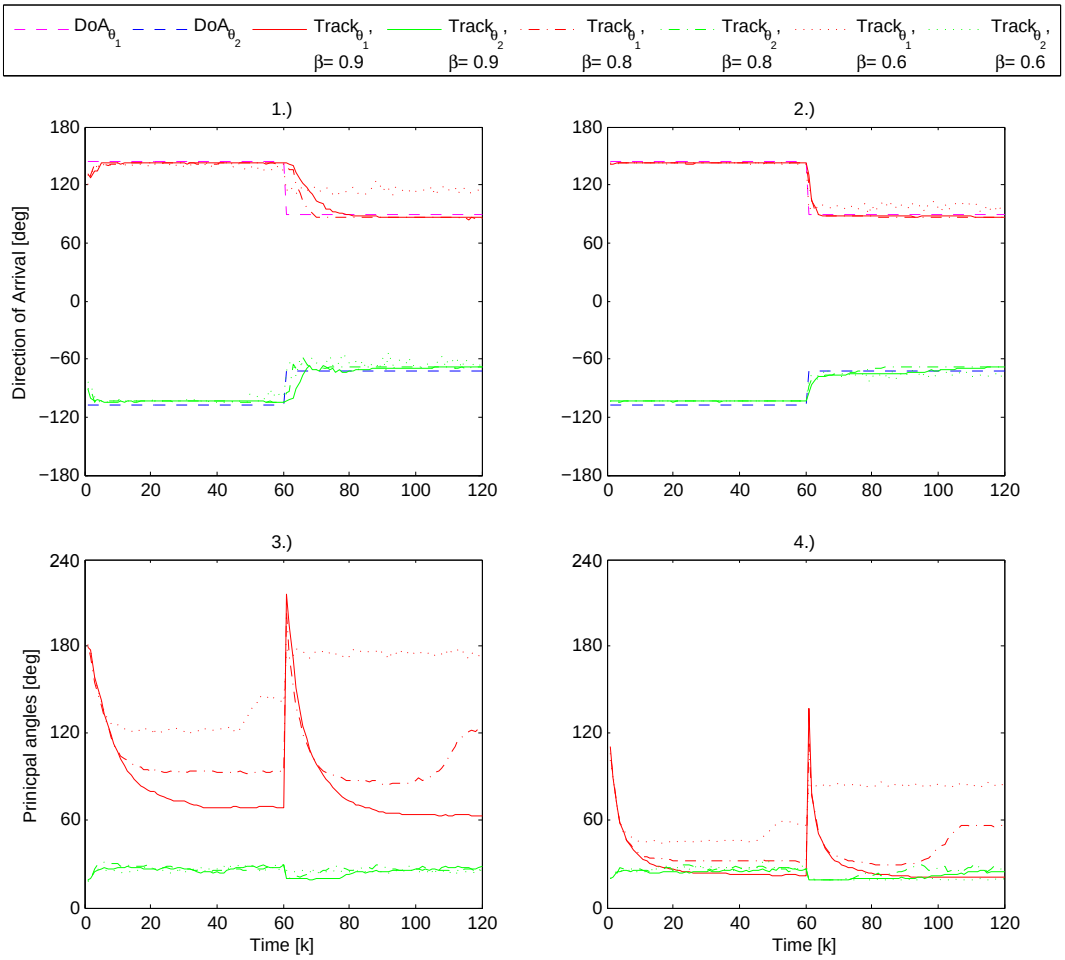

Figure A.1: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

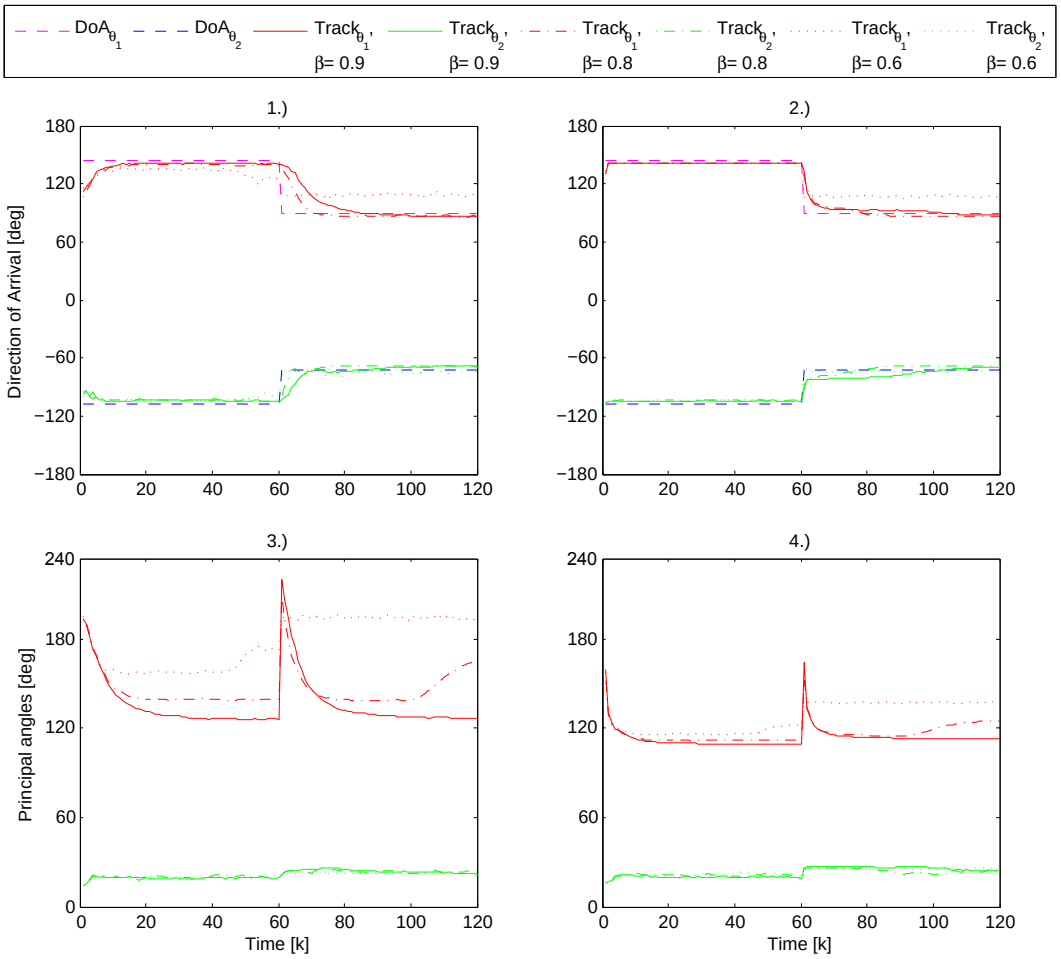

Figure A.2: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

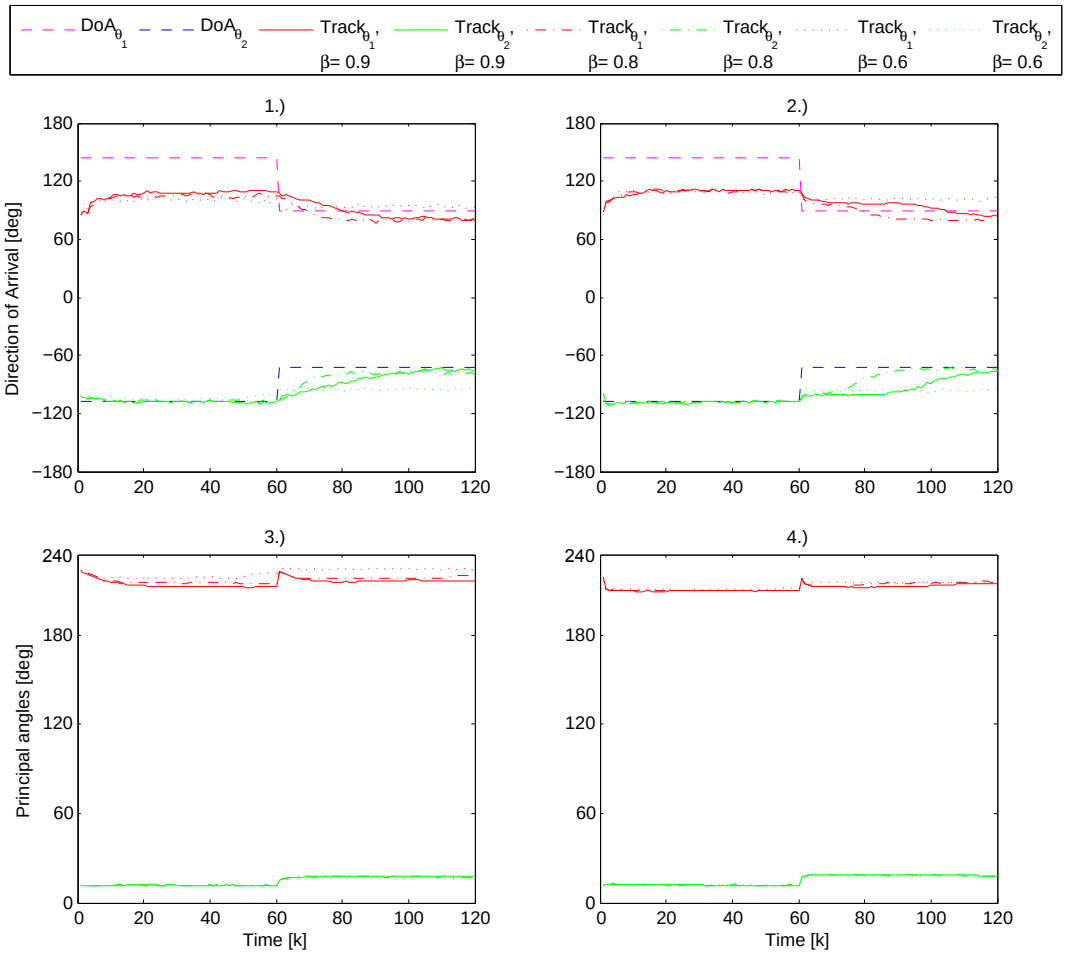

Figure A.3: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

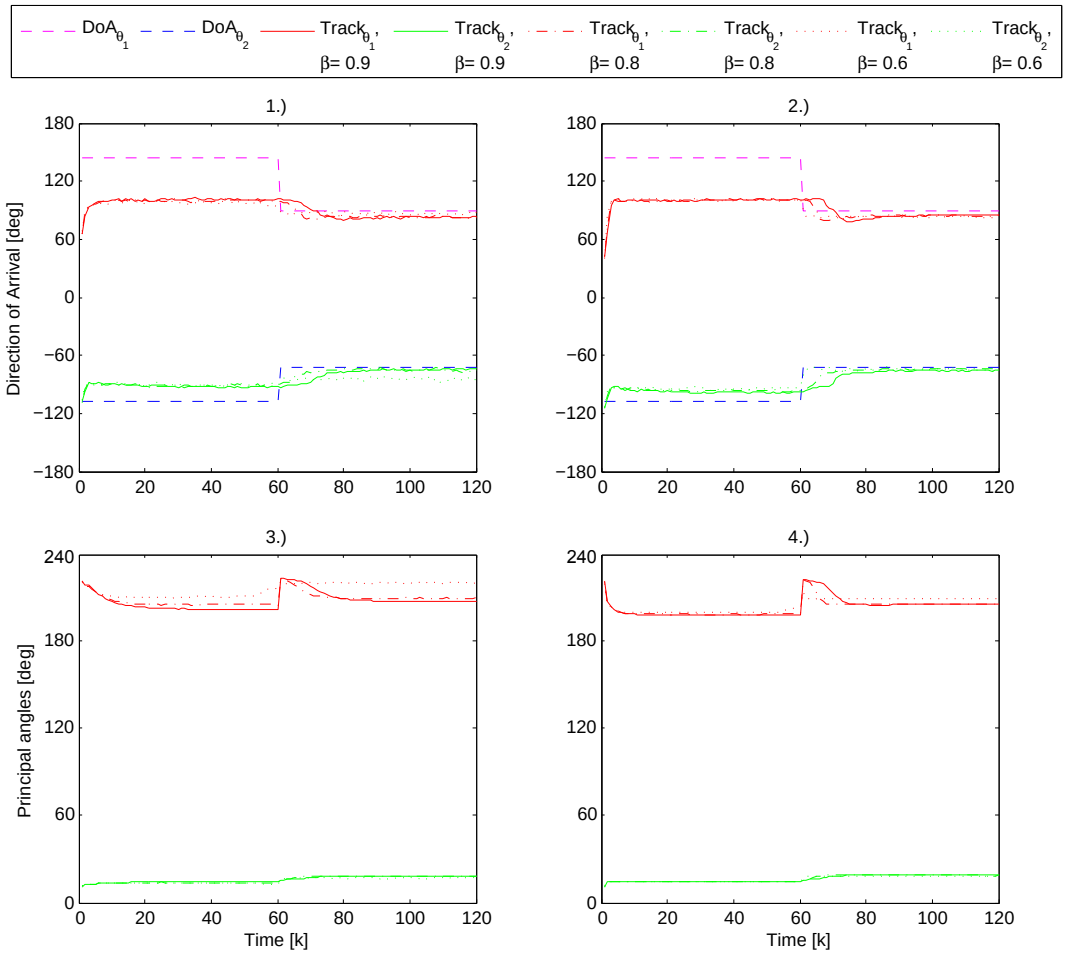

Figure A.4: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

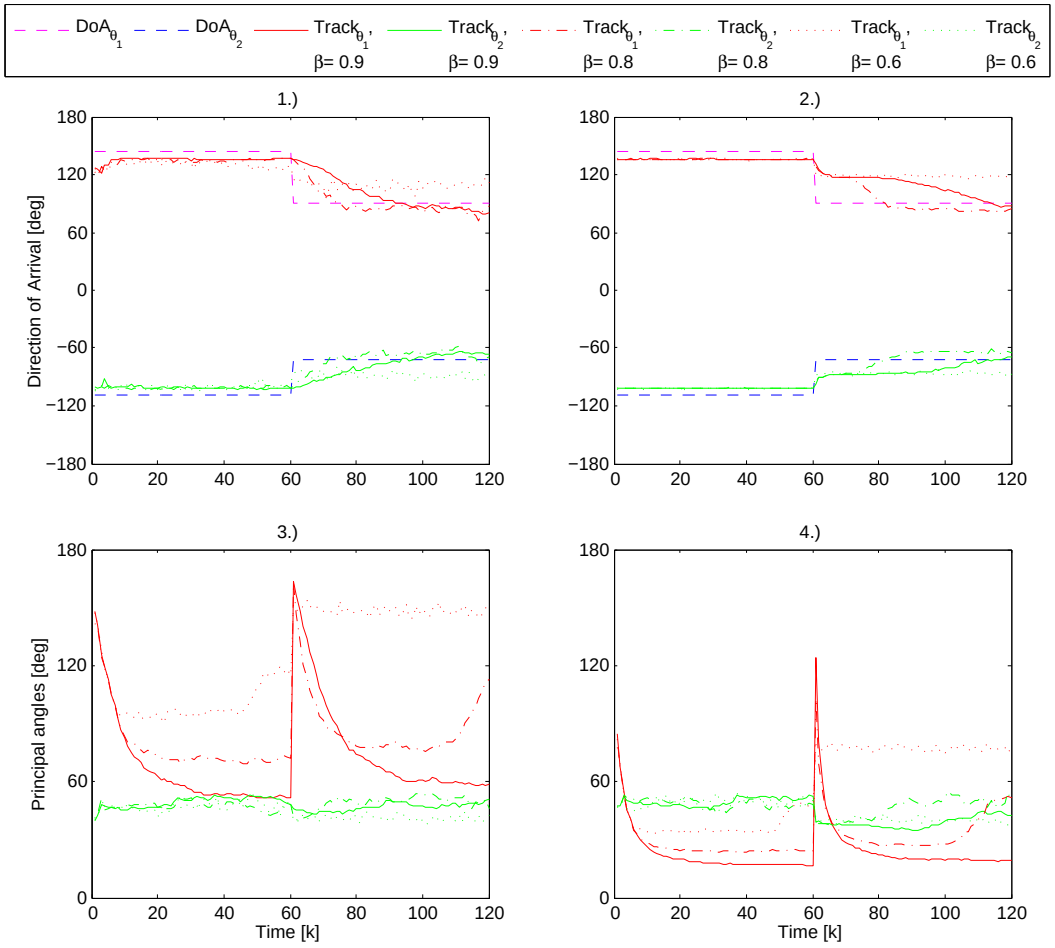

Figure A.5: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

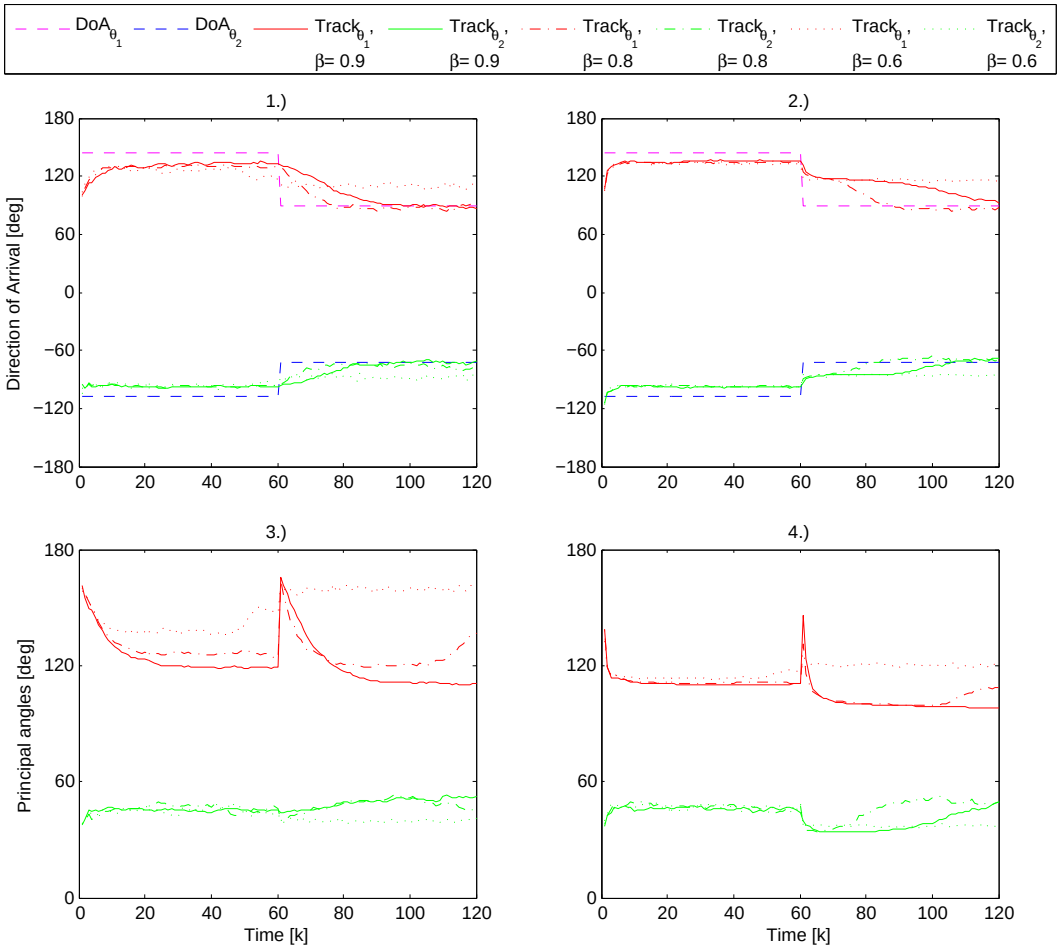

Figure A.6: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

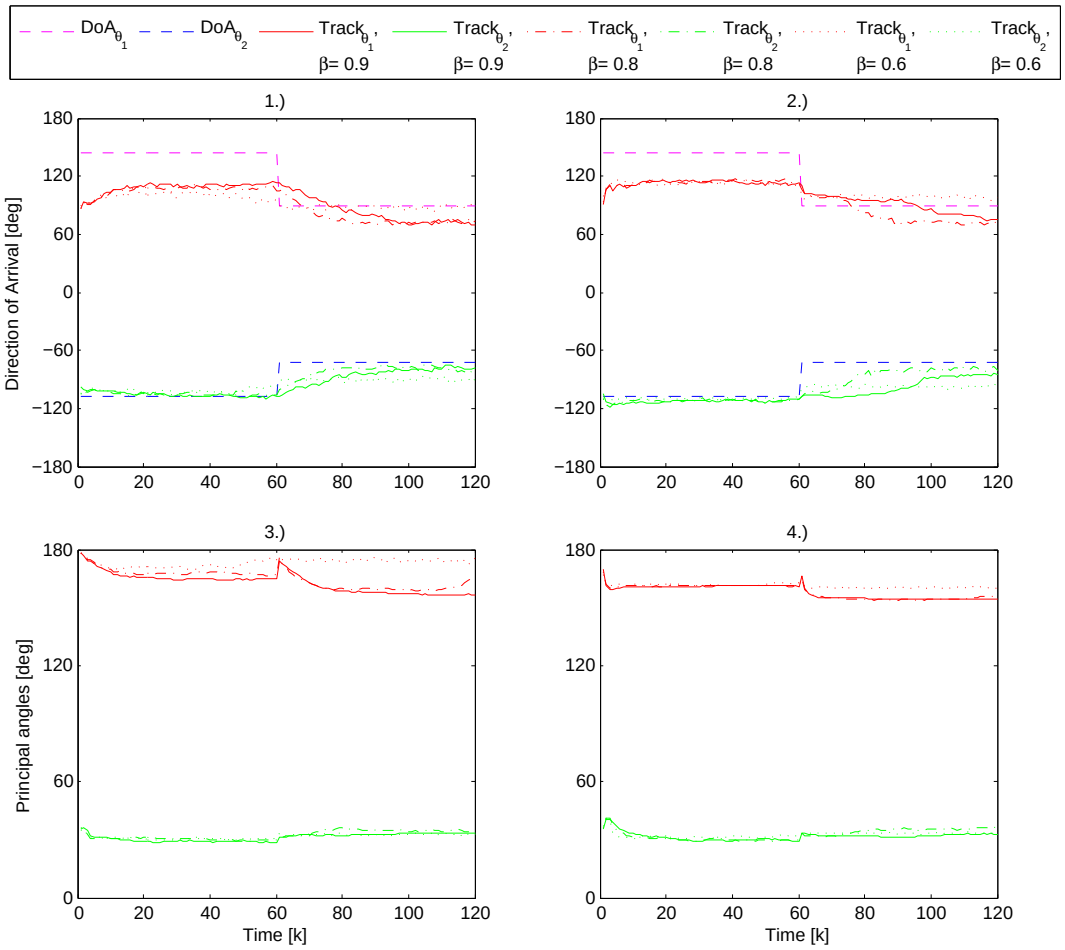

Figure A.7: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

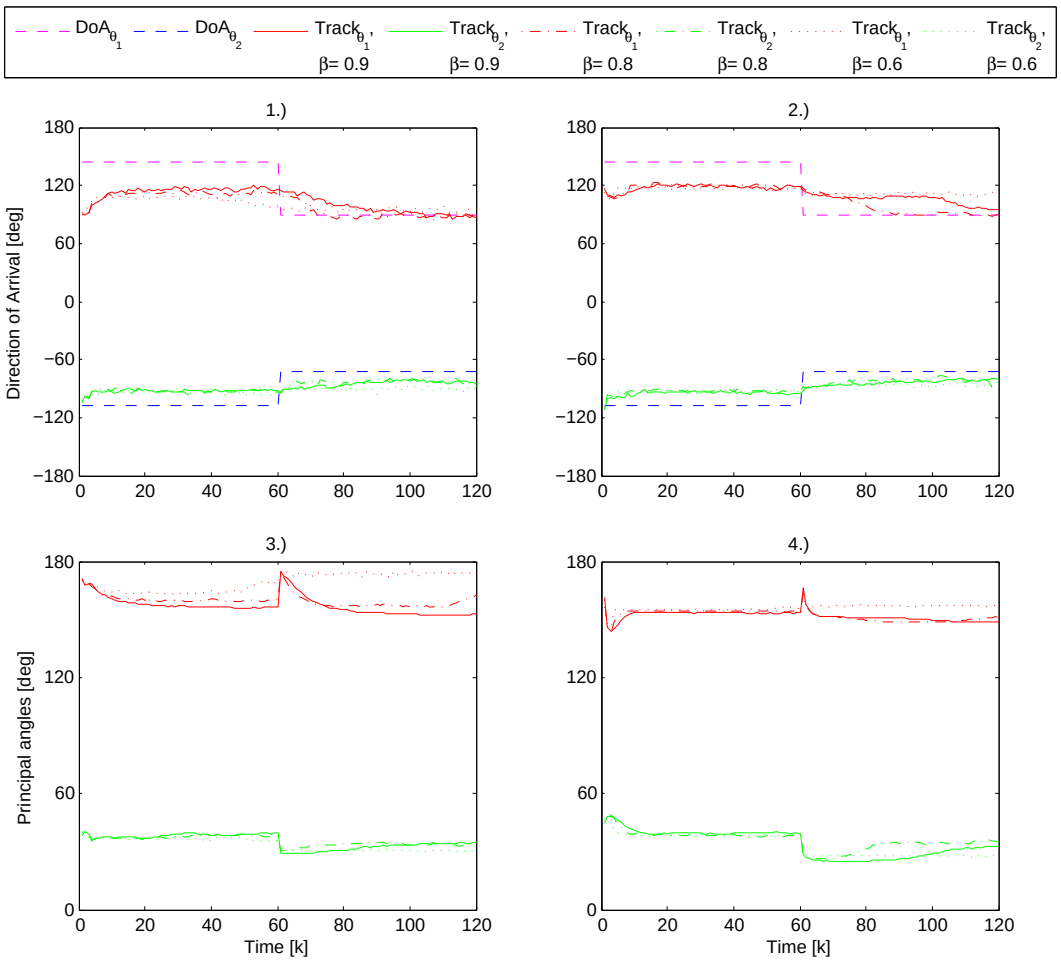

Figure A.8: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

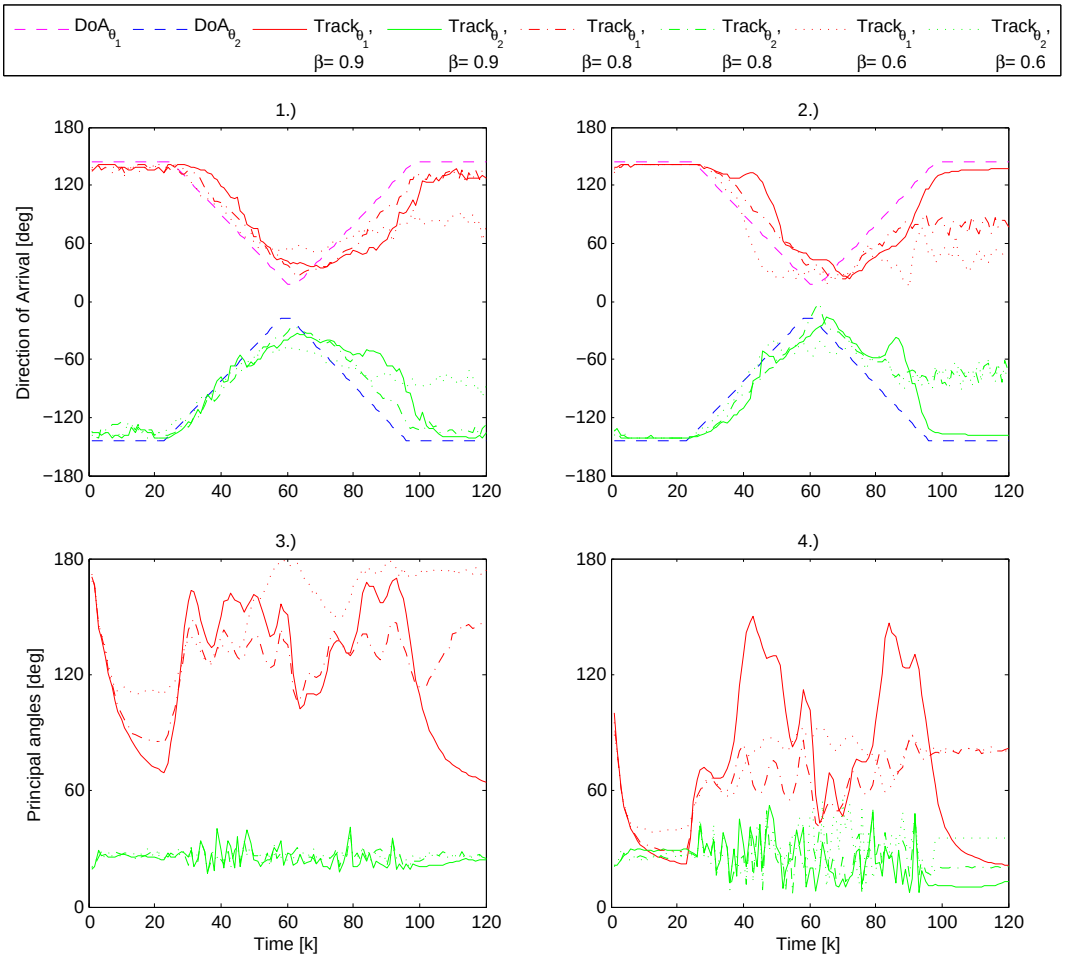

Figure A.9: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

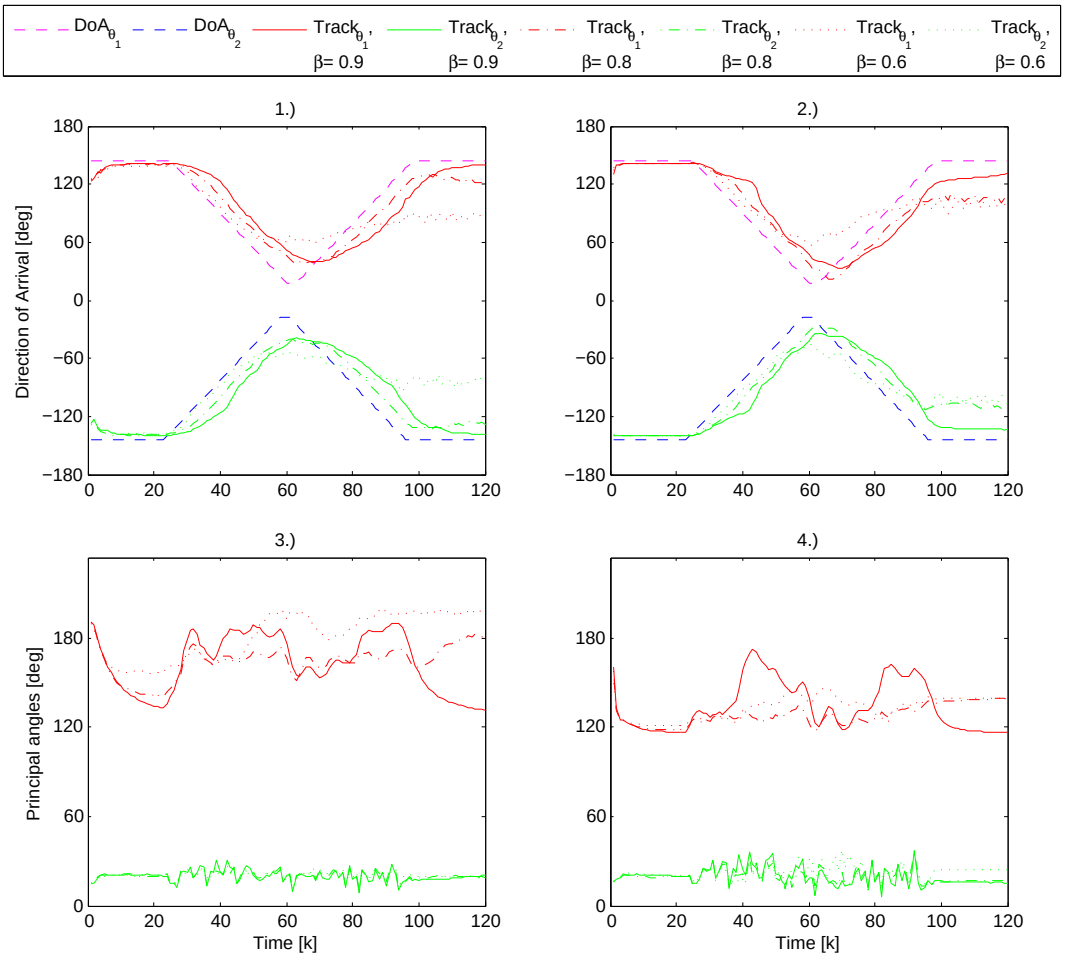

Figure A.10: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

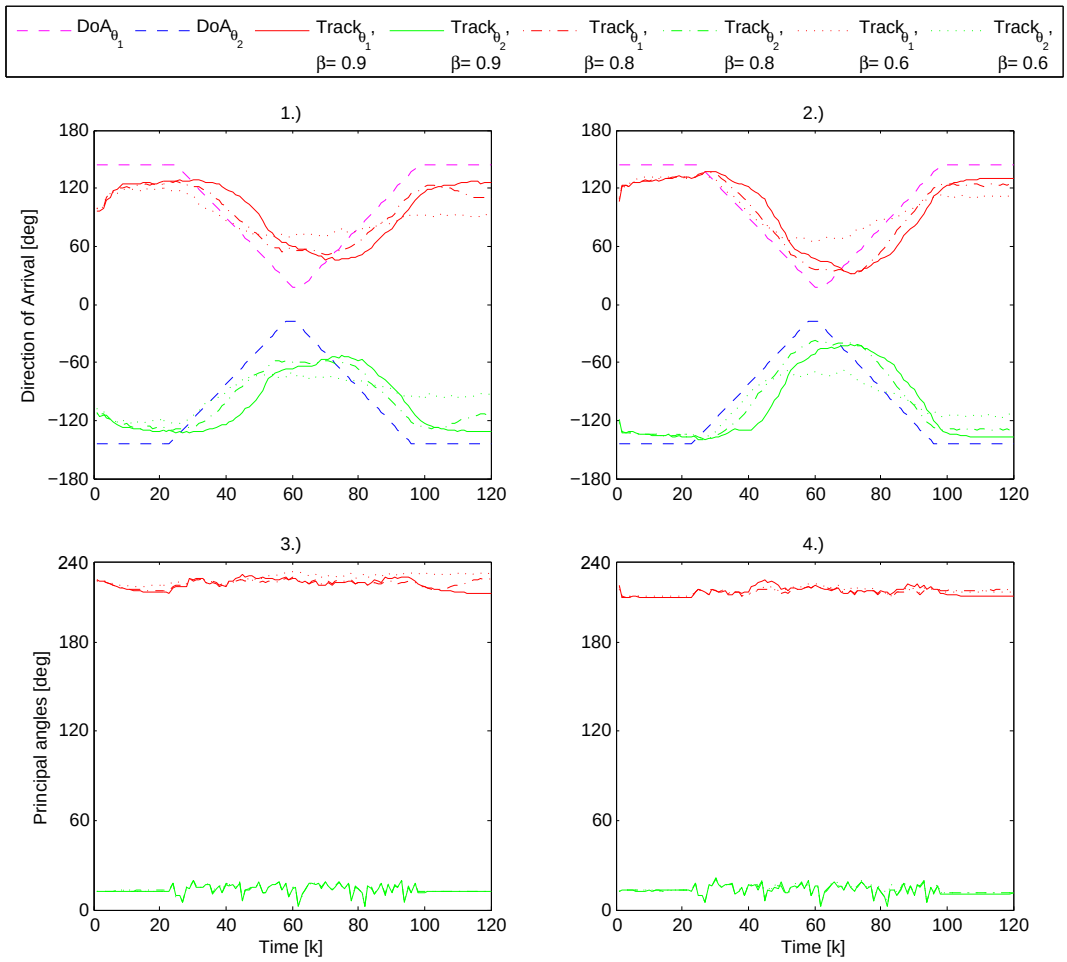

Figure A.11: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

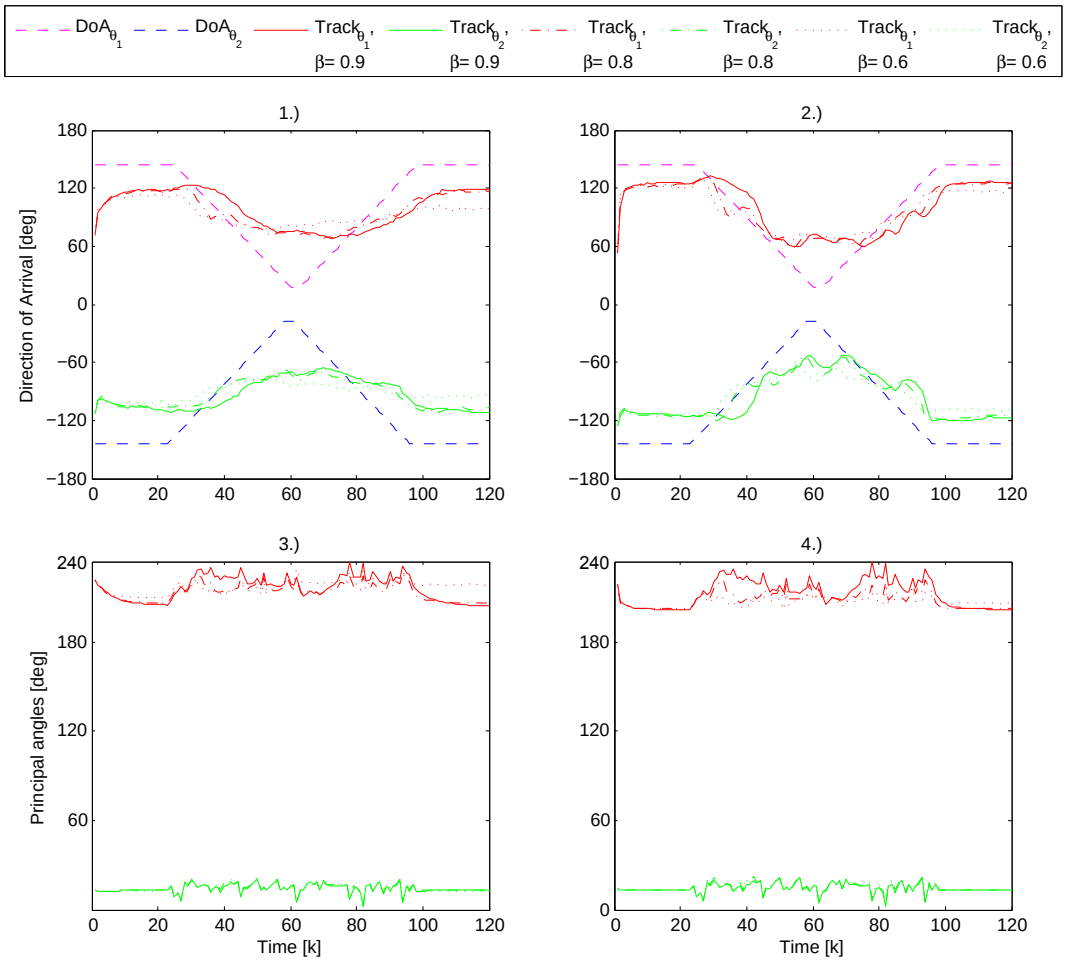

Figure A.12: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

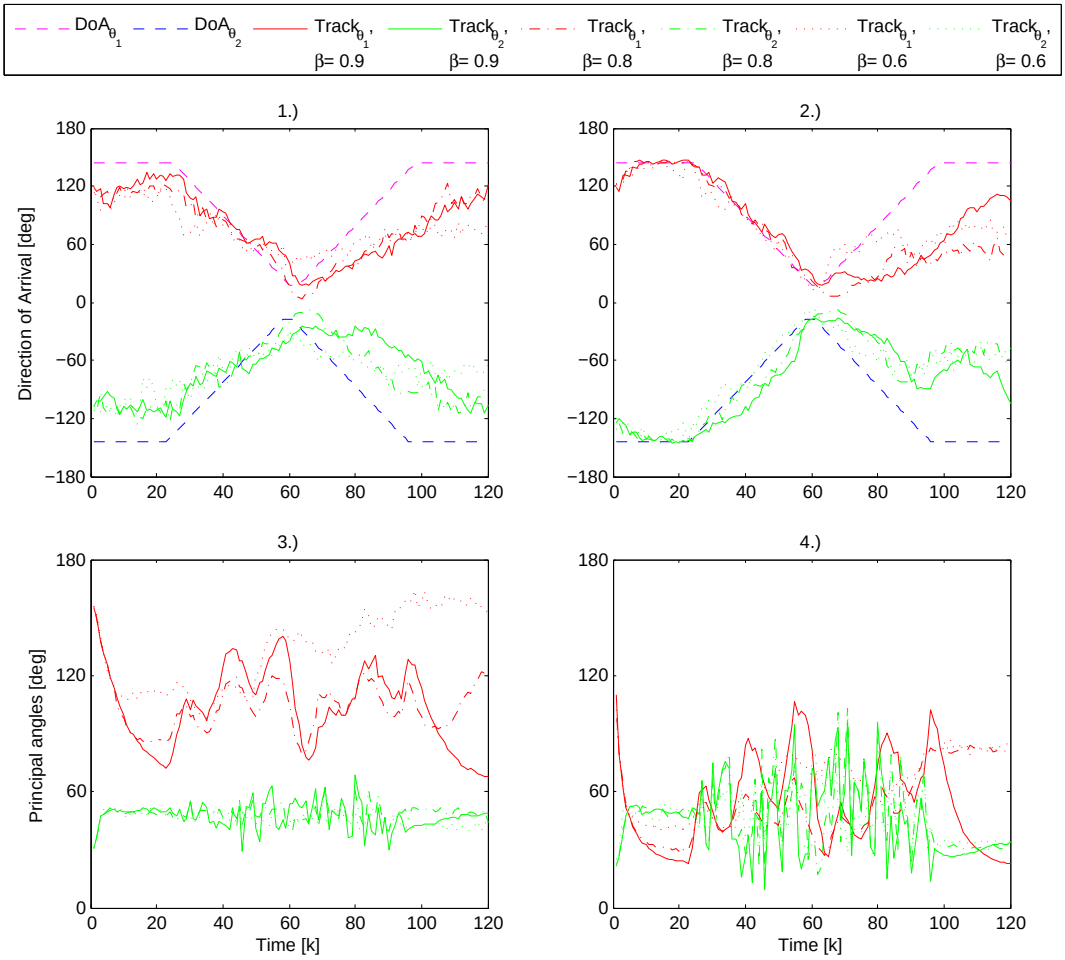

Figure A.13: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

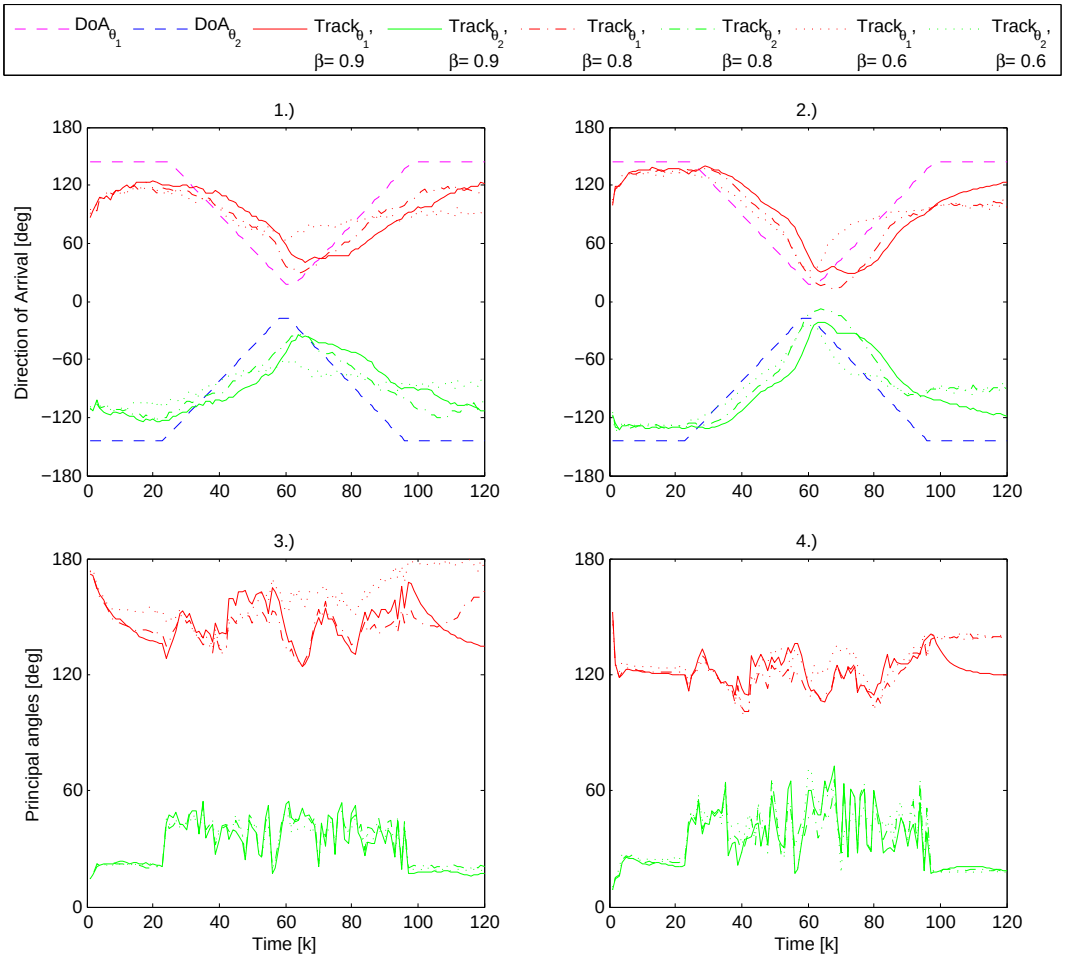

Figure A.14: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

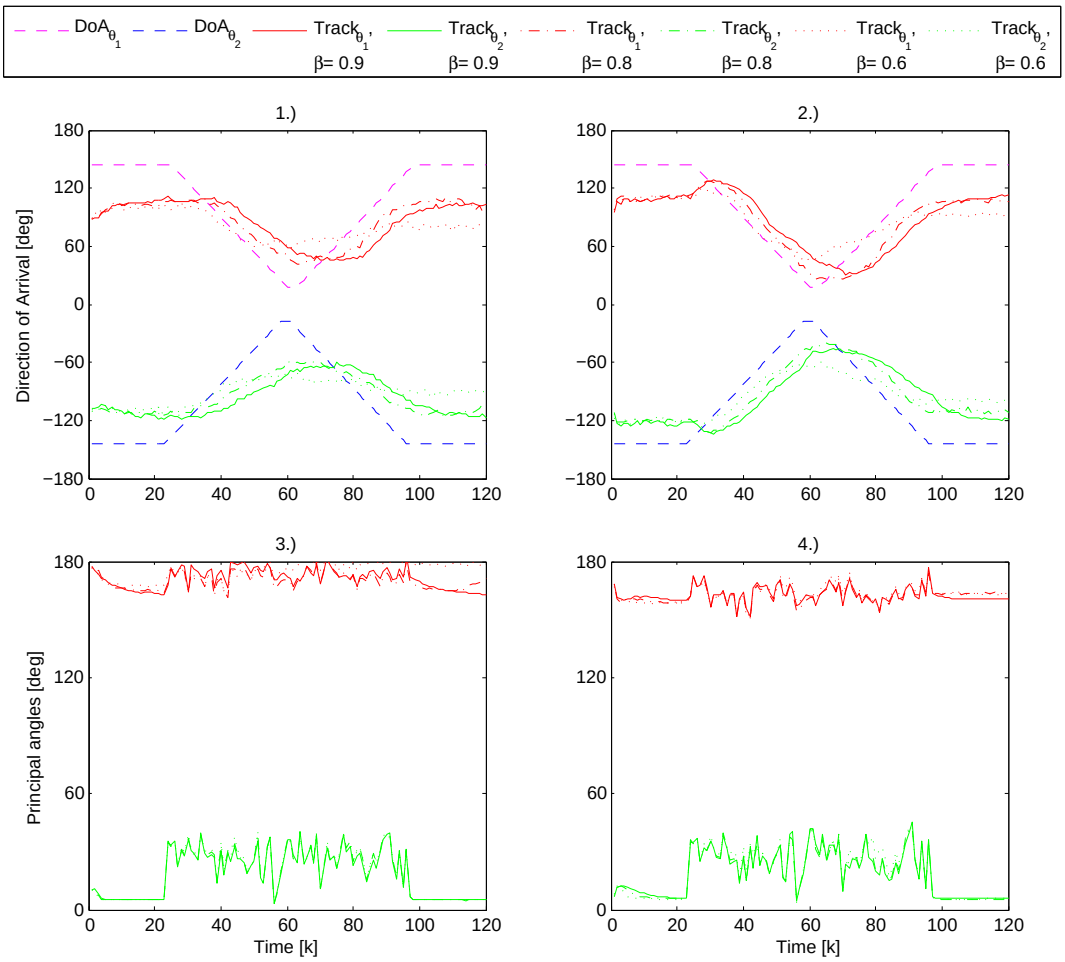

Figure A.15: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

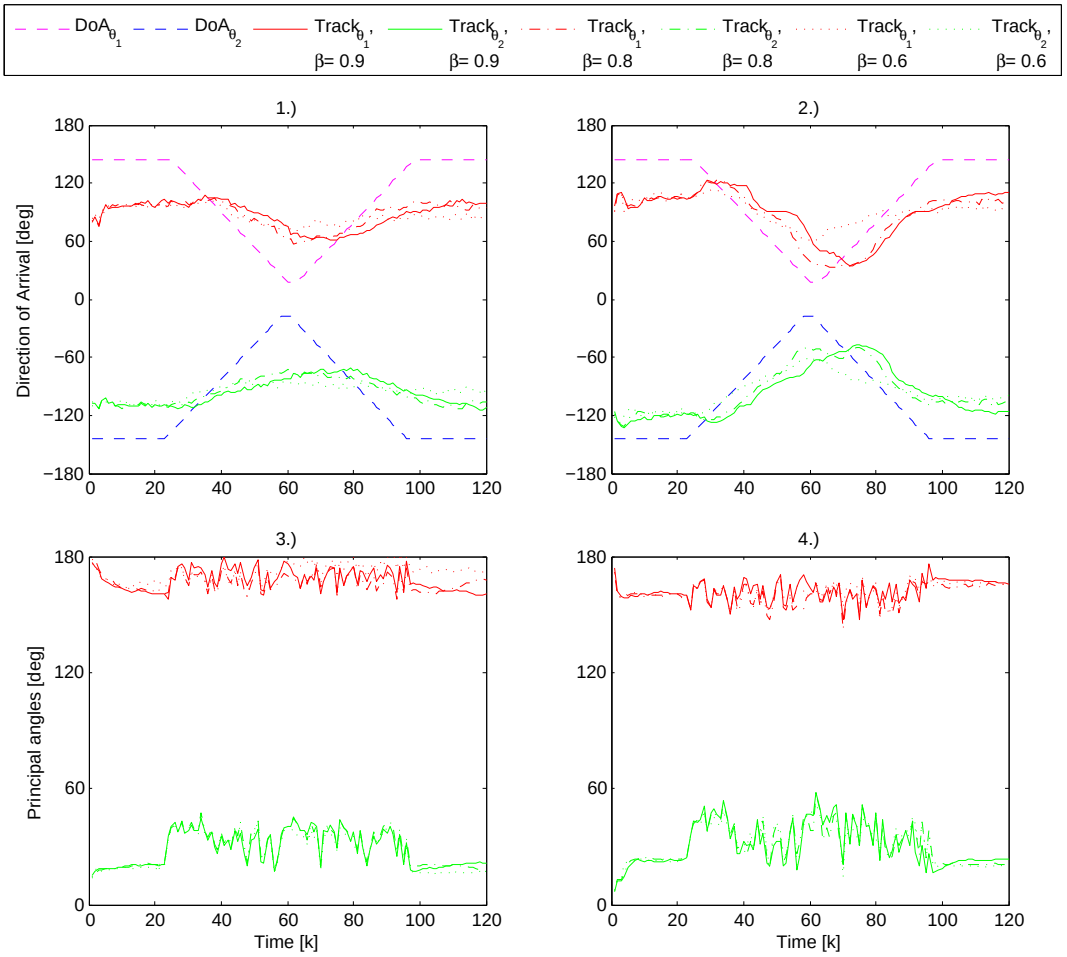

Figure A.16: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 2 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

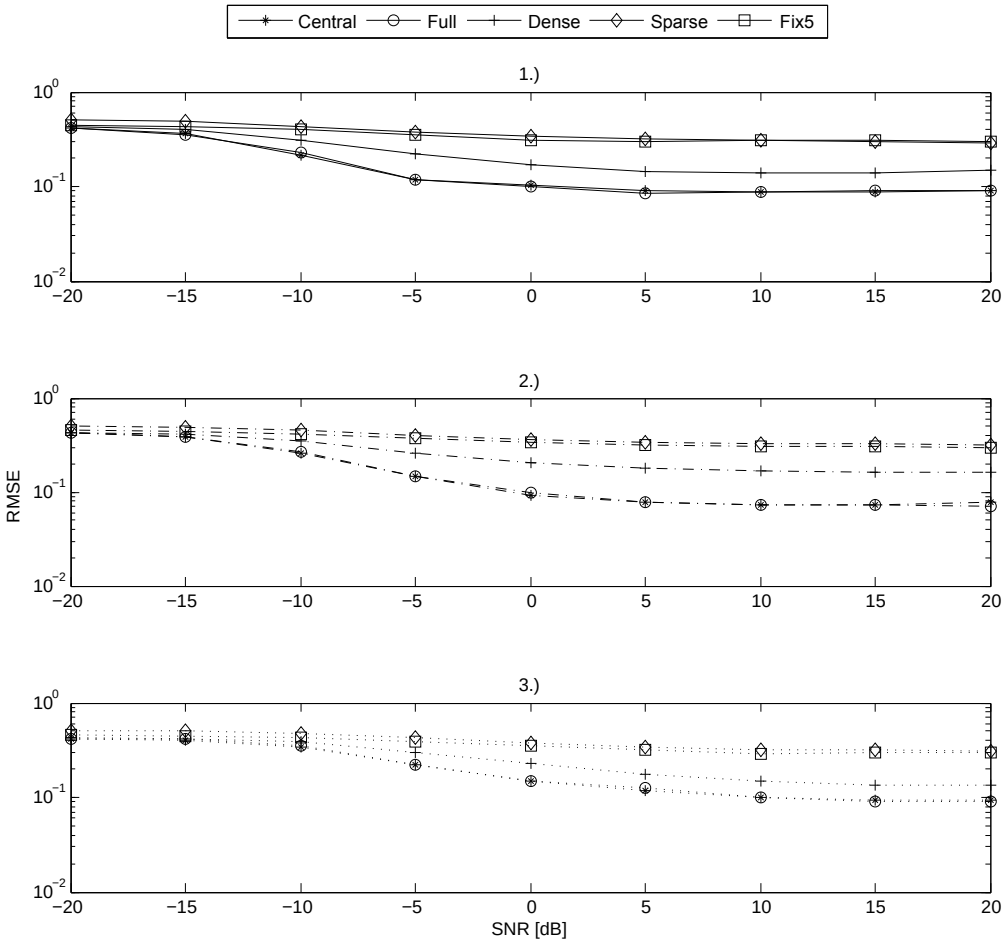

Figure A.17: Evaluation of the Root Mean Square Error for the step signals as in 4.6. The Algorithm 2 is evaluated for N=12 sensors and for all possible topologies from Figure 2.3. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

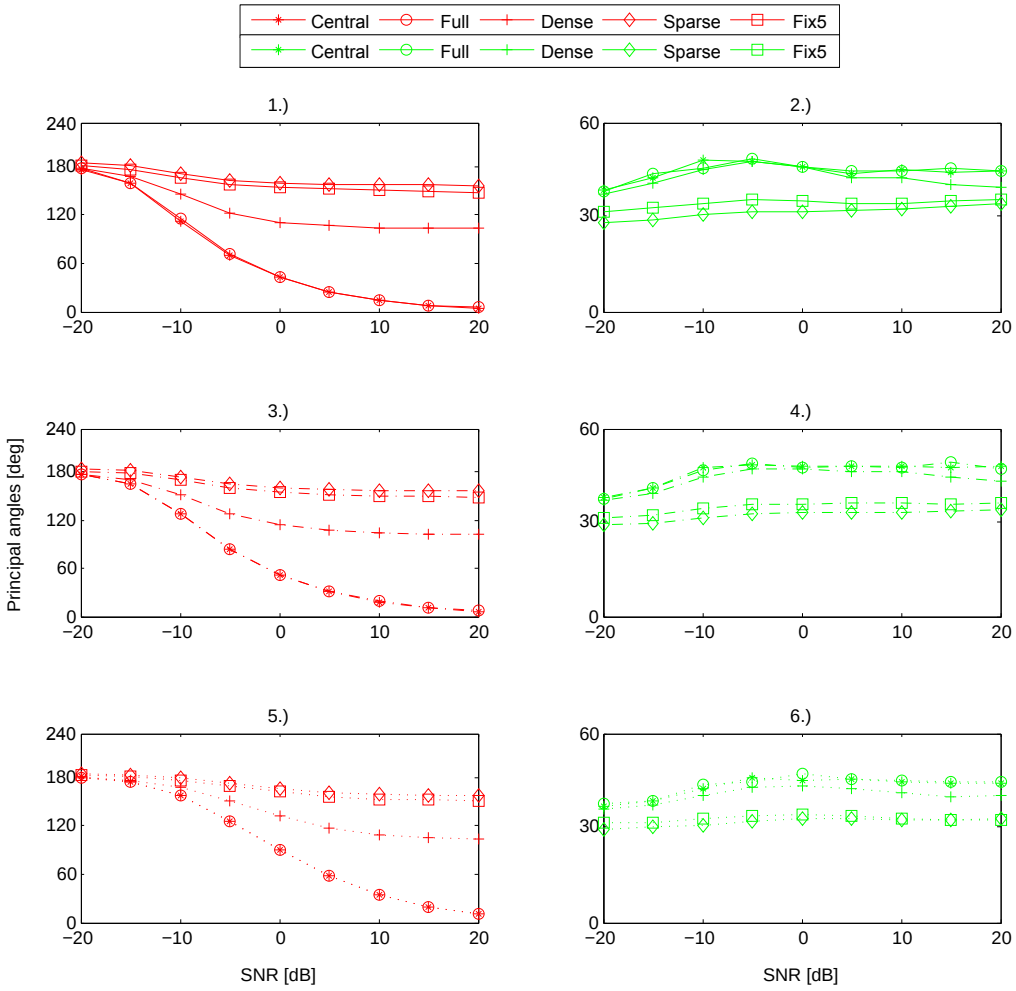

Figure A.18: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the step signals. This cases regards the step signal case. The Algorithm 2 is analyzed for the topologies proposed in Figure 2.3. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

Appendix B

Simulation results when $\widehat{\mathbf{R}}$ is distributed

In this section, further figures to study the previously addressed scenarios are displayed. No more details are provided, as they are included only for comparison purposes. The reader is invited to have a deeper look into the behavior of Algorithm 3.

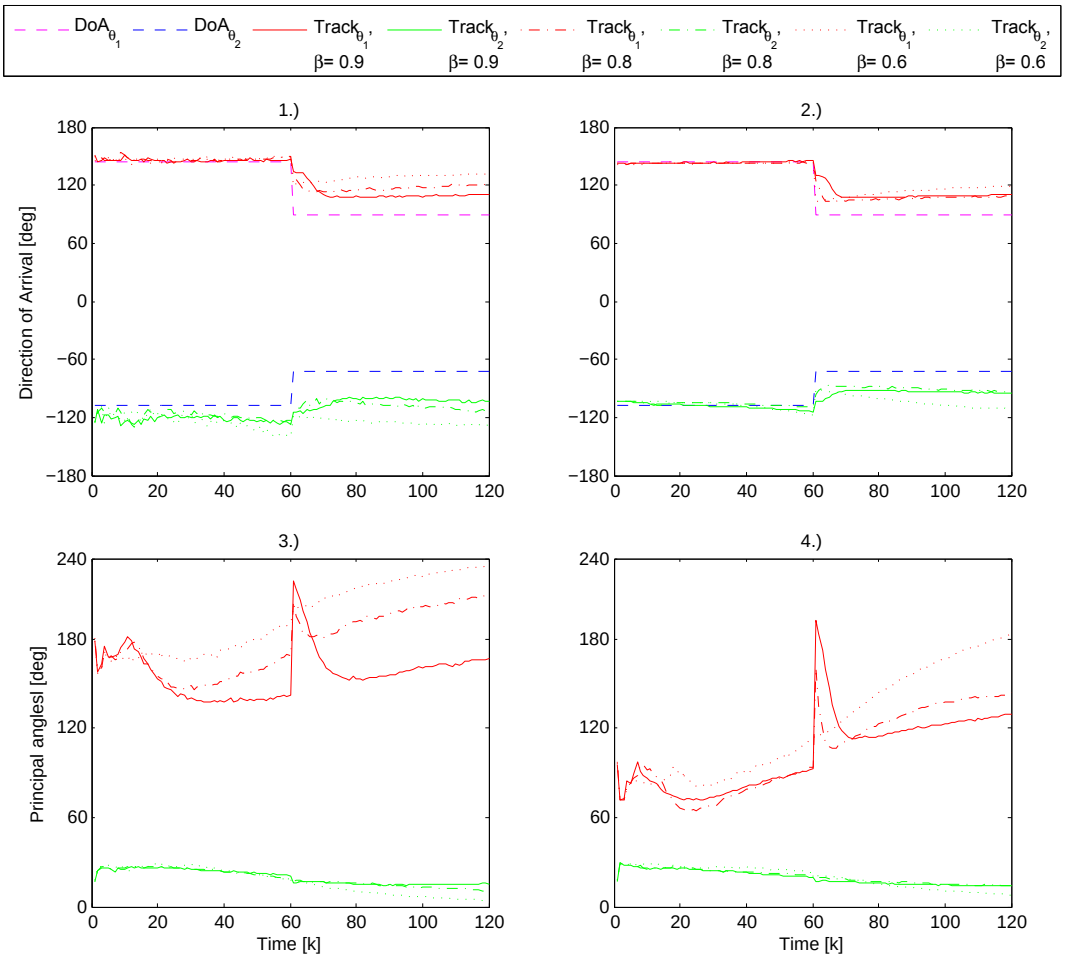

Figure B.1: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

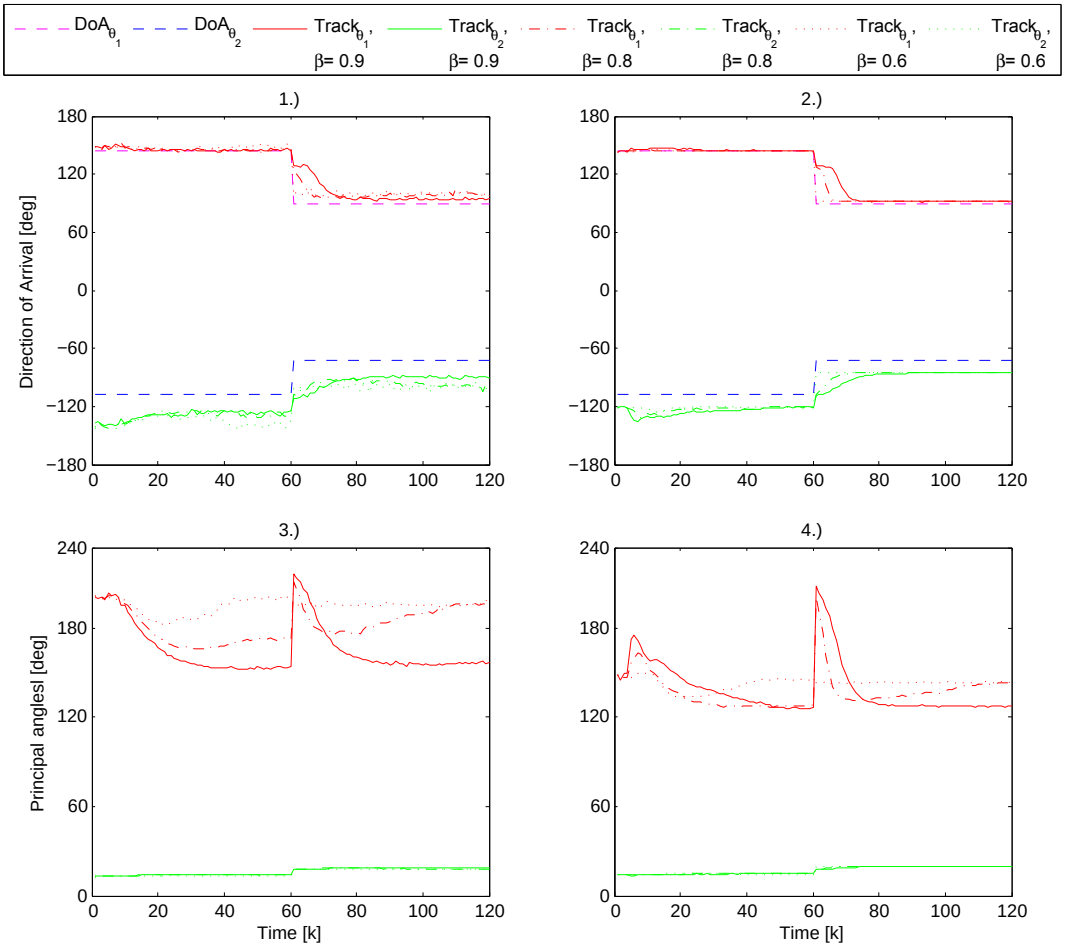

Figure B.2: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

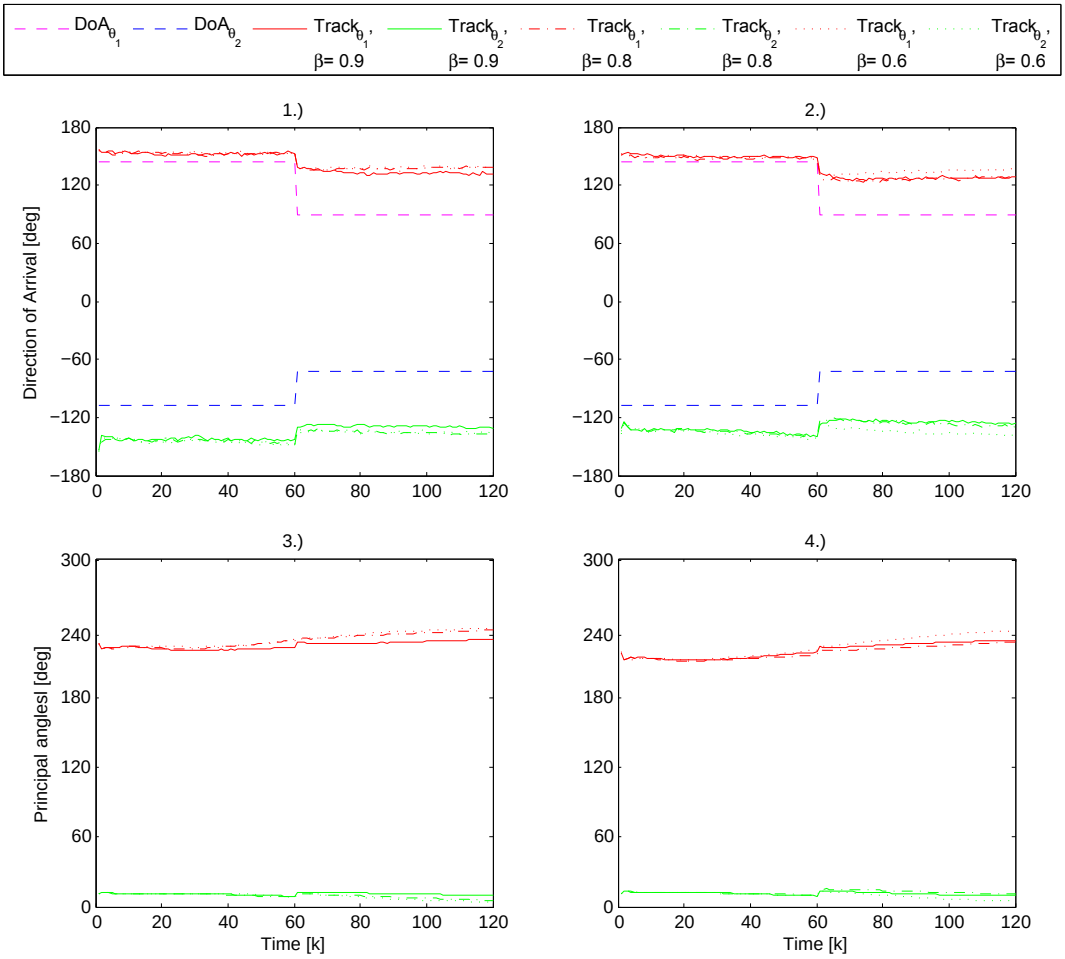

Figure B.3: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

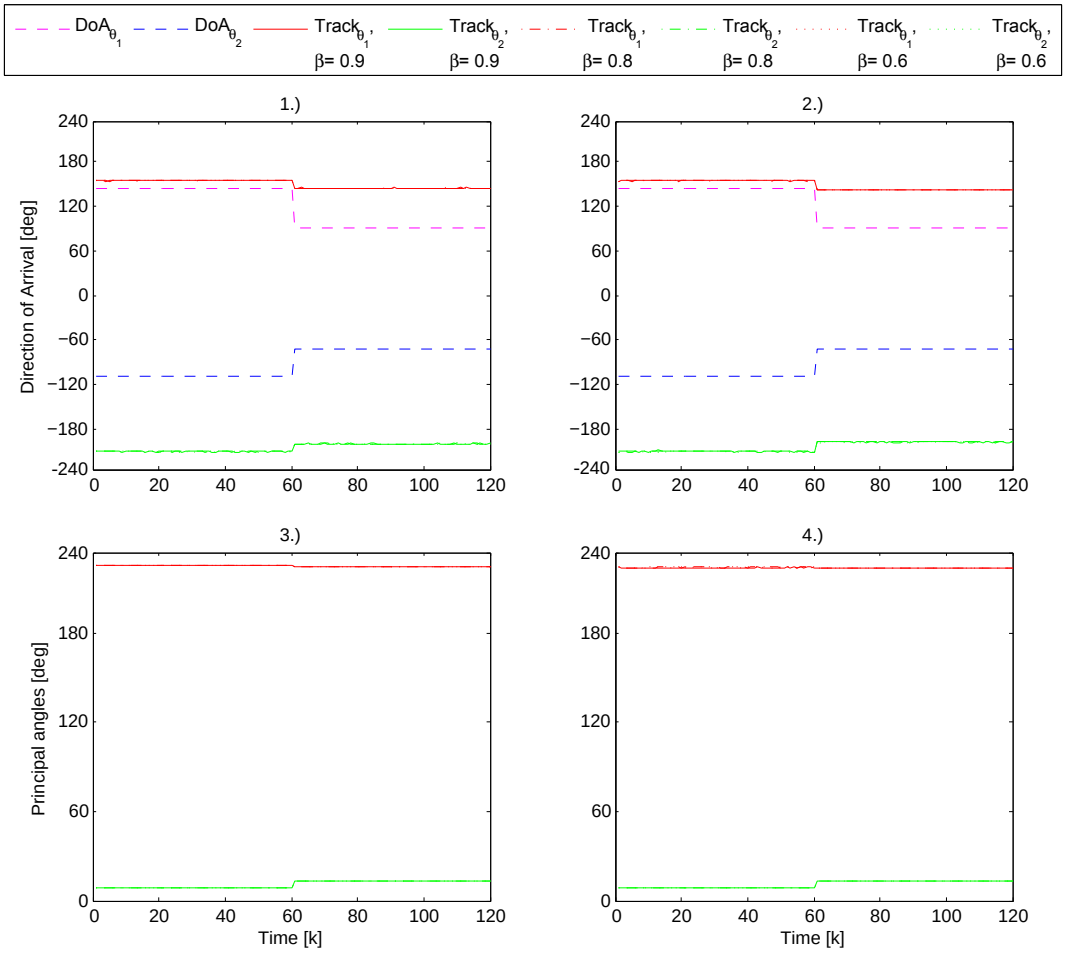

Figure B.4: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

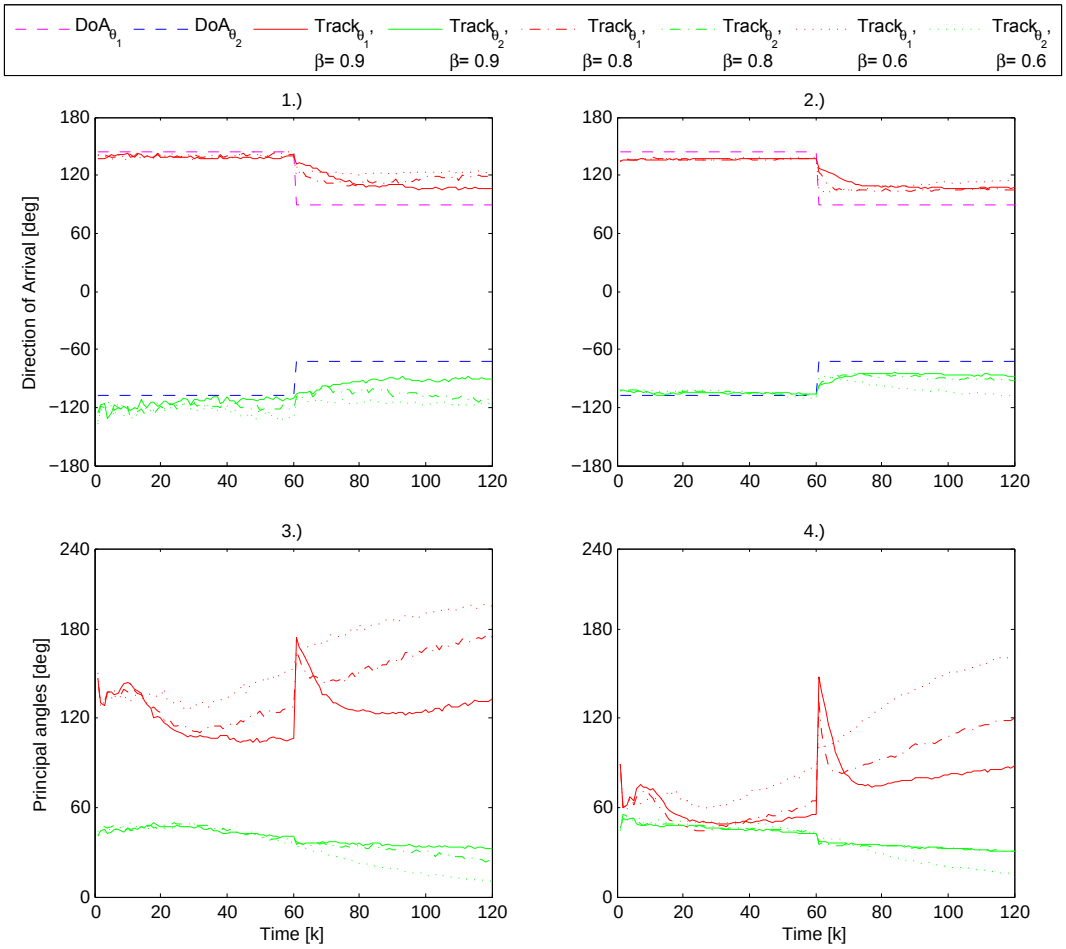

Figure B.5: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

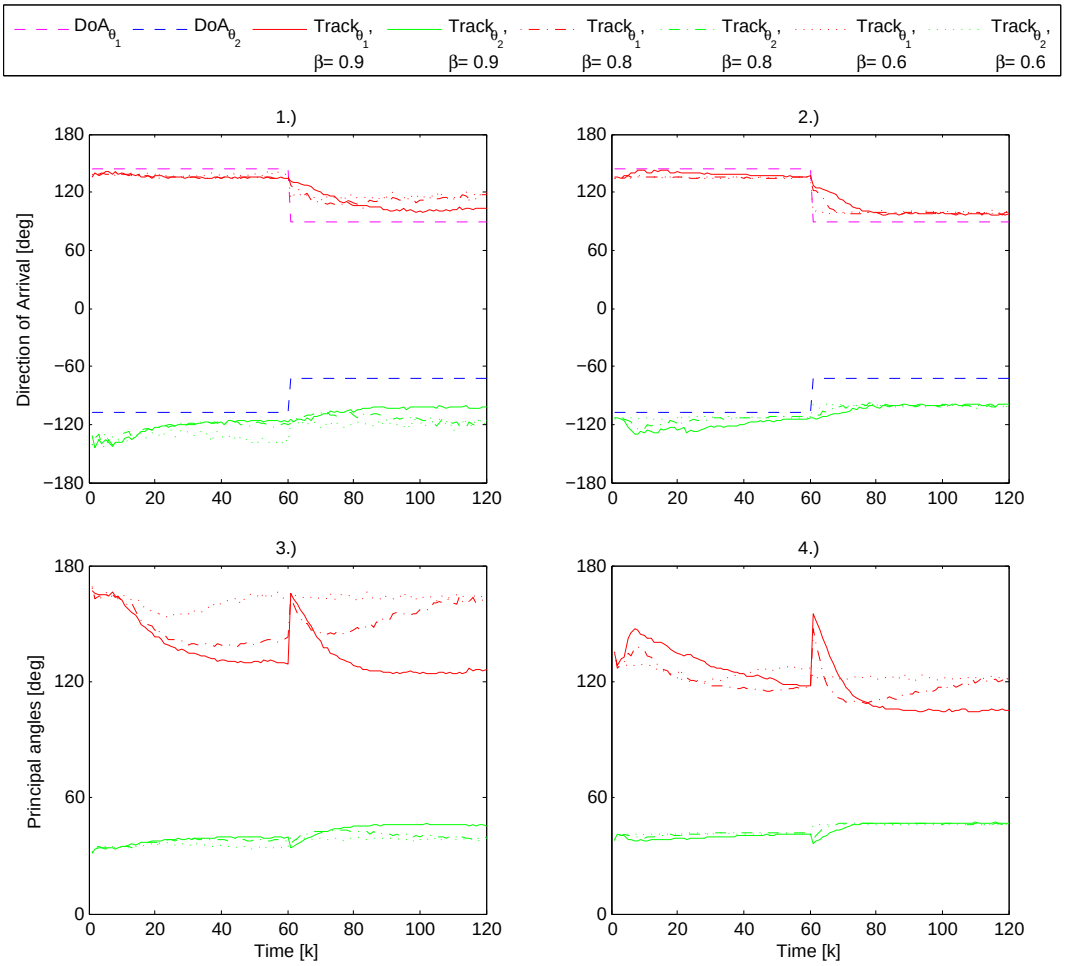

Figure B.6: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

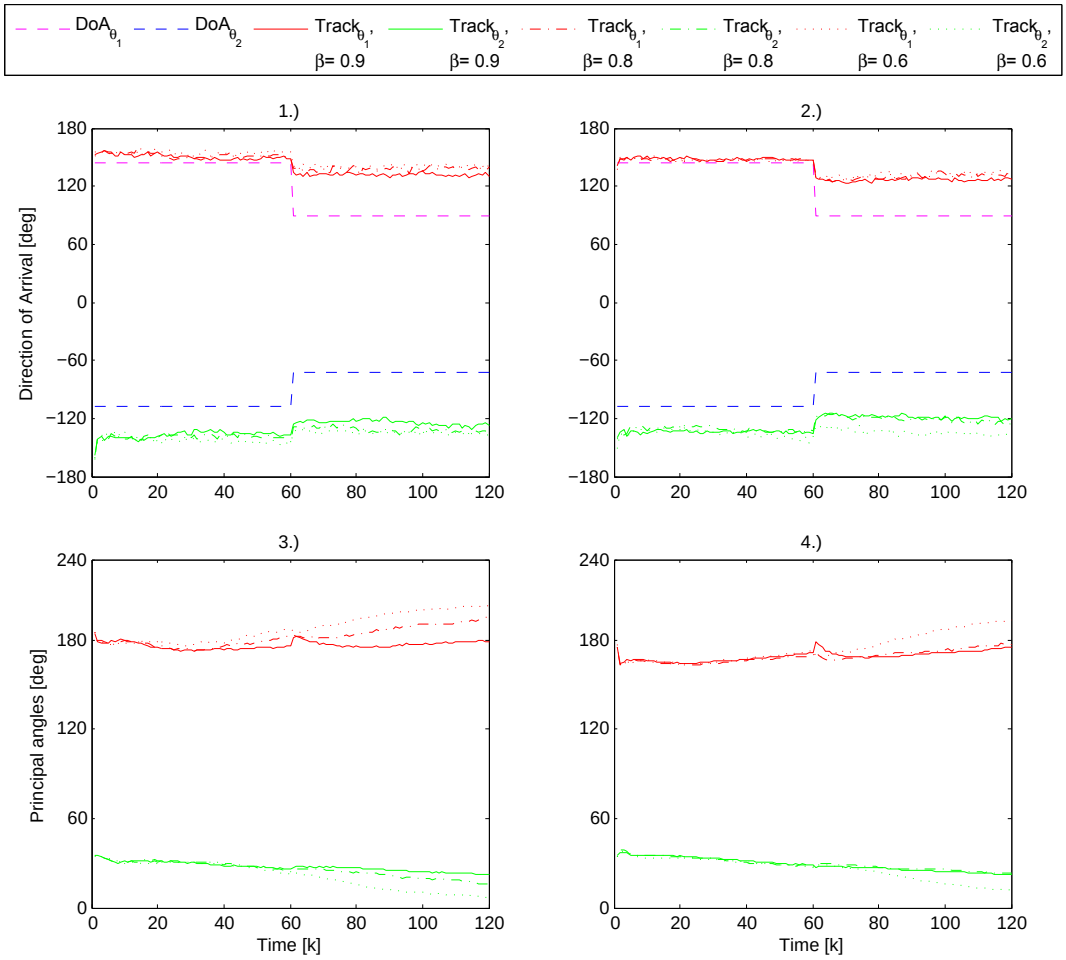

Figure B.7: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

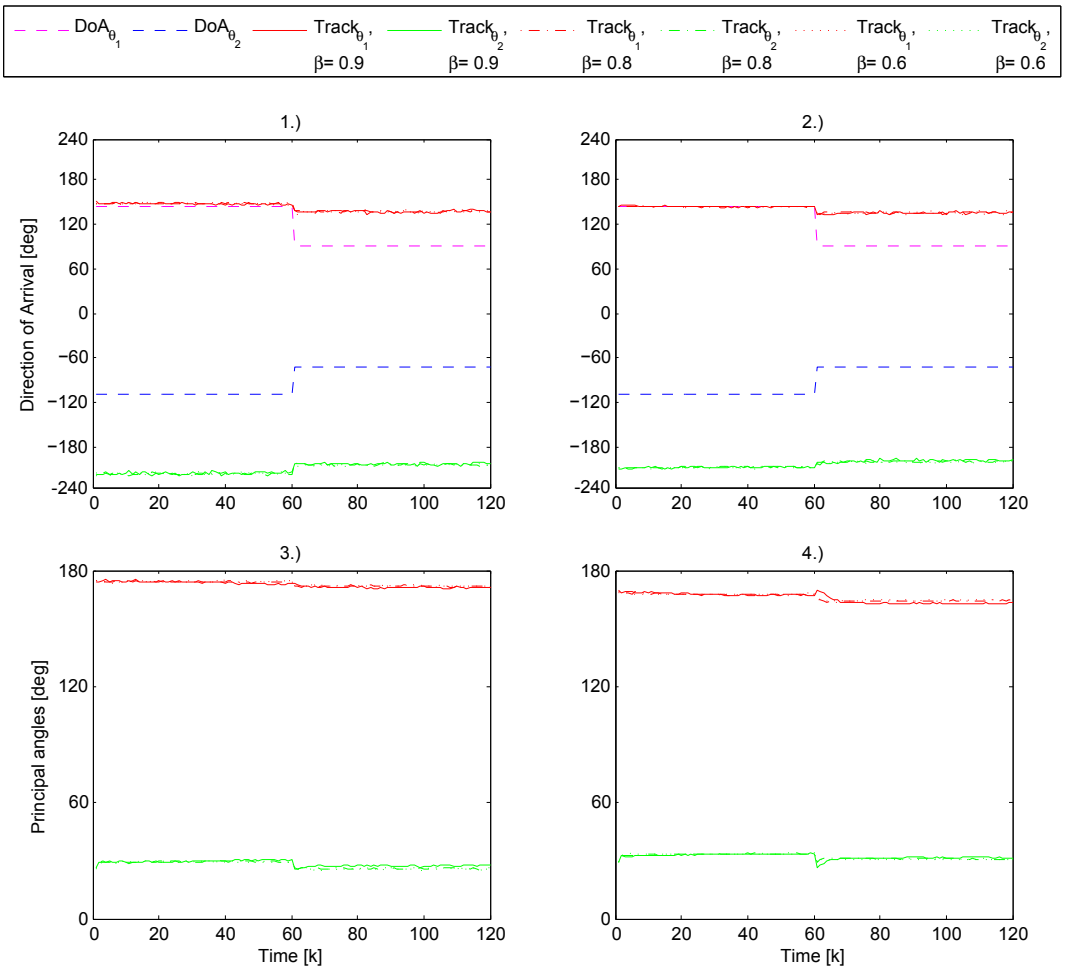

Figure B.8: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The forgetting factor is set to $\beta = 0.9, 0.8, 0.6$.

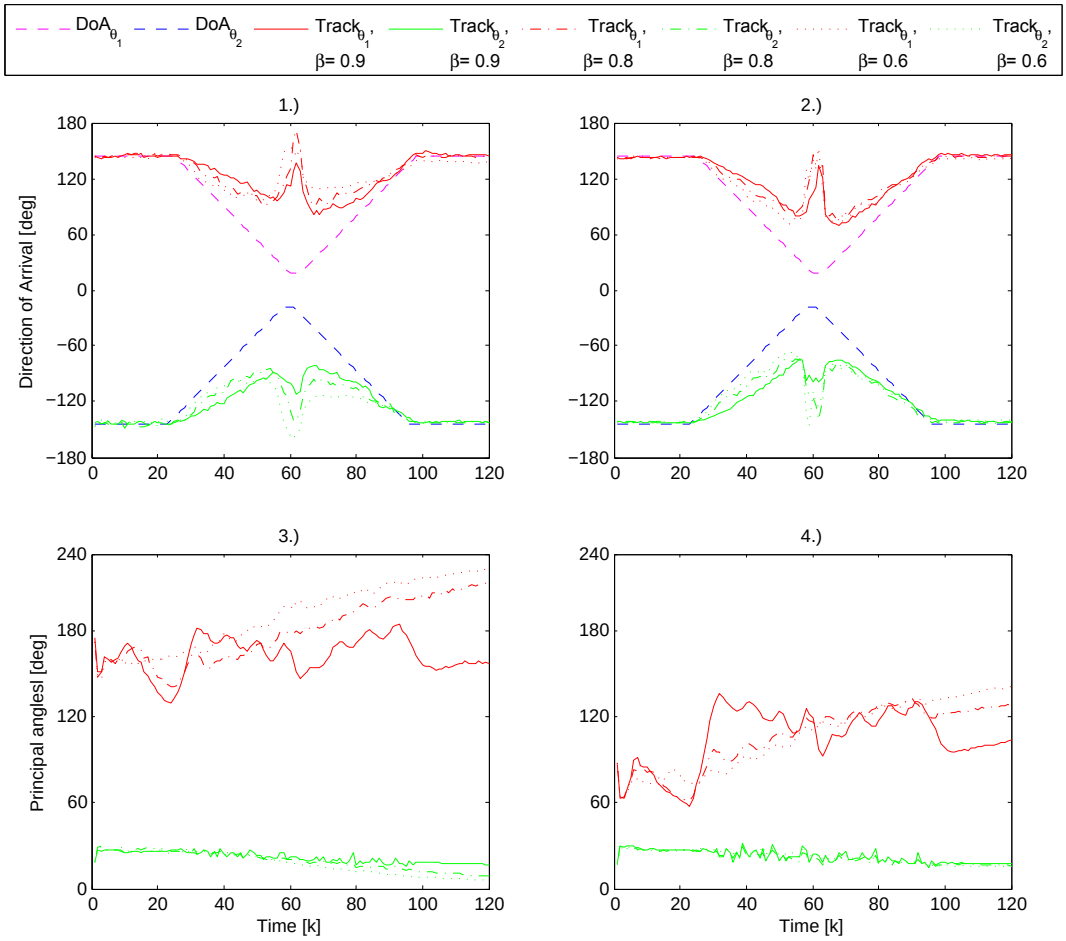

Figure B.9: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

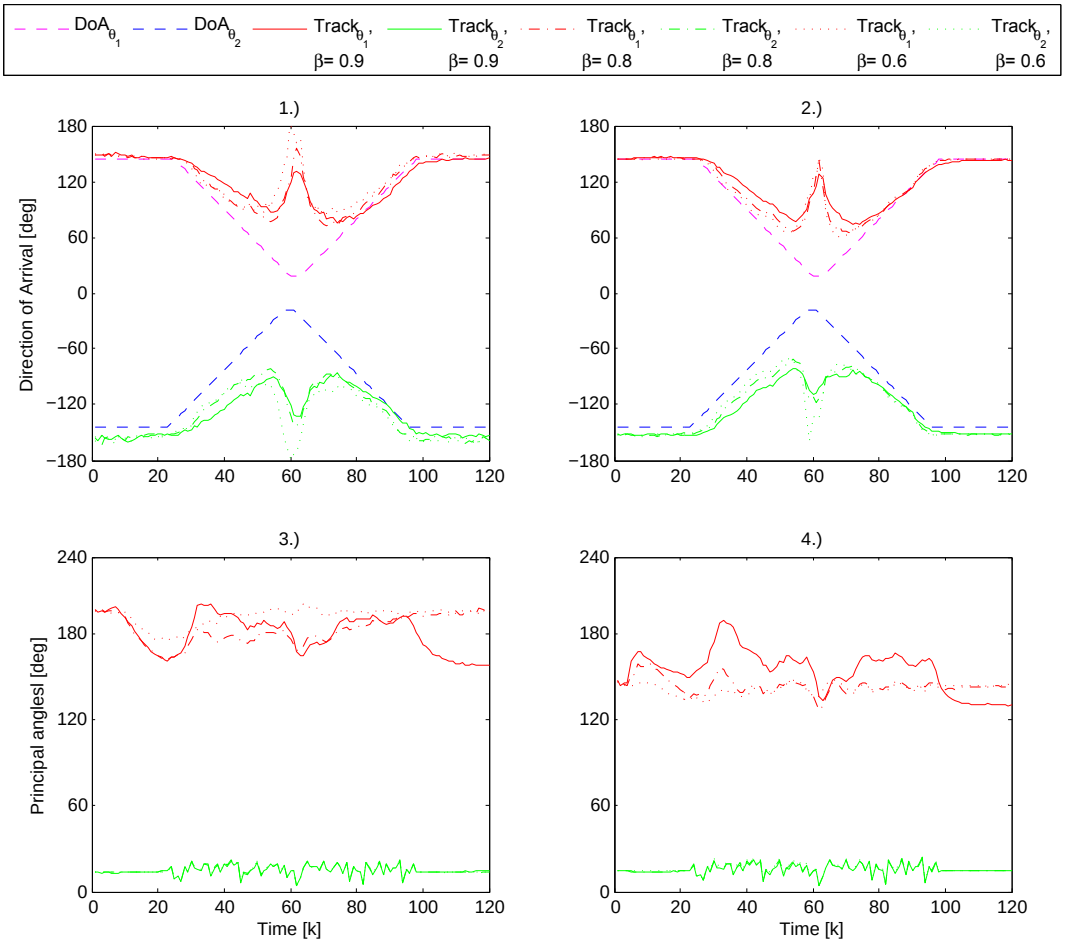

Figure B.10: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the dense connected scenario b) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

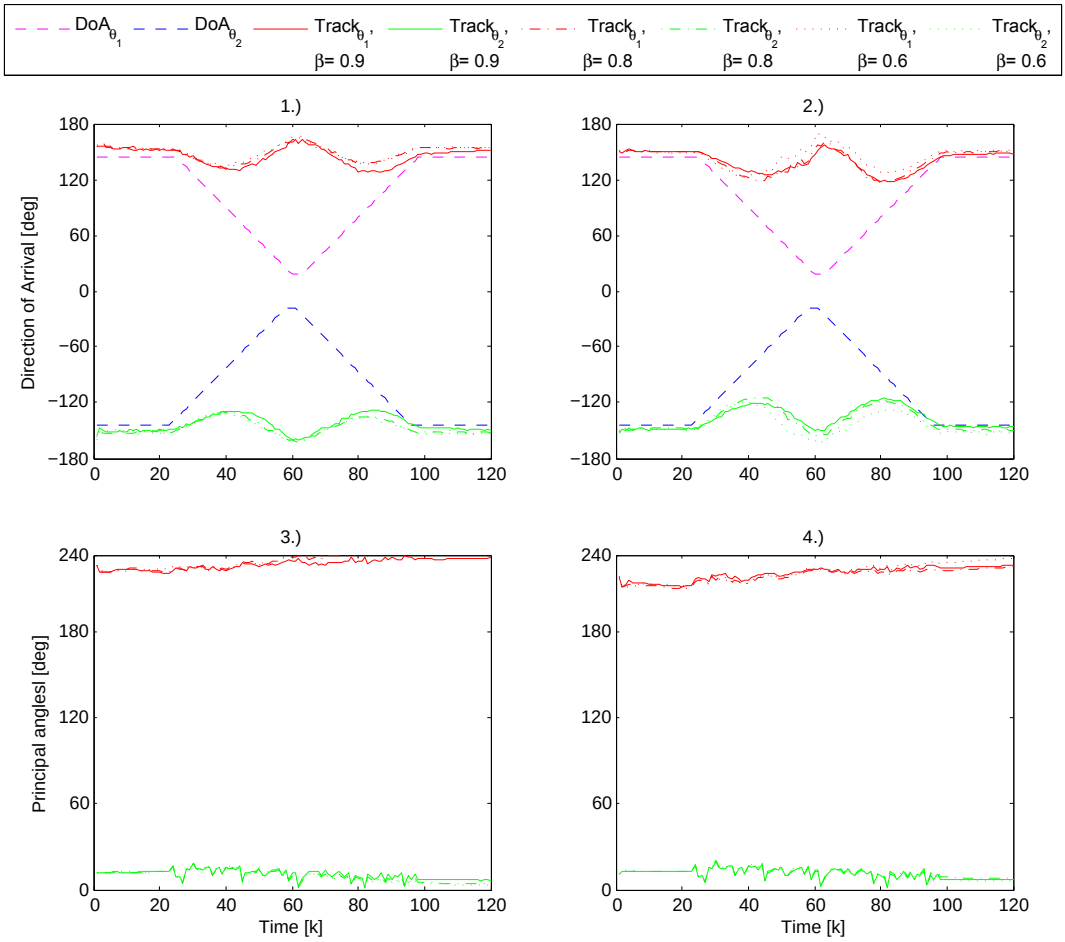

Figure B.11: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

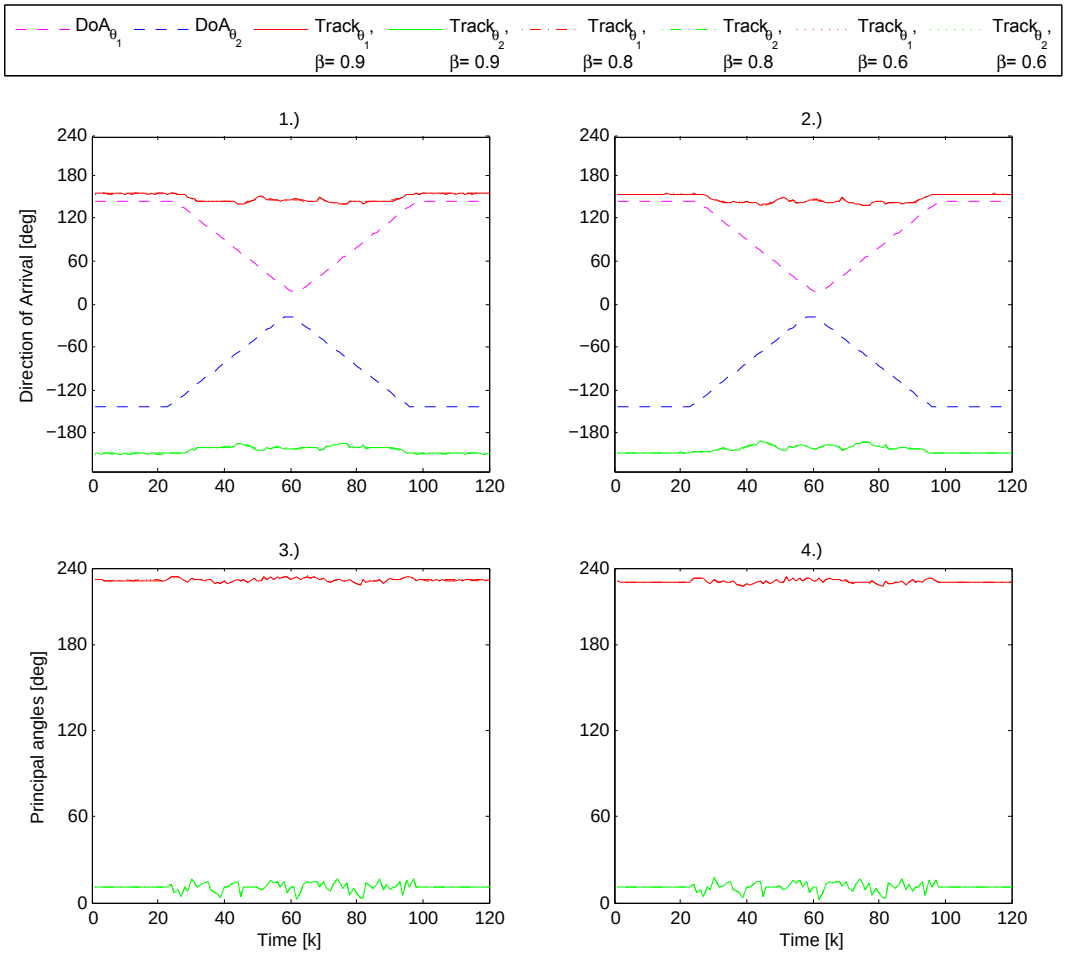

Figure B.12: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.4 for $N = 36$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

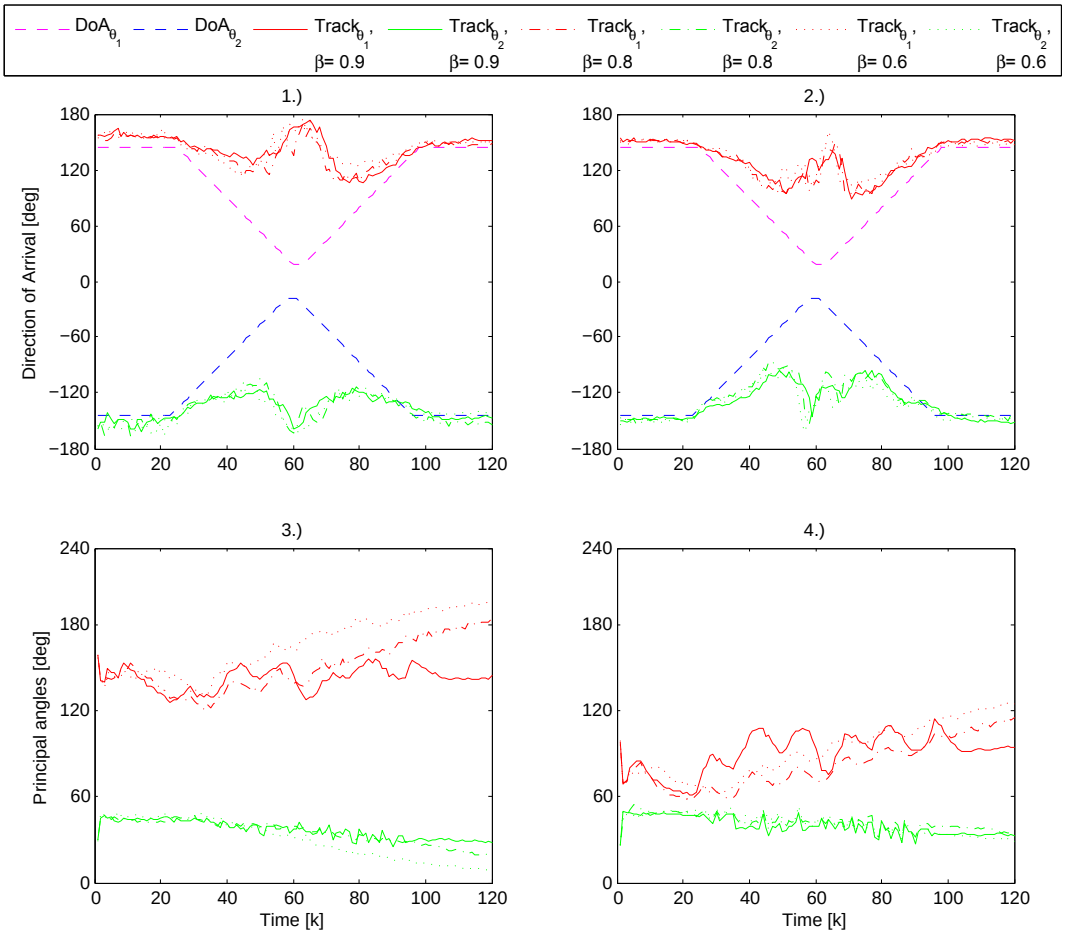

Figure B.13: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the fully connected scenario a) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

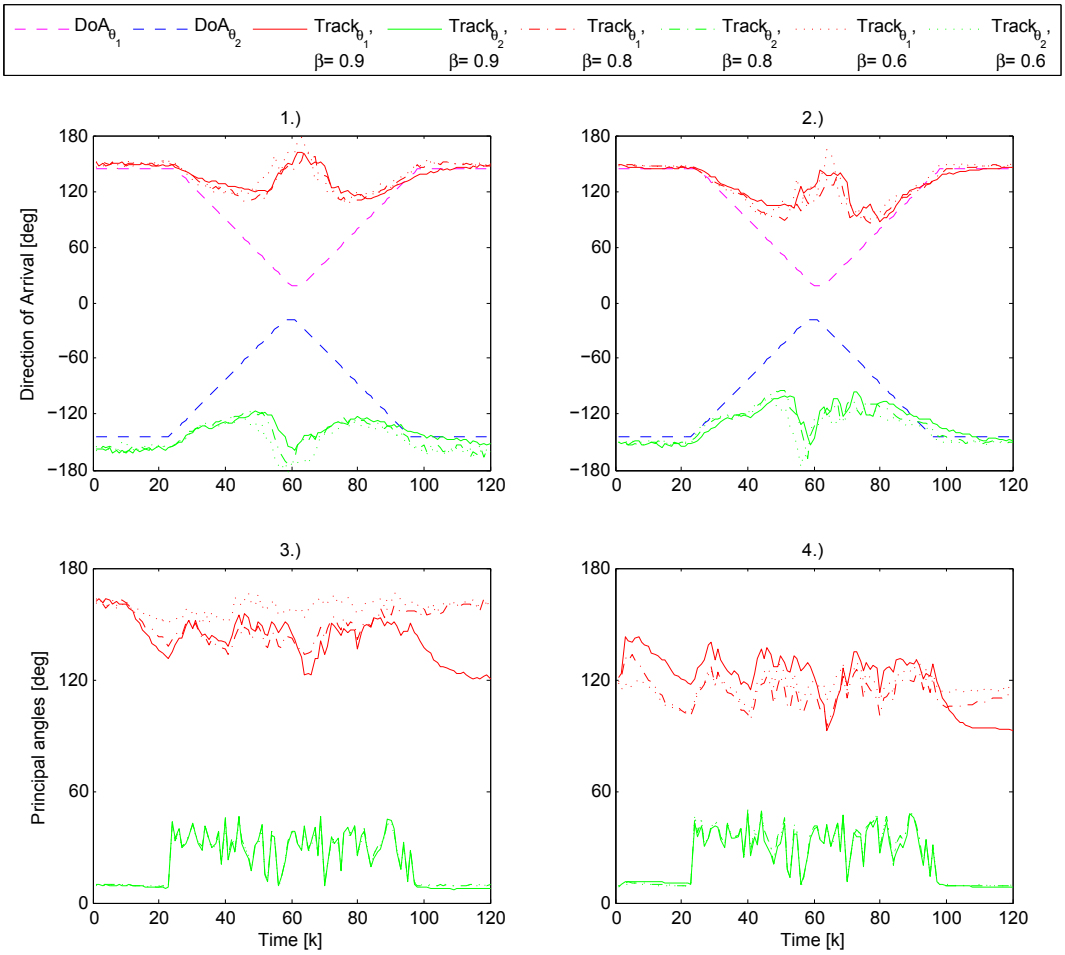

Figure B.14: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in a dense connected scenario b) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

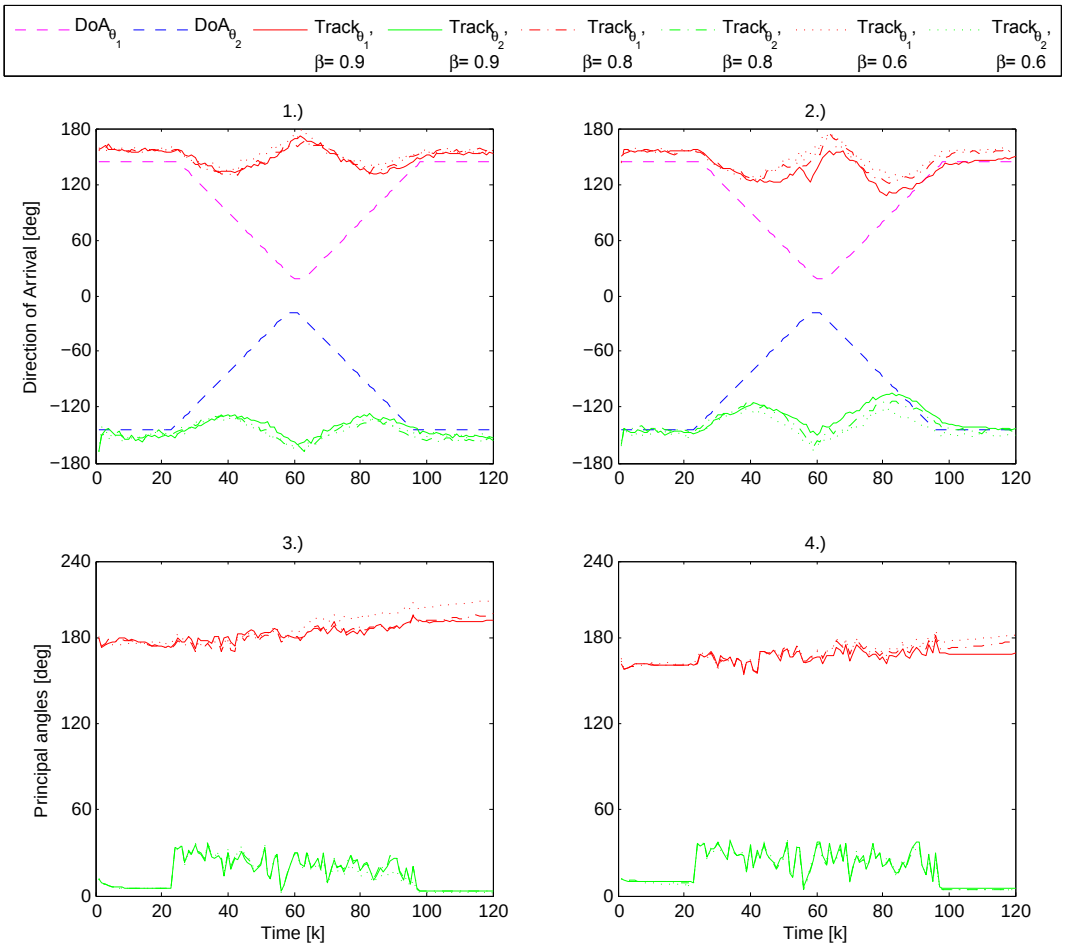

Figure B.15: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in the sparse connected scenario c) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

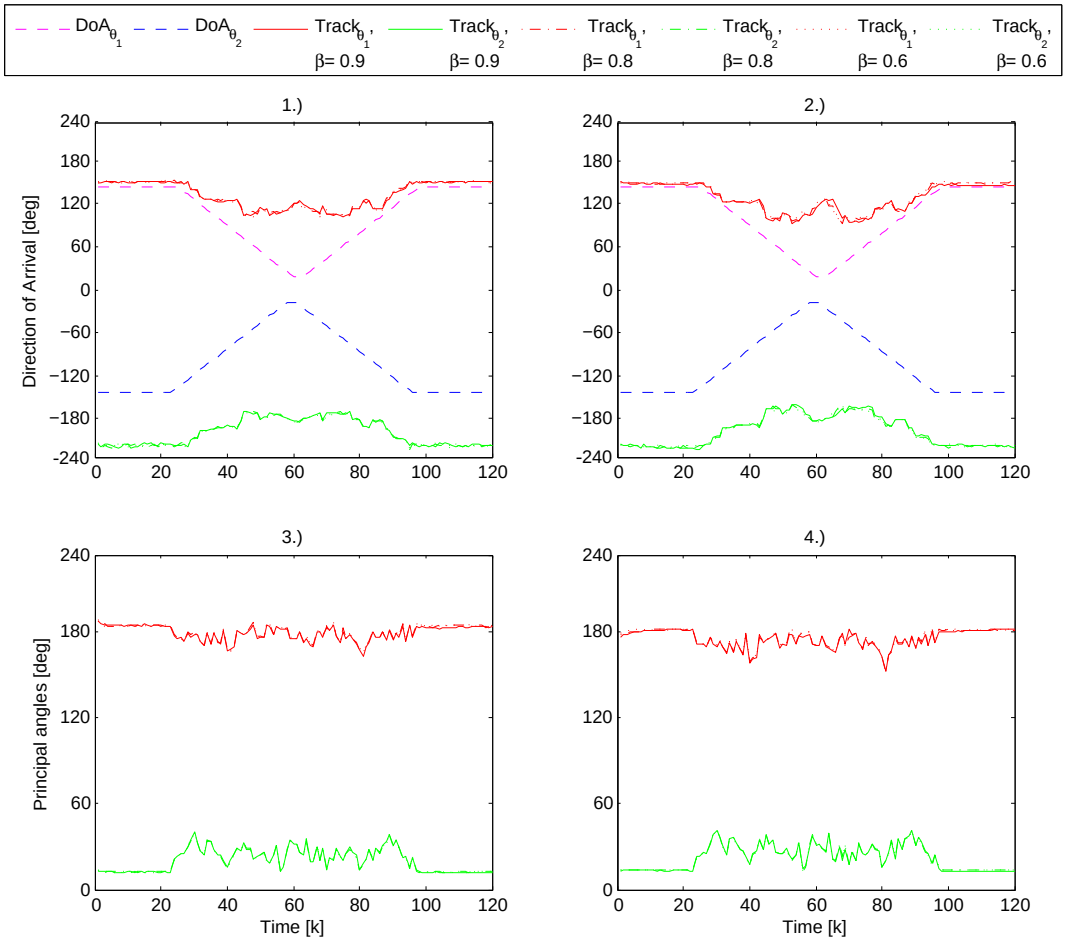

Figure B.16: Subfigure 1.) and 2.) display the tracking capabilities of Algorithm 3 when $\underline{\mathbf{y}}_i$ is distributed in a network with regular neighborhood size, i.e., $\mathcal{N}_i = 5$. This is the scenario d) from Figure 2.3 for $N = 12$ sensor elements. Subfigures 3.) and 4.) present the angle error between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$. The SNR=-5dB for those left side subfigures and SNR=5dB for those at the right. The impact of the forgetting factor when $\beta = 0.9, 0.8, 0.6$ is likewise showed.

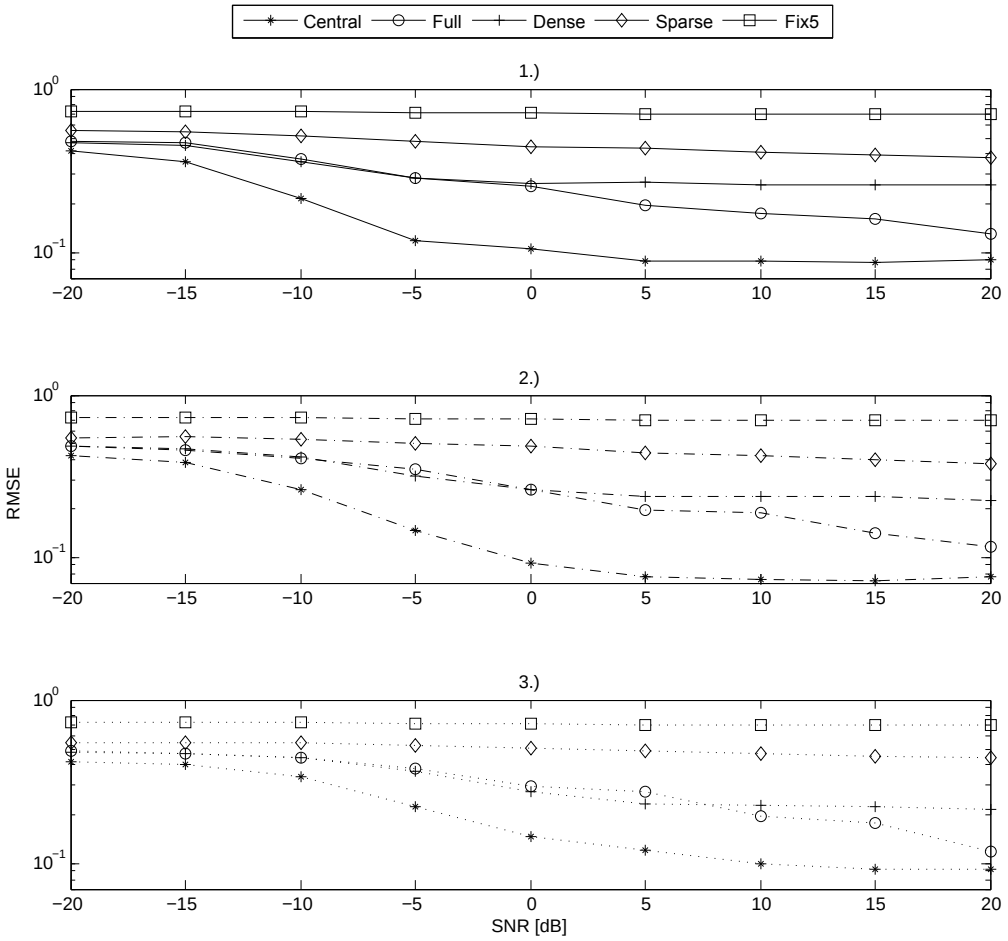

Figure B.17: Evaluation of the Root Mean Square Error for the step signals as in 4.6. The Algorithm 3 is evaluated for N=12 sensors and for all possible topologies from Figure 2.3. Subfigures 1.) , 2.) and 3.) relate to $\beta = 0.9, 0.8, 0.6$, respectively.

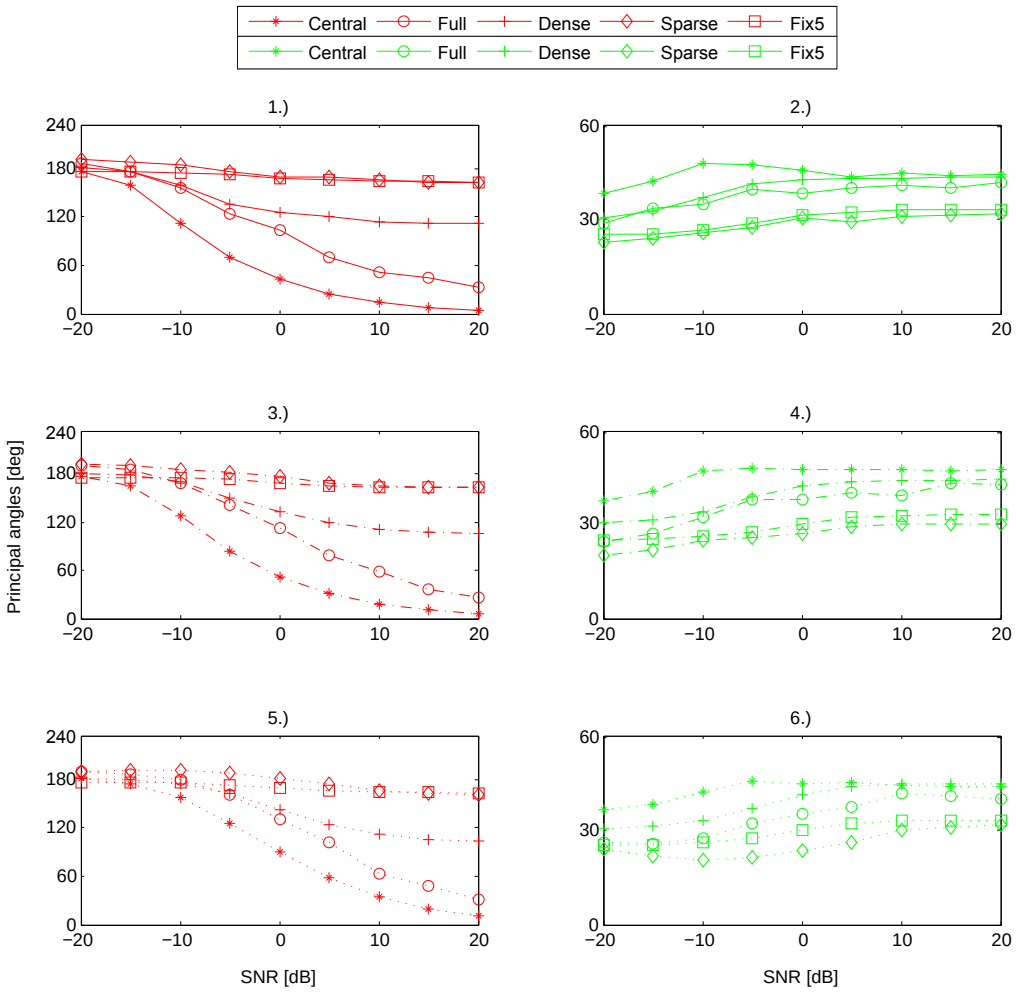

Figure B.18: Evaluation of the subspace angle difference between the estimated subspace $\mathbf{W}$ and the true one provided by the steering matrix $\boldsymbol{\Upsilon}$ for the step signals. This cases regards the step signal case. The Algorithm 3 is analyzed for the topologies proposed in Figure 2.3. The results displayed at the right side correspond to the first principal angle, and those at the left side, relate to the second principal angle. Again, the performance for several forgetting factors are depicted: $\beta = 0.9$ in 1.) and 2.), $\beta = 0.8$ in 3.) and 4.). At last, $\beta = 0.6$ is shown in 5.) and 6.).

Appendix C

Definitions related to Chapter 5

Here we start with both local algorithmic descriptions and arrive at two different global frameworks consisting of block diagonal matrices.

C.1 Calculation for the global description of Algorithm 4.

The target here is to find a global description for the set of equations (5.33)-(5.36). We start by defining the global expression for the local signal update $\underline{\mathbf{y}}_i(k)$ in (5.33) as

$$\mathbf{Y}(k) = \begin{pmatrix} \underline{\mathbf{y}}_i(k) & & & \\ & \underline{\mathbf{y}}_{2,k} & & \\ & & \ddots & \\ & & & \underline{\mathbf{y}}_N(k) \end{pmatrix} = \begin{pmatrix} \mathbf{V}_{1,k-1}^H \underline{\mathbf{x}}(k) & & & \\ & \mathbf{V}_{2,k-1}^H \underline{\mathbf{x}}(k) & & \\ & & \ddots & \\ & & & \mathbf{V}_{N,k-1}^H \underline{\mathbf{x}}(k) \end{pmatrix}.$$

By factorization, it is possible to separate the terms above and build the global $\mathbf{V}^H(k-1)$ block diagonal matrix in terms of its local elements. Likewise, a vector containing N times the incoming observation $\underline{\mathbf{x}}(k)$ is provided through a Kronecker product. That is to say,

$$\mathbf{Y}(k) = \begin{pmatrix} \mathbf{V}_{1,k-1}^H & & & \\ & \mathbf{V}_{2,k-1}^H & & \\ & & \ddots & \\ & & & \mathbf{V}_{N,k-1}^H \end{pmatrix} \begin{pmatrix} \underline{\mathbf{x}}(k) & & & \\ & \underline{\mathbf{x}}(k) & & \\ & & \ddots & \\ & & & \underline{\mathbf{x}}(k) \end{pmatrix} \tag{C.1}$$

$$= \mathbf{V}^H(k-1)(\underline{\mathbf{I}}_N \otimes \underline{\mathbf{x}}(k)). \tag{C.2}$$

A single average consensus step of (3.7) is given by

$$\begin{pmatrix} g_{11}\mathbf{V}_{1,k-1}^H\underline{\mathbf{x}}(k) + \sum\limits_{\substack{j\in\mathcal{N}_1,\\ j\neq 1}} g_{1j}\mathbf{V}_{j,k-1}^H\underline{\mathbf{x}}(k) & & \\ & \ddots & \\ & & g_{NN}\mathbf{V}_{N,k-1}^H\underline{\mathbf{x}}(k) + \sum\limits_{\substack{j\in\mathcal{N}_N,\\ j\neq N}} g_{Nj}\mathbf{V}_{j,k-1}^H\underline{\mathbf{x}}(k) \end{pmatrix},$$

which is factorable as

$$= \begin{pmatrix} g_{11}\mathbf{V}_{1,k-1}^H + \sum\limits_{\substack{j\in\mathcal{N}_1,\\ j\neq 1}} g_{1j}\mathbf{V}_{j,k-1}^H & & \\ & \ddots & \\ & & g_{NN}\mathbf{V}_{N,k-1}^H + \sum\limits_{\substack{j\in\mathcal{N}_N,\\ j\neq N}} g_{Nj}\mathbf{V}_{j,k-1}^H \end{pmatrix}$$

$$\times \begin{pmatrix} \underline{\mathbf{x}}(k) & & & \\ & \underline{\mathbf{x}}(k) & & \\ & & \ddots & \\ & & & \underline{\mathbf{x}}(k) \end{pmatrix} = (\mathbf{G}\otimes\mathbf{I}_r)\,\mathbf{V}^H(k-1)\,(\underline{\mathbf{x}}(k)\otimes\underline{\mathbf{1}}_N) = (\mathbf{G}\otimes\mathbf{I}_r)\,\mathbf{Y}(k)\mathbf{1}_N.$$

Since a consensus in the network is guaranteed if and only if $t_{\max}\to\infty$, the global expression for the averaging step in (3.7) is defined as

$$\widetilde{\mathbf{Y}}(k) = \begin{pmatrix} \underline{\mathbf{y}}_i(k)(t_{\max}) \\ \underline{\mathbf{y}}_{2,k}(t_{\max}) \\ \vdots \\ \underline{\mathbf{y}}_N(k)(t_{\max}) \end{pmatrix} = \left(\mathbf{G}^{t_{\max}}\otimes\mathbf{I}_r\right)\mathbf{V}^H(k-1)\,(\underline{\mathbf{x}}(k)\otimes\underline{\mathbf{1}}_N) = \left(\mathbf{G}^{t_{\max}}\otimes\mathbf{I}_r\right)\mathbf{Y}(k)\mathbf{1}_N.$$

The following step is to find a global description for the a posteriori error. Let's rewrite (5.34) in terms of the local signal subspace matrix $\mathbf{W}_i(k-1)$ by recalling $\mathbf{V}=\mathbf{S}_i\mathbf{W}_i(k-1)$ and $\mathbf{S}_i^T\mathbf{S}_i=\mathbf{I}_{\mathcal{N}_i}$. Namely,

$$\begin{aligned}\underline{\mathbf{e}}_i(k) &= \mathbf{S}_i^T\left(\underline{\mathbf{x}}(k)-\mathbf{V}_i(k-1)\mathbf{V}_i^H(k-1)\underline{\mathbf{x}}(k)\right) = \mathbf{S}_i^T\underline{\mathbf{x}}(k) - \mathbf{S}_i^T\mathbf{S}_i\mathbf{W}_i(k-1)\mathbf{W}_i^H(k-1)\mathbf{S}_i^T\underline{\mathbf{x}}(k)\\ &= \mathbf{S}_i^T\underline{\mathbf{x}}(k) - \mathbf{W}_i(k-1)\mathbf{W}_i^H(k-1)\mathbf{S}_i^T\underline{\mathbf{x}}(k) = \underline{\mathbf{x}}_i(k) - \mathbf{W}_i(k-1)\mathbf{W}_i^H(k-1)\underline{\mathbf{x}}_i(k)\\ &= \underline{\mathbf{x}}_i(k) - \mathbf{W}_i(k-1)\underline{\mathbf{y}}_i(k).\end{aligned}$$

However, as the error is actually influenced by the averaged value $\widetilde{\mathbf{y}}_i(k)$, it is important to include (3.7) in the calculations. We therefore reformulate the above equation as $\underline{\mathbf{e}}_i(k)=\underline{\mathbf{x}}_i(k)-\mathbf{W}_i(k-1)\underline{\widetilde{\mathbf{y}}}_i(k)$ and let the a posteriori error be determined as

$$\underline{\mathbf{e}}_i(k) = \underline{\mathbf{x}}_i(k) - \mathbf{W}_i(k-1)\left(g_{ii}\mathbf{W}_i^H(k-1)\underline{\mathbf{x}}_i(k) + \sum_{\substack{j\in\mathcal{N}_i,\\ j\neq i}} g_{ij}\mathbf{W}_{j,k-1}^H\underline{\mathbf{x}}_{j,k}\right).$$

Replacing $\underline{\mathbf{x}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k)$ and $\mathbf{W}_i(k-1) = \mathbf{S}_i^T \mathbf{V}_i(k-1)$ in the previous equation leads to

$$\underline{\mathbf{e}}_i(k) = \mathbf{S}_i^T \underline{\mathbf{x}}(k) - \mathbf{S}_i^T \mathbf{V}_i(k-1) \left(g_{ii} \mathbf{V}_i^H(k-1) \mathbf{S}_i \mathbf{S}_i^T \underline{\mathbf{x}}(k) + \sum_{\substack{j \in \mathcal{N}_i, \\ j \neq i}} g_{ij} \mathbf{V}_{j,k-1}^H \mathbf{S}_j \mathbf{S}_j^T \underline{\mathbf{x}}(k) \right),$$

and the back substitution of $\mathbf{V}^H(k-1) = \mathbf{W}_i^H(k-1)\mathbf{S}_i^T$ allows for further simplification

$$\begin{aligned} \underline{\mathbf{e}}_i(k) &= \mathbf{S}_i^T \underline{\mathbf{x}}(k) - \mathbf{S}_i^T \mathbf{V}_i(k-1) \left(g_{ii} \mathbf{W}_i^H(k-1) \mathbf{S}_i^T \mathbf{S}_i \mathbf{S}_i^T \underline{\mathbf{x}}(k) + \sum_{\substack{j \in \mathcal{N}_i, \\ j \neq i}} g_{ij} \mathbf{W}_{j,k-1}^H \mathbf{S}_j^T \mathbf{S}_j \mathbf{S}_j^T \underline{\mathbf{x}}(k) \right), \\ &= \mathbf{S}_i^T \left(\mathbf{I}_N - \mathbf{V}_i(k-1) \left(g_{ii} \mathbf{V}_i^H(k-1) + \sum_{\substack{j \in \mathcal{N}_i, \\ j \neq i}} g_{ij} \mathbf{V}_{j,k-1}^H \right) \right) \underline{\mathbf{x}}(k). \end{aligned}$$

A global selection matrix is likewise addressed in a block matrix fashion as $\mathbf{S} \in \{0,1\}^{N^2 \times \sum_{i=1}^N |\mathcal{N}_i|}$. Thus, the global error update equation is characterized by

$$\widetilde{\mathbf{E}}(k) = \begin{pmatrix} \widetilde{\underline{\mathbf{e}}}_i(k) & & & \\ & \widetilde{\underline{\mathbf{e}}}_{2,k} & & \\ & & \ddots & \\ & & & \widetilde{\underline{\mathbf{e}}}_N(k) \end{pmatrix} = \mathbf{S}^T \left(I_{N^2} - \mathbf{V}(k-1)(\mathbf{G} \otimes \mathbf{I}_r)\mathbf{V}^H(k-1) \right) \left(\mathbf{I}_N \otimes \underline{\mathbf{x}}(k) \right).$$

The above structure is similarly obtained for (5.35) and (5.36) when $\widehat{\mathbf{R}}^{yy}(k), \widetilde{\mathbf{Y}}(k)$ and $\mathbf{E}(k)$ are used. That is to say,

$$\begin{aligned} \widehat{\mathbf{R}}^{yy}(k) &= \widehat{\mathbf{R}}^{yy}(k-1) + \gamma(k) \left(\widetilde{\mathbf{Y}}(k) \widetilde{\mathbf{Y}}^H(k) - \widehat{\mathbf{R}}^{yy}(k-1) \right). \\ \mathbf{V}(k) &= \mathbf{V}(k-1) + \gamma(k) \mathbf{E}(k) \mathrm{diag} \left(\widetilde{\mathbf{Y}}^H(k) \right) \left(\widehat{\mathbf{R}}^{yy}(k) \right)^{-1}. \end{aligned}$$

Here, we let $\mathbf{E}(k) = \mathbf{S}\widetilde{\mathbf{E}}(k)$, $\gamma(k)$ is a vector of size $(N \times 1)$ containing the local $\gamma_i(k)$ and allow the global $\widetilde{\mathbf{Y}}^H(k)$ to become a block diagonal matrix $(rN \times N)$. The reason for this factorization is to preserve the correct dimension of $\mathbf{V}(k)$. Finally, we realize that our distributed signal subspace update equation (5.42) has the same structure from the centralized solution (5.4). Therefore, it is straightforward to use the guidelines introduced in [80] for analyzing the convergence properties of Algorithm 4.

C.2 Calculation for the global description of Algorithm 5.

As mentioned in 5.1.4, the signal update vector given in (5.33) does not change for the second algorithmic variant. Therefore, the global expression for $\mathbf{Y}(k)$ is exactly the same as the one in (C.2). When factorizing (5.37), the data estimation error $\underline{\mathbf{e}}_i(k)$ (independent of $\underline{\widetilde{\mathbf{y}}}_i(k)$) is determined by

$$\underline{\mathbf{e}}_i(k) = \mathbf{S}_i^T \left(\underline{\mathbf{x}}(k) - \mathbf{V}_i(k-1)\mathbf{V}_i^H(k-1)\underline{\mathbf{x}}(k)\right).$$

If each $\underline{\mathbf{e}}_i(k)$ is located at the diagonal of a block matrix $\mathbf{E}(k)$, its global description yields:

$$\mathbf{E}(k) = \begin{pmatrix} \underline{\mathbf{e}}_i(k) & & & \\ & \underline{\mathbf{e}}_{2,k} & & \\ & & \ddots & \\ & & & \underline{\mathbf{e}}_N(k) \end{pmatrix} = \mathbf{S}^T \left(I_{N^2} - \mathbf{V}(k-1)\mathbf{V}^H(k-1)\right)\left(\mathbf{I}_N \otimes \underline{\mathbf{x}}(k)\right).$$

Every node i locally updates the autocorrelation matrix using the equation

$$\widehat{\mathbf{R}}_i^{yy}(k) = \widehat{\mathbf{R}}_i^{yy}(k-1) + \gamma(k)\left[\underline{\mathbf{y}}_i(k)\underline{\mathbf{y}}_i^H(k) - \widehat{\mathbf{R}}_i^{yy}(k-1)\right],$$

and its block diagonal representation is given as

$$\widehat{\mathbf{R}}^{yy}(k) = \begin{pmatrix} \widehat{\mathbf{R}}_i^{yy}(k) & & & \\ & \widehat{\mathbf{R}}_{2,k}^{yy} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{R}}_i^{yy}(k) \end{pmatrix} = \widehat{\mathbf{R}}^{yy}(k-1) + \gamma(k)\left[\mathbf{Y}(k)\mathbf{Y}^H(k) - \widehat{\mathbf{R}}^{yy}(k-1)\right].$$

The averaging steps occur immediately upon the variable $\mathbf{R}_i(k)$, where a general expression for (3.10) is obtained through

$$\begin{aligned} \widetilde{\mathbf{R}}^{yy}(k) &= \begin{pmatrix} \widetilde{\mathbf{R}}_i^{yy}(k) \\ \widetilde{\mathbf{R}}_{2,k}^{yy} \\ \vdots \\ \widetilde{\mathbf{R}}_i^{yy}(k) \end{pmatrix} = (\mathbf{G} \otimes \mathbf{I}_r) \begin{pmatrix} \widehat{\mathbf{R}}_i^{yy}(k) \\ \widehat{\mathbf{R}}_{2,k}^{yy} \\ \vdots \\ \widehat{\mathbf{R}}_i^{yy}(k) \end{pmatrix} = (\mathbf{G} \otimes \mathbf{I}_r)\, \widehat{\mathbf{R}}^{yy}(k) \begin{pmatrix} \mathbf{I}_r \\ \mathbf{I}_r \\ \vdots \\ \mathbf{I}_r \end{pmatrix} \\ &= \left(\mathbf{G}^{t_{\max}} \otimes \mathbf{I}_r\right) \widehat{\mathbf{R}}^{yy}(k) \left(\mathbf{1}_N \otimes \mathbf{I}_r\right) \end{aligned}$$

Notice that instead of a diagonal matrix representation, $\widetilde{\mathbf{R}}^{yy}(k)$ turns out to be a vector whose elements are matrix blocks. This matrix has to be diagonalized in the next step to preserve the structure of the general form of $\mathbf{V}(k)$. Again, letting

$\mathbf{E}(k) = \mathbf{S}\mathbf{E}(k)$ yields,

$$\mathbf{V}(k) = \mathbf{V}(k-1) + \gamma(k)\mathbf{E}(k)\mathbf{Y}^H(k)\left(diag\,\widetilde{\mathbf{R}}^{yy}(k)\right)^{-1}.$$

C.3 Variables dimensions

The matrix size from a local to a global representation is given as:

$$\begin{aligned}
\underline{\mathbf{y}}_i(k) \in \mathbb{C}^{r\times 1} &\longrightarrow \mathbf{Y}(k) \in \mathbb{C}^{rN\times N} \\
\widehat{\mathbf{R}}_i^{yy}(k) \in \mathbb{C}^{r\times r} &\longrightarrow \widehat{\mathbf{R}}^{yy}(k) \in \mathbb{C}^{rN\times rN} \\
\underline{\mathbf{e}}_i(k) \in \mathbb{C}^{|\mathcal{N}_i|\times 1} &\longrightarrow \mathbf{E}(k) \in \mathbb{C}^{\sum_{i=1}^{N}|\mathcal{N}_i|\times N} \\
\mathbf{S}_i \in \{0,1\}^{N\times|\mathcal{N}_i|} &\longrightarrow \mathbf{S} \in \{0,1\}^{N^2\times\sum_{i=1}^{N}|\mathcal{N}_i|}.
\end{aligned}$$

$$\begin{aligned}
\widetilde{\underline{\mathbf{y}}}_i(k) \in \mathbb{C}^{r\times 1} &\longrightarrow \widetilde{\mathbf{Y}}(k) \in \mathbb{C}^{rN\times N} \\
\widetilde{\mathbf{R}}_i^{yy}(k) \in \mathbb{C}^{r\times r} &\longrightarrow \widetilde{\mathbf{R}}^{yy}(k) \in \mathbb{C}^{rN\times rN} \\
\mathbf{V}_i(k) \in \mathbb{C}^{N\times r} &\longrightarrow \mathbf{V}(k) \in \mathbb{C}^{N^2\times rN}
\end{aligned}$$

Bibliography

[1] B. O'Flyrm, R. Martinez, J. Cleary, C. Slater, F. Regan, D. Diamond, and H. Murphy. Smartcoast: A wireless sensor network for water quality monitoring. *32nd IEEE Conference on Local Computer Networks, 2007. LCN 2007.*, October 2007.

[2] I. F. Akyildiz, D. Pompoli, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, pages 257–279, 2005.

[3] G.H. Golub and C.F. Van Loan. Some modified matrix eigenvalue problems. *SIAM review*, 15(318-334), 1973.

[4] J. R. Bunch and C. P. Nielsen. Updating the singular value decomposition. *Numer. Math.*, (31):111–129, 1978.

[5] J. R. Bunch, C. P. Nielsen, and D.C. Sorensen. Rank-one modification of the symmetric eigenvalue problem. *Numer. Math.*, (31):31–48, 1978.

[6] J. F. Yang and M. Kaveh. Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Trans. ASSP*, 36:241–251, 1988.

[7] D. J. Rabideau. Fast, ranks adaptive subspace tracking and applications. *IEEE Trans. Sig. Proc.*, 44(9):2229–2244, 1996.

[8] B. Yang. Projection approximation subspace tracking. *IEEE Trans. Sig. Proc.*, 43(1):95–107, 1995.

[9] D.W. Tufts, E.C. Real, and J.W. Cooley. Fast approximate subspace tracking. *ICAASP*, pages 547–550, 1997.

[10] R. Olfati-saber and R. Murray. Consensus protocols for networks of dynamic agents. *American Control Conf.*, 2:951–956, June 2003.

[11] K. Khedo, R. Perseedoss, and A. Mungur. A wireless sensor network air pollution monitoring system. *International Journal of Wireless and Mobile Networks (IJWMN)*, 2(2):31–45, May 2010.

[12] H. Labiod. *Wireless Ad-Hoc and Sensor Networks.* Wiley and Sons, 2008.

[13] I. F. Akyildiz and M. C. Vuran. *Wireless Sensor Networks.* Wiley and Sons, 2010.

[14] Inverse-square law.

[15] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science*, 2003.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. on Inf. Theor.*, 52(6):2508–2530, June 2006.

[17] Sosus: Sound surveillance system.

[18] C.Y. Chong and S.P. Kumar. Sensor networks: evolution, opportunities and challenges. *Proceedings of the IEEE*, 91(8):1247–1256, August 2003.

[19] Dept. Comput. Sci., Carnegie Mellon University. *Proceedings of the Distributed Sensor Nets Workshop*, 1978.

[20] S. H. Lee, S. Lee, H. Song, and H. S. Lee. Wireless sensor network design for tactical military applications : Remote large-scale environments. *IEEE Military Communications Conference, MILCOM*, pages 1–7, October 2009.

[21] J. George and L. M. Kaplan. Shooter localization using a wireless sensor network of soldier-worn gunfire. *Journal of advances in information fusion*, 8(1):15–32, June 2013.

[22] R. L. Schafer and R. J. Hammell II. Spatial interpolation for a wireless sensor network of chemical point detectors: spatial interpolation for a wireless sensor: Preliminary results. *Conference on Information Systems Applied Research Proceedings, CONISAR*, pages 1–8, November 2009.

[23] B. Song, C. Ding, A. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury. Distributed camera networks. *IEEE signal processing magazine*, pages 20–31, May 2011.

[24] B. A. Khan, M. Sharif, M. Raza, T. Umer, K. Hussain, and A.U. Khan. An approach for surveillance using wireless sensor networks (wsn). *Journal of Information and Communication Technology*, 1:35–42, 2007.

[25] R. Ohbayashi, Y. Nakajima, H. Nishikado, and S. Takayama. Monitoring system for landslide disaster by wireless sensing node network. *SICE Annual Conference*, pages 1704–1010, August 2008.

[26] C. Alippi, G. Anastasi, C. Galperti, F. Mancini, and M. Roveri. Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, 2007.

[27] H. Sone. Ai based agriculture support system with precisely deployed versatile sensors and sensor network. *International Conference on Advance Intelligence and Awareness Internet (AIAI)*, October 2010.

[28] C. Wan-Young, L. Young-Dong, and J. Sang-Joong. A wireless sensor network compatible wearable u-healthcare monitoring system using integrated ecg, accelerometer and spo.2. *International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1529–1532, 2008.

[29] J. Sang-Joong, L. Young-Dong, S. Yong-Su, and C. Wan-Young. Design of a low-power consumption wearable reflectance pulse oximetry for ubiquitous healthcare system. *International Conference on Control, Automation and Systems*, pages 526–529, October 2008.

[30] J. Foutz, A. Spanias, and M. K. Banavar. *Narrowband direction of arrival estimation for antenna arrays*. Morgan and Claypool Publishers, 2008.

[31] Lawrence J. Ziomek. *Fundamentals of acoustic field theory and space-times signal processing*. CRC Press, 1995.

[32] D. Johnson and D. Dudgeon. *Array signal processing: Concepts and thechniques*. Prentice-Hall, 1993.

[33] H. L. Van Trees. *Optimum Array Processing. Part IV of Detection, Estimation and Modulation Theory*. Number 0-471-22110-4. John Wiley, 2002.

[34] J. Akhtman and L. Hanzo. Decision directed channel estimation employing projection approximation subspace tracking. *IEEE Proceedings*, pages 3056–3060, 2007.

[35] T. Ahmed, M. Coates, and A. Lakhina. Multivariate online anomaly detection using kernel recursive least squares. *Proc. 26th IEEE International Conference on Computer Communications*, pages 625–633, 2007.

[36] L. Balzano, B. Recht, and R. Nowak. High-dimensional matched subspace detection when data are missing. *Proc. ISIT*, June 2010.

[37] R. Kumaresan and D. W. Tufts. Estimating the angles of arrival of multiple plane waves. *IEEE Trans. on Aerospace and Electronic Systems*, pages 134–139, 1983.

[38] A.H. Sayed. *Adaptive filters*. John Wiley, 2008.

[39] R. Roy and T. Kailath. Esprit- estimation of signal parameters via rotational invatiance techniques. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 37(7):984–995, July 1989.

[40] R. O. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Trans. Ant. Prop.*, AP-34(3):276 – 280, March 1986.

[41] C.F. Mecklenbräuker, P. Gerstoft, A. Panahi, and M. Viberg. Sequential bayesian sparse source reconstruction using array data. *IEEE Trans. Sig. Proc.*, 2013.

[42] B. Schölkopf, J. Platt, and T. Hofmann. Blind source separation for over determined delayed mixtures. In *Advances in Neural Information Processing Systems*, pages 1049–1056. MIT Press, 2007.

[43] L. Fu and R. J. Vaccaro. Sensitivity analysis of doa estimation algorithms to sensor errors. *IEEE Trans. on Aerospace and Electronic Systems*, 28:708–717, August 1992.

[44] P. Vallet, W. Hachem, P. Loubaton, and X. Mestre. An improved music algorithm based on low rank perturbation of large random matrices. *2011 IEEE Statistical Signal Processing Workshop*, pages 689–692, June 2011.

[45] J. A. Fessler and A. O. Hero. Space-alternating generalized expectation-maximization algorithm. *IEEE Trans. Sig. Proc.*, October 1994.

[46] B. Yang. Subspace tracking based on the projection approach and the recursive least squares method. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'93)*, volume 4, pages 145 –148, April 1993.

[47] B. Yang. Convergence analysis of the subspace algorithms PAST and Pastd. *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'96)*, pages 3192 – 3193, 1996.

[48] B. Yang. Asymptotic convergence analysis of the projection approximation subspace tracking algorithms. *Signal Processing*, 50:123–136, 1996.

[49] S. Haykin. *Adaptive filter theory.* Prentice-Hall, Englewood-Cliffs (NJ), USA, 1991.

[50] R. Olfati-Saber. Distributed kalman filtering for sensor networks. *IEEE Conference on Decision and Control*, pages 5492–5498, December 2007.

[51] L. Li, A. Scaglione, and J. H. Manton. Distributed principal subspace estimation in wireless sensor networks. *IEEE Journal on selected topics in signal processing*, 5(4):725–738, August 2011.

[52] Demetri P. Spanos. *Distributed gradient systems and dynamic coordination.* PhD thesis, California institute of technology, December 2005.

[53] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

[54] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization. *Advances in neural information processing systems*, pages 2080–2088, 2009.

[55] C. Hage and M. Kleinsteuber. Robust pca and subspace tracking from incomplete observations using l_0-surrogates. *Computational statistics*, July 2013.

[56] A. Srivastava. A bayesian approach to geometric subspace estimation. *IEEE Trans. Sig. Proc.*, 48(5):1390–1400, May 2000.

[57] S. Kar, S. Cui, H.V. Poor, and J.M.F. Moura. Convergence results in distributed kalman filtering. *IAcoustics, Speech and Signal Processing*, pages 2500–2503, May 2011.

[58] J. Kleinberg and E. Tardos. *Algorithm Design.* Pearson Int., 2006.

[59] R. Olfati-saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Contr.*, 49(9):1520–1533, Sept. 2004.

[60] C. Moallemi and B. Van Roy. Consensus propagation. *IEEE Trans. Inf. Theor.*, 52(11):4753–4766, Nov. 2006.

[61] C. Reyes, T. Hilaire, and C. F. Mecklenbräuker. Distributed projection approximation subspace tracking based on consensus propagation. In *The 3rd International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Aruba, December 2009.

[62] C. Reyes, R. Dallinger, and M. Rupp. Convergence analysis of distributed past based on consensus propagation. *Signals, Systems and Computers (ASILOMAR)*, pages 271 – 275, November 2012.

[63] C. Reyes, T. Hilaire, S. Paul, and C. F. Mecklenbräuker. Evaluation of the root mean square error performance of the PAST-consensus algorithm. In *International ITG Workshop on Smart Antennas (WSA 2010)*, Germany, February 2010.

[64] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *System and control letters*, 2004.

[65] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. *Proc. 4^{th} IPSN*, pages 63–67, April 2005.

[66] D.P. Bertsekas and J.N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice-Hall, 1989.

[67] M. Mehyar, D. Spanos, J. Pongsajapan, S.H. Low, and R.M. Murray. Asynchronous distributed averaging on communication networks. *IEEE Trans. on Networking*, 15(3):512–520, June 2007.

[68] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Letters*, 75(6):1226–1229, 1995.

[69] A. Jadbabaie, J. Lie, and A.S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. Autom. Contr.*, 48(6):988–1001, June 2003.

[70] L. Xiao, S. Boyd, and S. Lall. Distributed average consensus with time varying metropolis weights. June 2006.

[71] C. Jordan. *Essai sur la géométri à n dimensions*, volume 3. 1875.

[72] I. S. Dhillon, D. S. Modha, and W. S. Spangler. Class visualization of high-dimensional data with applications. *Computational Statistics and Data Analysis*, 41:59–90, 2002.

[73] M. Kirby. Classification of data bundles via parameter spaces. Technical report, Colorado State University, Fort Collins, CO 80523, December 2011.

[74] Y. Igarashi and K. Fukui. 3d object recognition based on canonical angles between shape subspaces. *Lecture Notes in Computer Sciences*, 6495:580–591, 2011.

[75] R. Shigenaka, B. Raytchev, and T. Tamaki abd K. Kaneda. Face sequence recognition using grassmann distances and grassmann kernels. *WCCI 2012 IEEE World Congress on Computational Intelligence*, pages 2630–2636, 2012.

[76] G. H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 1983.

[77] P. Schonemann. A generalized solution of the orthogonal procrustes problem. 31:1–10, March 1966.

[78] C. Shonkviller. Principal angles in terms of inner products. Technical report, University of Georgia.

[79] A.V. Knyazev and M.E. Argentati. Principal angles between subspaces in an a-based scalar product: Algorithms and perturbation estimates. *Society for Industrial and Applied Mathematics*, 23(6):2008–2040, 2002.

[80] M. Rupp and C. Reyes. Robust versions of the PAST algorithm. *Proc. of EUSIPCO*, Bucharest, Rumania. Sep. 2012.

[81] K. Abed-Meraim, A. Chkeif, and Y. Hua. Fast orthonormal past algorithm. *IEEE Signal processing letters*, 7(3), March 2000.

[82] S. C. Douglas. Simple adaptive algorithms for Cholesky, LDL^T , QR, and eigenvalue decompositions of autocorrelation matrices for sensor array data. In *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, November 2001.

[83] C. Mehlführer and M. Rupp. A robust MMSE equalizer for MIMO enhanced HSDPA. In *Conference Record of the Fourtieth Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, October 2006.

[84] P. Comon and G. H. Golub H. Golub. Tracking a few extreme singular values and vectors in signal processing vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, August 1990.

[85] Z.Bai et al. *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*. SIAM, Philadelphia, 2000.

[86] D.A. Linebarger, R.D. DeGroat, E.M. Dowling, P. Stoica, and G. Fudge. Incorporating a priori information into music - algorithms and analysis. *Signal Processing*, 46(1):85–104, 1995.