

Blind Beamforming Techniques
for Global Tracking Systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op
vrijdag 4 december 2015 om 10:00 uur
door

Mu ZHOU
Master of Science in Electrical Engineering
geboren te Yibin, China.

To my family

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | AIS | 2 |
| 1.3 | Satellite AIS and Nonstandard AIS Stations | 2 |
| 1.4 | Review of Literature | 5 |
| 1.5 | Goal of the Thesis | 6 |
| 1.6 | Thesis Outline and Contributions | 8 |
| 2 | Signed URV Decomposition and Principal Subspace Tracking | 13 |
| 2.1 | Introduction | 13 |
| 2.2 | J-unitary Matrices | 17 |
| 2.3 | Hyperbolic URV Decomposition | 19 |
| 2.4 | SURV: An Updating Algorithm for the Hyperbolic URV | 21 |
| 2.4.1 | Overview | 21 |
| 2.4.2 | Algorithm | 22 |
| 2.4.3 | Initialization | 26 |
| 2.4.4 | Computational Complexity | 26 |
| 2.5 | Updating the SSE-2 | 27 |
| 2.5.1 | SSE-2 Definition and Properties | 27 |
| 2.5.2 | The SURV Algorithm Provides an SSE-2 Decomposition | 28 |
| 2.5.3 | Numerical Stability | 29 |
| 2.6 | Simulation Results | 30 |
| 2.7 | Conclusions | 35 |
| 2.A | The Principal Subspace is an SSE-2 | 37 |
| 2.B | Proof for Theorem 1 in Section 2.5 | 37 |
| 2.C | Proof of the Updating Algorithm for SURV | 42 |
| 2.C.1 | $j_{\mathbf{c}} = +1$ and $j'_{\mathbf{c}} = +1$ | 42 |
| 2.C.2 | $j_{\mathbf{c}} = +1$ and $j'_{\mathbf{c}} = -1$ | 44 |
| 2.C.3 | $j_{\mathbf{c}} = -1$ and $j'_{\mathbf{c}} = -1$ | 45 |

| | | |
|----------|---|-----------|
| 2.C.4 | $j_c = -1$ and $j'_c = +1$ | 45 |
| 3 | Blind Beamforming Techniques for Separating Partially Overlapping Data Packets | 47 |
| 3.1 | Introduction | 47 |
| 3.2 | Data Model | 48 |
| 3.2.1 | Signals | 48 |
| 3.2.2 | Stationary Data Model | 49 |
| 3.2.3 | Separation Scenarios | 50 |
| 3.2.4 | Objective | 52 |
| 3.3 | Tools from Linear Algebra | 52 |
| 3.3.1 | Singular Value Decomposition (SVD) | 52 |
| 3.3.2 | Generalized Singular Value Decomposition (GSVD) | 53 |
| 3.3.3 | Generalized Eigenvalue Decomposition | 54 |
| 3.3.4 | Oblique Projections | 54 |
| 3.3.5 | Subspace Intersection | 54 |
| 3.4 | Separation Using a Single Data Block | 55 |
| 3.5 | Separation Using Two Data Blocks | 56 |
| 3.5.1 | Scenario 1 | 56 |
| 3.5.2 | Scenario 2 | 59 |
| 3.6 | Signed URV Decomposition | 61 |
| 3.6.1 | Definition of the SSE | 61 |
| 3.6.2 | Properties | 62 |
| 3.7 | Source Separation Using SURV | 64 |
| 3.7.1 | Application to Scenario 1 | 64 |
| 3.7.2 | Application to Scenario 2 | 68 |
| 3.8 | Generalization of Algorithms for Nonstationary Data Model | 71 |
| 3.9 | Simulation Results | 73 |
| 3.9.1 | Data Model with Stationary Sub-Blocks | 73 |
| 3.9.2 | Data Model with Partially Overlapping Data Packets | 74 |
| 3.10 | Experiments | 77 |
| 3.11 | Conclusions | 78 |
| 3.A | Comparison of SURV and SVD Principal Subspace Estimates | 78 |
| 3.B | GSVD is a Special Case of SSE-2 | 79 |
| 3.C | Bounds on Eigenvalues of Random Matrices | 80 |
| 3.D | Noise Power Shifting Using γ | 81 |
| 3.E | Convergence of Correlations | 82 |

| | | |
|----------|---|------------|
| 4 | Robust Multi-User Receiver for Satellite AIS Using Blind Beamforming Techniques and Tracking | 85 |
| 4.1 | Introduction | 85 |
| 4.2 | Data Model | 88 |
| 4.3 | Multi-User Receiver Structure | 90 |
| 4.4 | Blind Beamforming Technique | 92 |
| 4.5 | Simulation Results | 95 |
| 4.6 | Conclusions | 97 |
| 5 | SystemC-AMS Model of a Dynamic Large-Scale Network of Satellite AIS | 103 |
| 5.1 | Introduction | 103 |
| 5.2 | Mechanism behind the Model | 104 |
| 5.2.1 | AIS Physical Layer | 105 |
| 5.2.2 | AIS Link Layer | 106 |
| 5.2.3 | Physical Channel Effects | 106 |
| 5.2.4 | Ship Distribution and Satellite Orbit Design | 116 |
| 5.3 | C++ Model | 116 |
| 5.3.1 | Assistant C++ Classes | 117 |
| 5.3.2 | Virtual Transmitter | 117 |
| 5.3.3 | Channel | 118 |
| 5.3.4 | Satellite Receiver | 119 |
| 5.3.5 | Channel Effects and SOTDMA Protocol | 120 |
| 5.3.6 | Ship Distribution and Satellite Orbit | 121 |
| 5.3.7 | System Controller | 122 |
| 5.3.8 | Additional Functions | 124 |
| 5.4 | Simulation Results | 125 |
| 5.4.1 | Verification of the Model | 127 |
| 5.4.2 | Validation of the Model | 130 |
| 5.5 | Conclusions | 132 |
| 5.A | Lists of Codes | 133 |
| 6 | A Demonstration System of Advanced Receivers for Satellite AIS | 147 |
| 6.1 | Motivation and Overview | 147 |
| 6.2 | Hardware Receiver | 148 |
| 6.3 | Analysis of AIS Data | 152 |
| 6.4 | Presentation of AIS Data | 153 |
| 6.5 | Some Results | 153 |
| 6.6 | Conclusions | 153 |

| | |
|--------------------------------------|------------|
| 7 Conclusions and Future Work | 157 |
| 7.1 Conclusions | 157 |
| 7.2 Future Work | 159 |
| Bibliography | 161 |
| Summary | 171 |
| Samenvatting | 173 |
| Acknowledgements | 175 |
| Curriculum Vitae | 177 |

Chapter 1

Introduction

1.1 Motivation

After 120 years of development since the invention of the first radio transmission device, modern wireless communication systems are nowadays part of daily life. The whole world benefits from the development of wireless communication systems across generations. Behind the shining and astonishing achievements of the modern wireless communication systems, the exhaustion of existing frequency spectrum resources has been a concern. In higher frequency bands, the most advanced techniques are in development for the fifth generation cellular mobile communication system (5G) to meet the rapid growth of the demand for high performance in the market. The 5G system sits in the millimeter-wave band and consumes a wide bandwidth to offer a high data transmission speed and large system capacity. In the 5G system and other cellular wireless communication systems, frequency/time/code division multiple accesses (FDMA/TDMA/CDMA) are successful techniques to reuse and to save the frequency spectrum resources. In lower frequency bands, existing communication systems face similar unprecedented demands to accommodate more users in new applications. These growing demands exceed the designed system capacity and thus call for innovative solutions while keeping compatibility to the current setup to reduce the cost of users. For example, in the automatic identification system (AIS), satellite receivers are being used for expanding the service coverage of ship tracking to the global range, and similarly in the automatic dependent surveillance-broadcast (ADS-B) system for aircraft tracking. Although not many people have direct contact with these systems, these systems indeed play an important role in modern traffic safety. Looking at the history of traffic tragedies, we can find two typical cases in recent years:

- “On 5-Dec.-2012, the ship Baltic Ace collided with another ship Corvus J in a shipping lane around 30 miles from the port of Rotterdam...”

- “On 8-Mar.-2014, a Malaysia airlines flight MH370 went missing and silently ended in Southern India Ocean thousands miles away...”

In these tragedies, the tracking systems were somehow not functional (no matter intentionally or not), leading to loss of property and human lives. How useful the tracking systems are! It is of a high potential value building and developing a reliable network to monitor the real-time traffic in a scope from the vicinity to the global coverage. The systems were originally designed in the last century, but they will continue to run for at least another couple of years without major updating of the user-side equipment. This thesis focuses on one of the systems, AIS. In this thesis, new techniques for improving the service quality of AIS on satellites are proposed. On the way, we develop several more broadly applicable algorithms in signal processing.

1.2 AIS

AIS adopts a self-organized TDMA (SOTDMA) technique as its main channel access protocol. The word “self-organized” means that all the stations (users) in this system are not necessarily being centrally controlled by base stations. Stations can be naturally grouped into SOTDMA cells without a base station. In one SOTDMA cell and in ideal conditions, the transmission is well-organized and ought to be free from packet collisions because in current transmission, stations tell each other when the next transmission will be conducted. In practice, the organization of transmission is dynamic and never stable, thus two or more stations can happen to transmit into the same time slot and interfere each other. The frequency of occurrence of such a packet collision can be maintained at a low level, thus causing no harm to the majority of the system. The size of a SOTDMA cell is determined by the averaged maximum distance at which two ships can talk to each other on the sea surface. It is theoretically a circle, the radius of which is about 20 nmi. Normally, this regular shape does not appear in reality. SOTDMA cells are much smaller and of irregular shapes, especially near harbors and inland rivers. Due to this fact, the transmission in adjacent SOTDMA cells interfere each other, resulting in low-quality receptions. AIS is vulnerable to cochannel interferences from itself (packets are strong interferences to each other) as no channel coding is used in its physical layer unlike the many modern wireless communication systems. It is known that AIS requires 15 dB [1–3] of the signal-to-noise ratio (SNR) to ensure reliable communication if standard differential decoding is applied.

1.3 Satellite AIS and Nonstandard AIS Stations

The information exchanged in AIS is encapsulated in data packets, called AIS messages (a block of data). A default AIS message occupies a single time slot.

The information embedded in the messages are valuable to scientific research institutes in marine safety and traffics, providers offering tracking data, government departments engaged in national security and etc.. Their common interest concentrates on tracking ships everywhere on earth. The goal of expansion to the global service coverage requires additionally creating a large-scale network of stations for all the cells on the earth's surface, due to the limitation of the protocol of AIS. The protocol targets for exchanging data locally only in a small area. It is an incredibly heavy task to build these base stations. The goal is impossible to achieve by only relying on coastline and inland base stations, and it is unpractical to build stations deep in the oceans. At the beginning, some primary attempts were made by building base stations on top of hills, cliffs and lighthouses beside the sea. This provides an area coverage up to a distance around 100 nmi. The service providers are not satisfied with the coverage.

Instead, the proposal of putting base stations on low-earth-orbit (LEO) satellites was raised. The proposal described a new system, called Satellite AIS (See Fig. 1.1), which is not a standard AIS but an additional complement for Terrestrial AIS (this is the original AIS). In Satellite AIS, base stations are carried by LEO satellites. Typically, the satellites circle around the earth at a speed of about seven kilometers per second in the orbit hundreds of kilometers above the ground. A simple arithmetic shows that the footprint or the field of view (FoV) of one LEO satellite could spread up to a circle, the radius of which can be more than 1,000 nmi. In a normal setup, a polar earth orbit is chosen to meet a regular updating of all parts of the surface of the earth. A careful construction of a chain of LEO satellites would continuously monitor the earth in realtime. The LEO satellites can stay in operation in orbit for years, which could average down the cost of sending up satellites into the space.

The high speed of LEO satellites creates distortion in the frequency domain: the spectrum moves to the upper side of the baseband when the satellites are approaching the ships in front of it, and the spectrum moves to the lower side of the baseband when the satellites are leaving the ships behind it. A simple receiver handling real-valued signals could face the difficulty in tracking the frequency drift as the frequency of signals from ships could be distinct and the transient of frequency from messages to messages could be abrupt. The frequency drift of each signal must be estimated separately and calibrated. As the complex-signals are analytic and allow frequency calibration in digital processing, a complex-valued receiver taking advantage of both I/Q signals should be used. This thesis deals with such complex-valued signals.

Another critical issue of Satellite AIS is that the anti-collision protocol of AIS loses its effectiveness at the side of the satellites. The scheduled transmission in ground cells is seen by the satellites not aligned to each other because the propagation time delays of signals from the cells to the satellite are not the same with a high probability. Messages in adjacent time slots from cells apart from each other sometimes cross the boundary of the time slots, causing a partial overlapping case (called Type 2 collision shown in Fig. 1.2). In addition, the

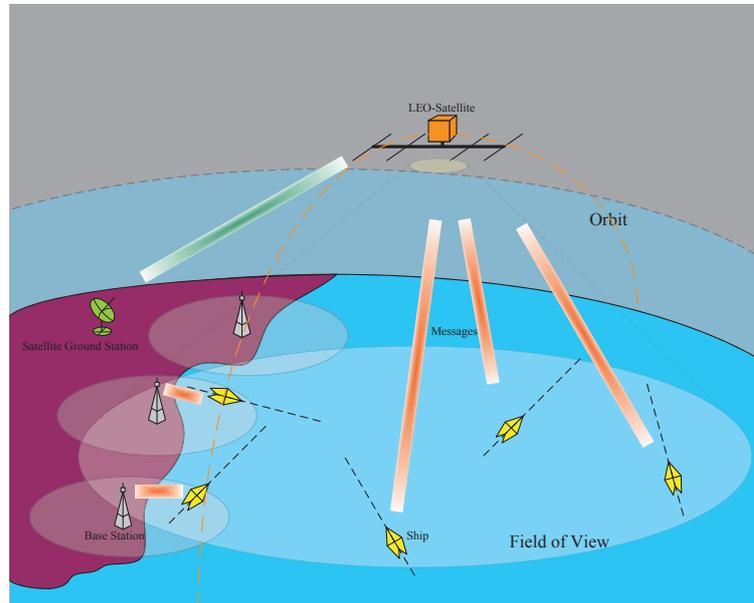


Figure 1.1: *Satellite AIS.*

ships in one cell do not know the occupation of the time slot associated with the ships in another cell, so the messages can be transmitted into the same time slot, and thus overlap each other, causing a full overlapping case (called Type 1 collision shown in Fig. 1.3).

Some methods have been proposed to combat the challenges of Satellite AIS. One method is to use a single high directional antenna (e.g., a helix antenna) to narrow down the size of the FoV of a satellite. This would alleviate the frequency distortion and reduce the occurrence of message collision at the cost of a much smaller real-time service coverage of a single satellite, resulting in a lower quality of service. Another method is to use an array of dipole antennas (unidirectional antennas) to separate the collided messages in spatial domain when they come from distinct directions (the probability is high). This method makes the system very complex when a large number of antennas are needed. This is the case for busy areas like the north coast of Europe as the AIS channels are over saturated. Based on the advantages of these two methods, one can use an array of directional (not necessarily high directional) antennas to achieve a balance between the real-time service coverage and the level of challenges. This thesis is interested in an array of less than 20 antennas. For instance, one can accept the construction of 8 antennas on the LEO satellite.

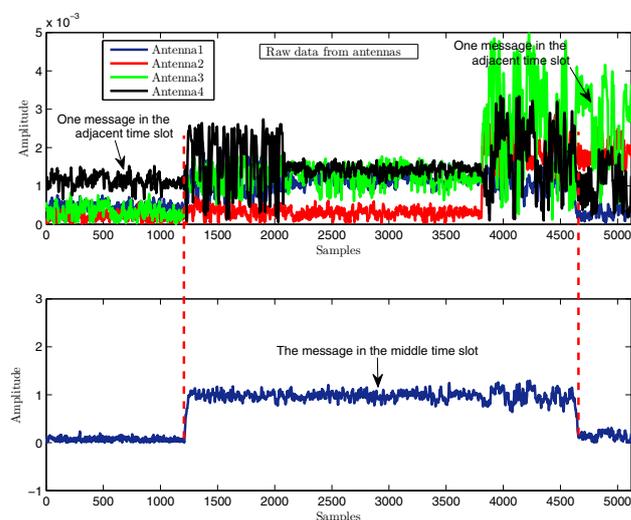


Figure 1.2: This figure shows one example of Type 2 collision from the hardware receiver presented in Chapter 6. The subfigure at the top shows the raw data from four antennas. The subfigure at the bottom shows the separated message in the middle time slot by the beamforming technique proposed in Chapter 4. There are two incomplete messages in the adjacent time slots at the left-end and right-end of the figure.

1.4 Review of Literature

The research of Satellite AIS [4] was started a few years ago. Some papers raised the topic of Satellite AIS [4] and did primary and useful work on LEO satellites [5–7]. A few reports and papers reported the simulation and tests of receivers with a single antenna on LEO satellites in orbit, where the practical ship distribution is taken into account. The modeling and analysis of both types of message collisions are conducted in [4, 6, 7] with a single antenna and in [8] with consideration of anti-collision techniques. Subsequent work addressed receivers with a single antenna, including discussions on signal recovery from Doppler shift, demodulation by special correlation [9] and CRC decoding [10].

The idea of using antenna arrays was proposed by [5]. The advantages of using antenna arrays are attractive because antenna arrays allow better reception quality by suppressing the interference from the traffic covered by Terrestrial AIS. However, in [5], the simulation, the beamforming process is not done in realtime.

Further research on (new) beamforming techniques is needed for improving the reception. This thesis researches and proposes algorithms for this need. The algorithms are proposed based on SURV (See Chapter 2), the generalized singular decomposition (GSVD) [11] and algebraic constant modulus algorithm (ACMA)

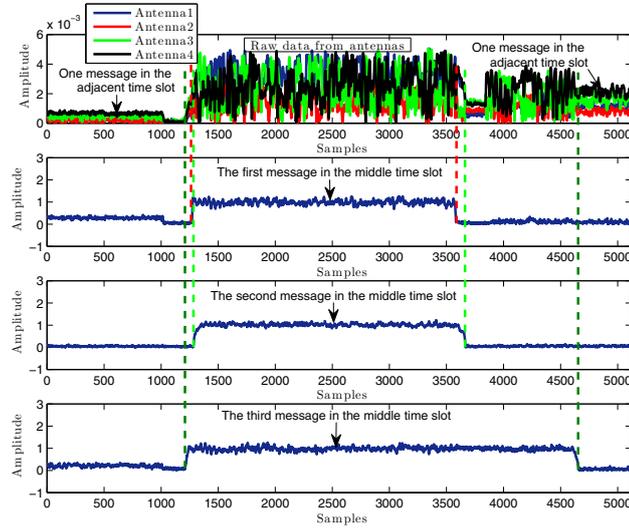


Figure 1.3: This figure shows one example of Type 1 collision from the hardware receiver presented in Chapter 6. The subfigure at the top shows the raw data from four antennas. The other three subfigures show each of the three separated messages in the middle time slot by the beamforming technique proposed in Chapter 4. There are two incomplete messages in the adjacent time slots at the left-end and right-end of the figure (Mixed with an Type 2 collision).

[12]. The detailed contribution of this thesis is shown in the next subsection. Some parallel works on beamforming to ours were proposed [13], where a classic beamforming technique on regular antenna arrays is proposed.

1.5 Goal of the Thesis

The goal of the thesis is to design and realize a feasible and effective beamforming technique for the receivers in Satellite AIS. The receivers are built by/from several basic tools, which are stated as follows:

1. A low-complexity beamforming technique to improve the signal-to-noise ratio.
2. A low-complexity beamforming technique to suppress interference and increase the signal-to-interference ratio (SIR).
3. A reconfigurable software testbed for simulating the receiver in Satellite AIS.

4. A hardware platform to primarily validate the techniques.

Our approach is to develop blind source separation algorithms to separate partially overlapping data packets. The blind approach is based on the following assumptions, which are realistic for the application of Satellite AIS:

1. The considered signals are narrowband. The signals are complex-valued at the baseband. The signals in each channel do not have to have structured phase responses. The direction of arrival (DOA) of signals may not exist.
2. The antennas and the related RF channels are coarsely calibrated so that the characteristic factors of the channels are not far from each other. In other words, noise levels and hardware gains are very close.
3. Signals are uncoded, that is, no channel coding or code division is used.
4. No training sequences are available to directly separate overlapping signals.
5. Signals are intermittent and transmitted in a form of short packets. In a short time period, the scenario could be block stationary but it is vulnerable to random time delays, where it turns into a nonstationary scenario.
6. All signals are wanted including those called targets or interferences in some chapters.

These are the “blind” conditions for this thesis. The chapters of the thesis discuss under all or part of the conditions.

In a block stationary scenario, especially where a single target signal is present, the main purpose of beamforming is to strengthen the only signal against noise. In array signal processing, the channel vector constructed by phase responses of one signal on the antenna array spans a rank-1 linear subspace in linear algebra unlike the additive white noise which is unidirectional and occupies the full space. The rank-1 subspace is named the signal subspace or the principal subspace as it is associated with strong power. The subspace orthogonal to the principal subspace is called the noise subspace or the minor/null subspace. Sometimes, a group of signals can be treated as “the target”, so the principal subspace can be multidimensional and of rank larger than 1.

The key part of this beamforming is to compute the channel (response) vector corresponding to the target signal. The singular value decomposition (SVD) is the standard tool to find the channel vector based on sample data matrices. The cubic order computational complexity of SVD is high and the process of computation is iterative, which is not friendly to hardware implementation. Various alternatives are proposed for replacing SVD under specific assumptions. It is noticed that most of them are not designed for the block scenario considered in this thesis: the number of target signals (or the rank of the principal subspace) is not known. The most relevant one is the URV decomposition, which works on exponential sliding windows. The URV has numerical issues when it works on rectangular

sliding windows. From a more generalized angle, the Hyperbolic URV (HURV) decomposition is proposed to try to replace URV, but HURV still has an issue of numerical stability, which is the key problem that prevents it from being used in practical applications. In Chapter 2, a new stable algorithm based on HURV, the signed URV algorithm (SURV), is proposed. SURV directly provides an efficient tracking form to reduce the updating complexity during tracking the principal subspaces.

In AIS, in areas with heavy traffic, cells are kept small to accommodate more ships. The transmission coordinated independently in adjacent cells may interfere each other. In Satellite AIS, the packets that are transmitted into a time slot are all signals needed to be received. Some of the packets can partially overlap the time slot and make the scenario nonstationary. These packets are treated temporarily as interference signals in the time slot while other packets are treated as the target signals. In such a nonstationary scenario, beamforming must achieve the purpose of not only strengthening the target signals but also suppressing the interference. The oblique projection is the standard tool to compute the beamformer when the target signal subspace and the interference subspace are a priori knowledge. The spans of these subspaces (but mixed together) can be obtained from the generalized singular value decomposition (GSVD). Clearly, a new tool to identify the subspaces is needed. In Chapter 3, an algorithm using the GSVD and SURV to identify the subspaces in one short observation period is proposed. In Chapter 4, an efficient tracking algorithm based on SURV to reduce the complexity of the algorithms in Chapter 3 is proposed.

The development of new beamforming techniques covers the first (theoretical) part of the the thesis. The process of building a software simulation model and that of developing an experimental hardware platform cover the second part and are presented in Chapter 5 and Chapter 6.

1.6 Thesis Outline and Contributions

This thesis is organized as a sequence of steps regarding the development of the beamforming techniques.

Chapter 2: A new tool for the principal subspace tracking, SURV, is introduced. In this chapter, the basic theory of SURV is proven and the updating steps of SURV are given. SURV is a stable and efficient decomposition to compute Schur subspace estimate-2 (SSE-2). When it is applied to the principal subspace tracking, SSE-2 can be constrained to be very close to the principal subspace. The principal subspace tracking computes the principal subspaces at a low computational complexity for the rank reduction step of array signal processing, e.g., source separation. The data model in this chapter is a generic classic model for linear arrays. This chapter shows how the stable algorithm, SURV, is found, where SURV implicitly imposes constraints on HURV such that important norm bounds that guarantee stability are satisfied. These constraints make the sub-

space estimate close to the principal subspace provided by the SVD (which is a special case within this class).

The complexity of the algorithm is of the same order as that of a QR update. Updating and downdating are of the same complexity and are both numerically stable. SURV is proven to provide rank estimates consistent with the SVD with the same rank threshold. It can replace the SVD where only subspace estimation is needed. Typical applications would be the detection of the number of signals in array signal processing, and subspace estimation for source separation and interference mitigation, such as the first step in MUSIC and ESPRIT-type algorithms. Simulation results demonstrate the numerical stability and confirm that this algorithm provides exact rank estimates and good principal subspace estimates as compared with the SVD.

Chapter 3: The separation of partially overlapping data packets by an antenna array in narrowband communication systems is discussed. The technique is originally designed for AIS but it can be a unified beamforming technique for the same problems in other asynchronous communication systems and several transponder systems such as RFID, SSR and ADS-B, where training sequences are not reliable for source separation under interferences. Partially overlapping data packages also occur as inter-cell interference in mutually unsynchronized communication systems. Arbitrary arrival times of the overlapping packets cause nonstationary scenarios and make it difficult to identify the signals using standard blind beamforming techniques. After selecting an observation interval, we propose subspace-based algorithms to suppress partially present (interfering) packets, as a preprocessing step for existing blind beamforming algorithms that assume synchronous or continuous overlap. The proposed algorithms are based on subspace intersections and oblique projections, computed using the generalized SVD (GSVD) and the generalized eigenvalue decomposition (GEVD). In the second part of the chapter, these algorithms are refined using the SURV algorithm. Simulation results show that the proposed algorithms significantly improve the performance of the classic algorithms designed for block stationary scenarios in cases where asynchronous co-channel interference is present. An example on experimental data from the AIS ship transponder system confirms the effectiveness of the proposed algorithms in a real application.

Chapter 4: A multi-user receiver using a new and simple blind beamforming technique based on the results in Chapter 3 is proposed, and it is used as a preprocessor that suppresses partially overlapping data blocks before the standard AIS receivers are used to decode the synchronous messages. The technique is a general tool that does not require a calibrated array and does not use the AIS signal modulation structure, but only relies on the finite size of interfering data blocks. Besides, the technique uses the straightforward tracking form of the SURV. The corresponding receiver is compared with previous receivers. The simulation results show that the proposed receiver is robust in the presence of partially overlapping data blocks and effectively improves the detection of AIS signals in Satellite AIS.

Chapter 5: The software testbed for the receivers for Satellite AIS is presented. This testbed is built by using several tools including SystemC-AMS and MATLAB. The software implementation of the receiver is done in MATABL and then translated into C++. The focus of this testbed is the construction of the physical model of the wireless channel and the transmission with the SOTDMA protocol in the network of Satellite AIS. This part is done in SystemC-AMS. The highlights of the SystemC-AMS part includes integrating the practical density of global ship distribution from open databases with satellite orbit designs from launched AIS satellites.

Chapter 6: The setup of an experimental hardware platform is presented. This platform uses an array of modified commercial RF frontends to downconvert the AIS signals to baseband. Sampled data are fed into PC and processed in MATLAB. The decoded AIS messages are analyzed and visualized on maps.

The thesis work has led to the following publications.

• **Journal Papers:**

- [J1] M. Zhou and A.-J. van der Veen, “Stable subspace tracking algorithm based on a signed URV decomposition,” *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3036–3051, Jun. 2012.
- [J2] M. Zhou and A.-J. van der Veen, “Blind beamforming techniques for separating partially overlapping data packets,” *submitted*, 2015.
- [J3] M. Zhou and A.-J. van der Veen, “Robust multi-user receiver for satellite AIS using blind beamforming techniques and tracking,” *submitted*, 2015.

• **Conference Papers:**

- [C1] M. Zhou, R. van Leuken, and H. J. L. Arriëns, “Modeling a configurable resistive touch screen system using SystemC and SystemC-AMS,” in *20th annual workshop on circuits, systems and signal processing-ProRISC*, Nov. 2009, pp. 393–398.
- [C2] M. Zhou and R. van Leuken, “Modeling an uplink for multi-antenna satellite receivers in AIS-like networks using SystemC-AMS,” in *21st annual workshop on circuits, systems and signal processing-ProRISC*, Nov. 2010.
- [C3] M. Zhou and A.-J. van der Veen, “Stable subspace tracking algorithm based on signed URV decomposition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 2720–2723.
- [C4] M. Zhou and A.-J. van der Veen, “Improved subspace intersection based on signed URV decomposition,” in *Proc. of the Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, California, USA, Nov. 2011.

- [C5] M. Zhou and R. van Leuken, “SystemC-AMS model of a dynamic large-scale satellite-based AIS-like network,” in *Forum on specification and Design Languages*, Oldenburg, Germany, Sept. 2011, pp. 24–31.
- [C6] M. Zhou and A. van der Veen, “Improved blind separation algorithm for overlapping secondary surveillance radar replies,” in *Proc. IEEE CAMSAP*, San Juan, Puerto Rico, Dec. 2011, pp. 181–184.
- [C7] M. Zhou, A.-J. van der Veen, and R. van Leuken, “Multi-user LEO-satellite receiver for robust space detection of AIS messages,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012.
- [C8] M. Zhou and A.-J. van der Veen, “A subspace tracking algorithm for separating partially overlapping data packets,” in *Proc. of the Seventh IEEE Sensor Array and Multichannel Signal Process. Workshop (SAM)*, Hoboken, NJ, USA, Jun. 2012 (**A nomination for the best student paper**. An award for entering the final candidates).
- [C9] A.-J. van der Veen and M. Zhou, “Subspace intersection tracking using GSVD and the signed URV algorithm,” in *TUM-IAS Workshop on Novel Numerical Methods*, TUM Institute for Advanced Study (TUM-IAS), Lichtenbergstraße 2 a, Garching, July 2013.
- [C10] M. Zhou and A.-J. van der Veen, “Bind beamforming techniques for automatic identification system using GSVD and tracking,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014.
- [C11] A.-J. van der Veen and M. Zhou, “Subspace intersection tracking using the signed URV algorithm,” in *ITG Workshop on Smart Antennas*, TU-ilmenau, Ilmenau, Germany, Mar. 2015 (**Keynote**).

Chapter 2

Signed URV Decomposition and Principal Subspace Tracking

2.1 Introduction

Fast adaptive subspace estimation and tracking plays an important role in modern signal processing. It forms the key ingredient in many algorithms, such as adaptive filtering, system identification, blind channel estimation, and blind signal separation and equalization algorithms [28], [29], [30].

Typically, in these applications we are interested in splitting the space into the principal subspace and the minor subspace. These two subspaces are spanned by the singular vectors corresponding to the singular values larger than and smaller than a threshold, respectively. Thus, the singular value decomposition (SVD) is commonly used for computing the desired subspaces. Because directly computing and updating the SVD is very expensive, many computationally cheaper and faster subspace tracking methods have been proposed, such as the Rank Revealing QR [31, 32] and the URV [33, 34]. The literature related to the topic of subspace tracking is extremely rich, and a brief overview of the literatures is provided later in this section.

An important observation is that, for splitting the space, we do not need the SVD. Let \mathbf{X} be an $m \times n_2$ data matrix and γ a threshold, and call d the number of singular values of \mathbf{X} larger than γ . Then $\mathbf{X}\mathbf{X}^H - \gamma^2\mathbf{I}$ has d eigenvalues larger than 0, and $m - d$ eigenvalues smaller than 0 (\mathbf{I} denotes an identity matrix; \mathbf{X} denotes a complex matrix, and the superscript H denotes the complex conjugate transpose of a matrix.) Instead of the SVD, we can compute the signed Cholesky factorization

$$\mathbf{X}\mathbf{X}^H - \gamma^2\mathbf{I} = \mathbf{B}\mathbf{B}^H - \mathbf{A}\mathbf{A}^H, \quad (2.1)$$

⁰published in IEEE Trans. Signal Process., vol. 60, no. 6, pp. 3036–3051, Jun. 2012. [14]

where \mathbf{A} , \mathbf{B} have minimal dimensions. Although \mathbf{A} and \mathbf{B} are not unique, based on Sylvester's inertia theorem, it is known that \mathbf{A} has size $m \times (m - d)$ and \mathbf{B} has size $m \times d$. Moreover, for any matrix \mathbf{M} such that $\|\mathbf{M}\| \leq 1$, it can be shown [35] that the columns of

$$\mathbf{U} = \mathbf{B} - \mathbf{A}\mathbf{M} \quad (2.2)$$

span a rank- d "principal subspace" in the sense that there is a corresponding approximant $\hat{\mathbf{X}}$ with column span $\text{ran}(\mathbf{U})$ (obtained e.g., by projecting onto this column span) such that

$$\|\mathbf{X} - \hat{\mathbf{X}}\| \leq \gamma, \quad (2.3)$$

where $\|\cdot\|$ denotes the matrix 2-norm (largest singular value). The matrix \mathbf{M} gives a complete parametrization of all such subspaces [35]; the principal subspace obtained by the SVD is within the class but is not explicitly identified. Note that \mathbf{B} is already a valid principal subspace estimate; Later in this chapter, the use of $\mathbf{M} \neq \mathbf{0}$ will be considered. The important point is that the required signed Cholesky factorization can be computed *implicitly* (by directly acting on \mathbf{X} and not forming $\mathbf{X}\mathbf{X}^H$) and *non-iteratively*, at a complexity comparable to that of a QR factorization. By acting column-by-column, the implicit decomposition can also be *updated* efficiently.

Indeed, the required \mathbf{A} and \mathbf{B} follow from the factorization

$$[\mathbf{N} \ \mathbf{X}]\Theta = [\mathbf{A} \ \mathbf{0} | \mathbf{B} \ \mathbf{0}], \quad (2.4)$$

where $\mathbf{N} = \gamma\mathbf{I}$, or any matrix such that $\mathbf{N}\mathbf{N}^H = \gamma^2\mathbf{I}$. The sign "|" in the right hand side matrix is a matrix separator which separates columns with a "positive signature" from those with a "negative signature". Also, Θ is a \mathbf{J} -unitary matrix as defined in Section 2.2: it satisfies $\Theta\mathbf{J}\Theta^H = \mathbf{J}$, where $\mathbf{J} = \mathbf{I} \oplus -\mathbf{I}$ is a signature matrix: a diagonal matrix with diagonal entries ± 1 . In this chapter, some algorithms for computing and updating such a decomposition will be considered.

Straightforward generalizations are possible. Let \mathbf{N} be any matrix, and define $\mathbf{R}_\mathbf{N} = \mathbf{N}\mathbf{N}^H$, then the factorization (2.4) provides an implicit signed Cholesky factorization of $\mathbf{X}\mathbf{X}^H - \mathbf{R}_\mathbf{N}$, leading to minimal-rank approximants $\hat{\mathbf{X}}$ such that

$$\|\mathbf{R}_\mathbf{N}^{-1/2}(\mathbf{X} - \hat{\mathbf{X}})\| \leq 1. \quad (2.5)$$

Such "whitened" approximants and corresponding subspace estimates are often needed, for example, in case of an array with uncalibrated antennas (resulting in different noise powers per antenna) [36], or in case of spatially correlated noise fields. In some cases, a nonwhite $\mathbf{R}_\mathbf{N}$ can be due to interference. In signal processing applications, \mathbf{X} is a noisy data matrix, measured column-by-column, and the columns of \mathbf{N} could be generated by taking samples from a noise process similar to the disturbance on \mathbf{X} (e.g., from a reference channel, or taken at a slightly different frequency where the desired signal is absent). Moreover, "downdating"

(recomputing the decomposition after removing a column from \mathbf{X}) can be done by updating the decomposition using that column as an update for \mathbf{N} . In this way, sliding window subspace tracking algorithms are easily implemented.

In this chapter, a new algorithm is proposed for computing and updating the decomposition (2.4) as more columns of \mathbf{X} and/or \mathbf{N} become available. The algorithm is called the “Signed URV” (SURV). In fact, by introducing a QR factorization of $[\mathbf{A}, \mathbf{B}]$, the algorithm computes the two-sided decomposition

$$\mathbf{Q}^H[\mathbf{N} \ \mathbf{X}]\Theta = [\mathbf{R}_A \ \mathbf{0} \mid \mathbf{R}_B \ \mathbf{0}], \quad (2.6)$$

where $[\mathbf{R}_A, \mathbf{R}_B]$ is lower triangular and \mathbf{Q} is a unitary matrix. This factorization may be called a Hyperbolic URV (HURV).¹ Generally, its computation is done by using hyperbolic rotations, which unfortunately can cause numerical problems as the result of the rotation does not need to be bounded. An algorithm for (2.6) was previously shown in [38], which requires 3 hyperbolic rotations per update vector, and thus is not always numerically stable. In the derivation of the new SURV algorithm, it is carefully designed that at most 1 hyperbolic rotation is used per update, and in such a way that the result is numerically stable.

Note that (\mathbf{A}, \mathbf{B}) are not unique. Among the choices of the matrices (\mathbf{A}, \mathbf{B}) , interest will be put in a specific class of “unbiased” decompositions (called SSE-2 in [35]). It will be seen that these decompositions can be obtained by adding a certain linear constraint to Θ in the HURV decomposition (2.6). With the constraint, the decomposition satisfies

$$\text{ran}(\mathbf{B}) \subset \text{ran}(\mathbf{X}), \quad \|\mathbf{B}\| \leq \|\mathbf{X}\|, \quad (2.7)$$

$$\text{ran}(\mathbf{A}) \subset \text{ran}(\mathbf{N}), \quad \|\mathbf{A}\| \leq \|\mathbf{N}\|. \quad (2.8)$$

It will be seen that these norm bounds are the key to numerical stability, and it will be shown that the SURV algorithm implies that an estimate in the SSE-2 class can be obtained, which is close to the principal subspace provided by the SVD (which is a special case within this class). Therefore, the above norm bounds that guarantee stability are satisfied. It is especially remarkable that in this way a stable downdating algorithm for subspace tracking can be obtained.

The proposed algorithm is non-iterative, does not require condition estimations, and has a complexity of the same order as that of a QR factorization of the data matrix. In particular, given a threshold γ on the singular values of the data matrix, SURV tracks the exact rank of the subspace (number of singular values above the threshold), as well as an orthonormal basis of the principal subspace estimate, at a complexity of at most order $21m^2$ per update or downdate vector if the size of the vector is m .

¹Not to be confused with the “high-rank” URV, or *hurv*, algorithm presented in [37].

Context

The literature on subspace tracking is rich and growing, and the right choice of a tracking algorithm is strongly dependent on the application. Here, a brief classification is provided to position the proposed SURV algorithm into context. A recent more elaborate overview can be found in [39], and an older overview is in [30].

Algorithms can be classified based on various aspects. An important one is *complexity*. Algorithms of order $O(m^2)$ per update are often two-sided decompositions such as the Rank Revealing QR [31, 32], the URV [33, 34], or other techniques that track full subspace information in terms of an $m \times m$ orthogonal matrix containing the basis vectors and an $m \times m$ coefficient matrix usually being triangular. SURV is in this class. With proper loss of information, it is sometimes possible to track only the $m \times d$ principal subspace estimate along with a $d \times d$ coefficient matrix losing details on the minor subspace. This is interesting if the rank d is very small, and sufficient if the minor subspace is filled by white noise, or if there is a large gap in singular values between both subspaces. A prototype algorithm was proposed by Karasalo [40] with complexity $O(md^2)$ per update, as well as many techniques based on power iterations. Further reductions are possible if the subspace basis is not kept strictly orthogonal. For example, in [41] the PAST algorithm with complexity $O(md)$ per update is derived from a power iteration, and the algorithm uses an approximately orthogonal basis, which can converge to an orthogonal basis under stationarity assumptions. The speed of convergence in these cases usually depends on the gap in singular values between the two subspaces. There are many other algorithms in this class, such as PASTd [41], NIC [42], OPAST [43], etc., see also [44] for an overview.

Another aspect is *rank tracking*. Some algorithms simply assume that the rank d of the data matrix is known in advance. This is the case for the $O(md^2)$ and $O(md)$ algorithms, since they only track a basis for the principal subspace. In these algorithms, when the norm of a certain residual is too large, the rank has to increase; when the coefficient matrix becomes singular, the rank has to decrease. Therefore, these algorithms need some special operations to deal with rank changes as the missing information in the coefficient matrix has to be recovered, and some iterative algorithms such as rank estimators to detect rank changes [39, 45, 46]. Other algorithms are based on a threshold (tolerance) that sits in the gap between the large and small singular values; the rank is the number of the singular values above the threshold. URV is such an example, and the proposed SURV is also in this class. This is more natural for many applications in signal processing, where there may be *a priori* information on the noise powers, but not on the number of signals.

Essential to any tracking algorithm is a form of data *windowing*, which is the third aspect. The majority of adaptive algorithms use an exponential window, where prior to each update, the old data in the coefficient matrix is scaled by a factor $\beta < 1$. This is particularly suitable for fixed-rank algorithms. For detecting

sudden rank changes in non-stationary applications (e.g., the appearance of data frames in ad hoc communications), sliding window algorithms are more attractive. These perform an update with a new data vector and a downdate to remove the influence of an old data vector. Sliding window downdating algorithms have been extensively studied for two-sided $O(m^2)$ decompositions such as RRQR and URV, and have led to a series of papers discussing how to handle the underlying indefinite Cholesky factorization and the related hyperbolic transformations [33, 47–50]. A numerically stable URV is “very complex” [37], and critically depends on accurate condition estimators. However, a “relational stability” property can be proven [50]. For the $O(md)$ algorithms, some sliding window algorithms have been developed most recently, such as SW-PAST [51], SWASVD [52], and FAPI [53].

The proposed SURV algorithm has complexity $O(m^2)$, is based on the two-sided Hyperbolic URV decomposition in [35, 38], and is a sliding window algorithm. In this SURV algorithm, an orthogonal basis of both principal and minor subspaces is stored and tracked, and a threshold on the singular values is used to detect rank changes. No condition estimation is needed. Nonetheless, the exact rank is obtained with respect to a given tolerance. The performance of the subspace estimator using computationally less favorable algorithms for typical subspace tracking applications (DOA estimation) has been assessed and compared with SVD, RRQR and URV in [54, 55].

Outline

The chapter is organized as follows. In Section 2.2, the elementary rotations used in this chapter are introduced. The theoretical foundation for the proposed Hyperbolic URV decomposition is provided in Section 2.3. In Section 2.4, the SURV algorithm is proposed and its computational complexity is discussed. In Section 2.5, it is proven that the proposed updating algorithm has favorable numerical properties. The simulation results are provided in Section 2.6, and the conclusions are placed in Section 2.7.

2.2 J-unitary Matrices

In this section, some background materials on \mathbf{J} -unitary matrices from [35] are reviewed. A square matrix Θ is \mathbf{J} -unitary if it satisfies

$$\Theta^H \mathbf{J} \Theta = \mathbf{J}, \quad \Theta \mathbf{J} \Theta^H = \mathbf{J}, \quad (2.9)$$

where \mathbf{J} is a signature matrix which follows some prescribed $(p + q) \times (p + q)$ block-partitioning of Θ :

$$\Theta = \begin{matrix} & \begin{matrix} p & q \end{matrix} \\ \begin{matrix} p \\ q \end{matrix} & \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \end{matrix}, \quad \mathbf{J} = \begin{bmatrix} +\mathbf{I}_p & \\ & -\mathbf{I}_q \end{bmatrix}. \quad (2.10)$$

Here, \mathbf{I}_p denotes the $p \times p$ identity matrix, and p or q may be zero. If Θ is applied to a block-partitioned matrix $[\mathbf{A} \ \mathbf{B}]$, then (2.9) implies

$$[\mathbf{A} \ \mathbf{B}] \Theta = [\mathbf{C} \ \mathbf{D}] \Rightarrow \mathbf{A}\mathbf{A}^H - \mathbf{B}\mathbf{B}^H = \mathbf{C}\mathbf{C}^H - \mathbf{D}\mathbf{D}^H. \quad (2.11)$$

Hence, \mathbf{J} assigns a positive signature to the columns of \mathbf{A} , \mathbf{C} , and a negative signature to those of \mathbf{B} , \mathbf{D} . The $+$ and $-$ above matrices denote the positive and negative signatures of the columns in those matrices.

The \mathbf{J} -unitarity of Θ implies $\Theta_{22}^H \Theta_{22} = \mathbf{I} + \Theta_{12}^H \Theta_{12}$ and $\Theta_{22} \Theta_{22}^H = \mathbf{I} + \Theta_{21} \Theta_{21}^H$. These expressions can deduce the following properties [35]:

$$\begin{aligned} \Theta_{22} \text{ is invertible,} \quad & \|\Theta_{22}^{-1}\| \leq 1, \quad \|\Theta_{12} \Theta_{22}^{-1}\| < 1, \\ & \|\Theta_{22}^{-1} \Theta_{21}\| < 1. \end{aligned} \quad (2.12)$$

Similar properties hold for Θ_{11} :

$$\begin{aligned} \Theta_{11} \text{ is invertible,} \quad & \|\Theta_{11}^{-1}\| \leq 1, \quad \|\Theta_{11}^{-1} \Theta_{12}\| < 1, \\ & \|\Theta_{21} \Theta_{11}^{-1}\| < 1. \end{aligned} \quad (2.13)$$

For updating purposes, it will be necessary to work with column permutations of $[\mathbf{A} \ \mathbf{B}]$ and $[\mathbf{C} \ \mathbf{D}]$, which induces row and column permutations of Θ . Thus it is necessary to introduce more general matrices Θ that are $(\mathbf{J}_1, \mathbf{J}_2)$ -unitary with respect to *unsorted* signature matrices $\mathbf{J}_1, \mathbf{J}_2$ satisfying $\Theta^H \mathbf{J}_1 \Theta = \mathbf{J}_2$, $\Theta \mathbf{J}_2 \Theta^H = \mathbf{J}_1$, where \mathbf{J}_1 and \mathbf{J}_2 are diagonal matrices with diagonal entries equal to ± 1 . If $\mathbf{M}\Theta = \mathbf{N}$, then $\mathbf{M}\mathbf{J}_1\mathbf{M}^H = \mathbf{N}\mathbf{J}_2\mathbf{N}^H$, so that \mathbf{J}_1 associates its signature to the columns of \mathbf{M} , and \mathbf{J}_2 associates its signature to the columns of \mathbf{N} . By inertia, the total number of positive entries in \mathbf{J}_1 has to be equal to that in \mathbf{J}_2 , and likewise for the negative entries. It is always possible to return to sorted signature matrices by applying suitable column and row permutations to Θ .

Any Θ can be constructed from a sequence of elementary 2×2 “rotations”. It is similar to constructing an big unitary matrix from 2×2 Givens rotations. Let $\mathbf{j}_1 = \text{diag}[\pm 1, \pm 1]$ be an (unsorted) 2×2 signature matrix, and similar for \mathbf{j}_2 . A 2×2 matrix θ is an elementary $(\mathbf{j}_1, \mathbf{j}_2)$ -unitary rotation if it satisfies $\theta^H \mathbf{j}_1 \theta = \mathbf{j}_2$, $\theta \mathbf{j}_2 \theta^H = \mathbf{j}_1$. Similar to Givens rotations, it can be used to zero specific entries of vectors: for a given row vector $[r \ x]$ and signature \mathbf{j}_1 , the proper θ and r' can be found so that $[r \ x]\theta = [r' \ 0]$. The “input signature” $\mathbf{j}_1 = \text{diag}[(j_1)_1 \ (j_1)_2]$ associates with $[r \ x]$, and the “output signature” $\mathbf{j}_2 = \text{diag}[(j_2)_1 \ (j_2)_2]$ is such that $|r|^2(j_1)_1 + |x|^2(j_1)_2 = |r'|^2(j_2)_1$. This determines $(j_2)_1$. The second entry $(j_2)_2$ follows by inertia considerations: the number of $+1$ and -1 entries in \mathbf{j}_1 and \mathbf{j}_2 are equal.

The precise form of θ depends on the input signature \mathbf{j}_1 , and also on whether $|r| > |x|$ or $|r| < |x|$, as listed in table 2.1. Cases 1 and 2 in the table correspond to an indefinite signature \mathbf{j}_1 (i.e., $\text{diag}[+1, -1]$ or $\text{diag}[-1, +1]$), and the required θ is an hyperbolic rotation, whereas case 3 occurs when \mathbf{j} is definite which leads to an ordinary circular (unitary or Givens) rotation.

Table 2.1: *Elementary \mathbf{j} -unitary zeroing rotations.*¹

| |
|--|
| Input: $r, x, \mathbf{j}_1 = \text{diag}[\pm 1, \pm 1]$ |
| Output: $\boldsymbol{\theta}$ such that $\begin{bmatrix} r & x \end{bmatrix} \boldsymbol{\theta} = \begin{bmatrix} r' & 0 \end{bmatrix}, \mathbf{j}_2 = \text{diag}[\pm 1, \pm 1]$ |
| 1. If $\mathbf{j}_1 = \text{diag}[+1, -1]$ or $\mathbf{j}_1 = \text{diag}[-1, +1]$, and $ r > x $: (Hyperbolic rotation) $\boldsymbol{\theta} = \begin{bmatrix} 1 & -s \\ -s^* & 1 \end{bmatrix} \frac{1}{c}$ where $s = \frac{x}{r}$, $c = \sqrt{1 - s ^2}; \mathbf{j}_2 = \mathbf{j}_1; r' = cr.$ |
| 2. If $\mathbf{j}_1 = \text{diag}[+1, -1]$ or $\mathbf{j}_1 = \text{diag}[-1, +1]$, and $ r < x $: (Hyperbolic rotation) $\boldsymbol{\theta} = \begin{bmatrix} -s^* & 1 \\ 1 & -s \end{bmatrix} \frac{1}{c}$ where $s = \frac{r}{x}$, $c = \sqrt{1 - s ^2}; \mathbf{j}_2 = -\mathbf{j}_1$ (sign reversal); $r' = cx.$ |
| 3. If $\mathbf{j}_1 = \text{diag}[+1, +1]$ or $\mathbf{j}_1 = \text{diag}[-1, -1]$: (Givens rotation) $\boldsymbol{\theta} = \begin{bmatrix} c^* & -s \\ s^* & c \end{bmatrix}$ where $s = \frac{x}{\sqrt{ r ^2 + x ^2}}$, $c = \sqrt{1 - s ^2}; \mathbf{j}_2 = \mathbf{j}_1; r' = \sqrt{ r ^2 + x ^2}.$ |

¹ * denotes the complex conjugate operator.

Situations where $|r| = |x|$ with an indefinite signature \mathbf{j}_1 are degenerate: the result $\begin{bmatrix} 0 & 0 \end{bmatrix}$ is well defined but $\boldsymbol{\theta}$ is unbounded, and the output signature \mathbf{j}_2 is arbitrary.

2.3 Hyperbolic URV Decomposition

Let $\mathbf{N} : m \times n_1$ and $\mathbf{X} : m \times n_2$ be given matrices. Define the implicit factorizations $\mathbf{X}\mathbf{X}^H - \mathbf{N}\mathbf{N}^H$ as

$$\mathbf{X}\mathbf{X}^H - \mathbf{N}\mathbf{N}^H = \mathbf{B}\mathbf{B}^H - \mathbf{A}\mathbf{A}^H, \quad (2.14)$$

where \mathbf{A} and \mathbf{B} are nonsingular and together have m columns. \mathbf{A} and \mathbf{B} follow from the factorization

$$m \begin{bmatrix} \overset{+}{\mathbf{N}} & \overset{-}{\mathbf{X}} \end{bmatrix} \boldsymbol{\Theta} = m \begin{bmatrix} \overset{+}{\mathbf{A}_e} & \overset{-}{\mathbf{B}_e} \end{bmatrix}, \quad (2.15)$$

where $+$ and $-$ above the matrices denote the positive and negative signatures of the corresponding columns,

$$\mathbf{A}_e = m \begin{bmatrix} \overset{m-d}{\mathbf{A}} & \overset{n_1-}{m+d} \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_e = m \begin{bmatrix} \overset{d}{\mathbf{B}} & \overset{n_2-d}{\mathbf{0}} \end{bmatrix}, \quad (2.16)$$

and Θ is a \mathbf{J} -unitary matrix partitioned conformably as

$$\Theta = \begin{matrix} & \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \end{matrix}, \quad \mathbf{J} = \begin{bmatrix} +\mathbf{I}_{n_1} & \\ & -\mathbf{I}_{n_2} \end{bmatrix}. \quad (2.17)$$

The factorization (2.15) always exists although Θ is unbounded when $\mathbf{X}\mathbf{X}^H - \mathbf{N}\mathbf{N}^H$ is singular [35]; this corresponds to the case where a singular value of $\mathbf{R}_{\mathbf{N}}^{-1/2}\mathbf{X}$ is exactly equal to 1. However, the factorization is not unique, and even the subspaces $\text{ran}(\mathbf{A})$ and $\text{ran}(\mathbf{B})$ are not unique.

A straightforward way to find a factorization (2.15) is by the hyperbolic QR factorization (HQR) [56], [35], [57]

$$\begin{matrix} n_1 & n_2 & m & n_1+n_2 \\ + & - & \pm & -\pm \\ \begin{bmatrix} \mathbf{N} & \mathbf{X} \end{bmatrix} \Theta & = & \begin{bmatrix} \mathbf{R} & \\ & \mathbf{0} \end{bmatrix}, \end{matrix} \quad (2.18)$$

where the sign “ \pm ” indicates an unsorted signature, \mathbf{R} is a lower or upper triangular $m \times m$ matrix, and Θ is $(\mathbf{J}_1, \mathbf{J}_2)$ -unitary. Here, $\mathbf{J}_1 = \mathbf{J} = \mathbf{I}_{n_1} \oplus -\mathbf{I}_{n_2}$ is the sorted signature given from the outset, and \mathbf{J}_2 is an unsorted signature matrix which associates with the right hand side of (2.18). Although this factorization is straightforward to compute and update, it has the drawback that it does not always exist: the triangular form of \mathbf{R} is too restrictive [35]. Although the set of exceptions is finite, in the neighborhood of an exception the resulting \mathbf{R} may become numerically unbounded and nearly rank deficient, equivalently \mathbf{A} and \mathbf{B} may become unbounded with parts of their column nearly collinear, leading to cancelling each other during $\mathbf{B}\mathbf{B}^H - \mathbf{A}\mathbf{A}^H$ being formed.

To get around this, introduce a QR factorization of $[\mathbf{A} \ \mathbf{B}]$:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_A & \mathbf{R}_B \end{bmatrix} = \mathbf{Q}^H [\mathbf{A} \ \mathbf{B}] = \begin{bmatrix} \mathbf{Q}_A & \mathbf{Q}_B \end{bmatrix}^H [\mathbf{A} \ \mathbf{B}], \quad (2.19)$$

where \mathbf{R} is triangular and \mathbf{Q} is unitary. This leads to the more general two-sided decomposition [38]

$$\mathbf{Q}^H \begin{bmatrix} + & - \\ \mathbf{N} & \mathbf{X} \end{bmatrix} \Theta = \begin{bmatrix} + & + & - & - \\ \mathbf{R}_A & \mathbf{0} & | & \mathbf{R}_B & \mathbf{0} \end{bmatrix}. \quad (2.20)$$

Note that still $[\mathbf{A} \ \mathbf{0} \ | \ \mathbf{B} \ \mathbf{0}] = [\mathbf{N} \ \mathbf{X}] \Theta$. This two-sided decomposition always exists [35], although some columns of \mathbf{R} may be zero if \mathbf{A} and \mathbf{B} together are rank-deficient. The decomposition will be called a Hyperbolic URV (HURV).

In the next section, a stable algorithm to compute and update the HURV, called the Signed URV (SURV) algorithm, will be proposed. The decomposition is not unique, and hence, an additional constraint that leads to favorable properties (Section 2.5) will be added. In fact, the derived SURV algorithm will be shown to satisfy this constraint.

2.4 SURV: An Updating Algorithm for the Hyperbolic URV

2.4.1 Overview

In this section, an algorithm will be proposed to update the HURV (2.20) as new columns for \mathbf{X} and \mathbf{N} become available. With proper initialization, this also serves as an algorithm to compute the HURV. The 2×2 \mathbf{j} -unitary rotations specified in Table 2.1 will be used as building blocks (cf. Givens rotations). Requirements on the updating algorithm are (i) it should avoid the storage of Θ , as this matrix grows in dimensions and can potentially be unbounded in magnitude (even if the resulting $[\mathbf{A}, \mathbf{B}]$ are well defined), and (ii) the update should be done in a numerically stable way.

Several updating algorithms are possible, depending on one's objectives. Because of potential numerical instability associated with hyperbolic rotations, the updating steps are designed to use at most only one hyperbolic rotation per vector update, which corresponds to a single decision whether the dimension of the principal subspace grows, remains constant, or shrinks. Moreover, that hyperbolic rotation should occur only when all the other entries in the two vectors on which the rotation is acting are already zero, as discussed later, so that an unbounded rotation has no detrimental numerical effects. (In [38], an updating algorithm was proposed that had at most 3 hyperbolic rotations per update, which is not minimal, and that algorithm was not always numerically stable.)

For tracking the principal subspace $\text{ran}(\mathbf{B})$, it is necessary to keep $\mathbf{R} = [\mathbf{R}_A \ \mathbf{R}_B]$ in (2.20) lower triangular, so that the columns of \mathbf{Q}_B are an orthonormal basis of the principal subspace, and likewise the columns of \mathbf{Q}_A are an orthonormal basis of the minor subspace.

At the start, suppose that the decomposition has already been computed as $\mathbf{Q}^H[\mathbf{N} \ \mathbf{X}]\Theta = [\mathbf{R}_A \ \mathbf{0} | \mathbf{R}_B \ \mathbf{0}]$, where $\mathbf{R} = [\mathbf{R}_A \ \mathbf{R}_B]$ is square, lower triangular and sorted according to signature. In principle, updating the factorization with new columns of \mathbf{X} or \mathbf{N} is straightforward. Indeed, a new factorization needs to be found as $\mathbf{Q}'^H[\mathbf{N}' \ \mathbf{X}']\Theta' = [\mathbf{R}'_A \ \mathbf{0} | \mathbf{R}'_B \ \mathbf{0}]$, where either $\mathbf{N}' = [\mathbf{N} \ \mathbf{n}]$, $\mathbf{X}' = \mathbf{X}$ if it is needed to add a new column \mathbf{n} to \mathbf{N} (also known as a “downdate” where we want to remove the effect of a previous \mathbf{x}), or $\mathbf{N}' = \mathbf{N}$, $\mathbf{X}' = [\mathbf{X} \ \mathbf{x}]$ if it is needed to augment \mathbf{X} by \mathbf{x} (the usual “update”). All these cases will be referred as “updating”. The only difference is in the signature of the update vector. Making use of the previously computed decomposition, it suffices to find \mathbf{Q}_c and Θ_c such that

$$\mathbf{Q}_c^H \begin{bmatrix} m-d & d & 1 \\ + & - & j_c \\ \mathbf{R}_A & \mathbf{R}_B & \mathbf{c} \end{bmatrix} \Theta_c = \begin{bmatrix} m-d' & d' & 1 \\ + & - & j'_c \\ \mathbf{R}'_A & \mathbf{R}'_B & \mathbf{0} \end{bmatrix}, \quad (2.21)$$

$$\mathbf{Q}' := \mathbf{Q}\mathbf{Q}_c,$$

where $\mathbf{c} = \mathbf{Q}^H \mathbf{n}$ when adding a column \mathbf{n} to \mathbf{N} or $\mathbf{c} = \mathbf{Q}^H \mathbf{x}$ when adding a column \mathbf{x} to \mathbf{X} . (Note that it is necessary to store and update \mathbf{Q} for applying this transformation.) In the first case, \mathbf{c} has a positive signature $j_c = +1$; in the second

case, $j_{\mathbf{c}} = -1$. Denote the signature of \mathbf{R} by $\mathbf{J} = \mathbf{I}_{m-d} \oplus -\mathbf{I}_d = \text{diag}[j_1, j_2, \dots, j_m]$. Let c_k denote the k -th entry of \mathbf{c} and $r_{k,l}$ denote the (k,l) -th entry of \mathbf{R} . The rank of the principal subspace before the update is d , and after the update it is d' .

The updating algorithm consists of a sequence of elementary rotations for reducing $[\mathbf{R} \ \mathbf{c}]$ to $[\mathbf{R}' \ \mathbf{0}]$, i.e., to zero \mathbf{c} . In the algorithm, the $m \times m$ matrices \mathbf{Q} and \mathbf{R} are stored and tracked, but Θ is not stored.

2.4.2 Algorithm

To compute the factorization (2.21), the entries c_1, c_2, \dots, c_m of \mathbf{c} are zeroed in turn. According to the first three methods listed in Table 2.2, there are three methods to zero c_k : by elementary column rotations θ (Givens Column Rotations, GCR, and Hyperbolic Column Rotations, HCR), or by elementary row rotations \mathbf{q} in combination with Givens rotations θ (GRCR). The GCR and HCR are the most natural and efficient, as they directly zero c_k against $r_{k,k}$, using a rotation θ such that $[r_{k,k} \ c_k]\theta = [r'_{k,k} \ 0]$. However, in the case that the signatures of $r_{k,k}$ and c_k differ, the required rotation is hyperbolic, leading to potential numerical instability if $|c_k| \approx |r_{k,k}|$. To avoid this, the GRRCR scheme first computes an elementary circular (unitary) rotation \mathbf{q} to zero c_k against c_{k+1} , and then a θ -rotation to zero the resulting fill-in in $r_{k,k+1}$ against $r_{k,k}$. If the signatures of $r_{k,k+1}$ and $r_{k,k}$ are the same, then this rotation is a Givens rotation.

For numerical stability, it is desired to minimize the number of *hyperbolic* rotations, i.e., rotations θ that act on columns with unequal signatures. The ‘‘Signed URV’’ algorithm that is about to be proposed is designed such that at most 1 hyperbolic rotation (HCR) is needed, which corresponds to a single decision whether the rank of the subspace estimate should be increased or decreased. As will be shown later, this can make the algorithm numerically stable.

The algorithm is described as follows. At the start of the algorithm, the data is represented as in Fig. 2.1(a) for the case $j_{\mathbf{c}} = +1$, and in Fig. 2.3(a) for the case $j_{\mathbf{c}} = -1$. These cases are treated separately.

Table 2.2: Zeroing schemes for c_k

| | |
|--------------|---|
| GCR: | <p>Givens Column Rotations Apply only if $j_k = j_{\mathbf{c}}$:</p> <ol style="list-style-type: none"> 1. Compute Givens rotation $\boldsymbol{\theta}$ such that $[r_{k,k} \ c_k]\boldsymbol{\theta} = [* \ 0]$; 2. Apply $\boldsymbol{\theta}$ to the k-th column of \mathbf{R} and \mathbf{c} (no sign change). |
| HCR: | <p>Hyperbolic Column Rotations Apply if $j_k = -j_{\mathbf{c}}$:</p> <ol style="list-style-type: none"> 1. Set $\mathbf{j}_1 = \text{diag}[j_k, j_{\mathbf{c}}]$, and compute $\boldsymbol{\theta}$ and \mathbf{j}_2 such that $[r_{k,k} \ c_k]\boldsymbol{\theta} = [* \ 0]$; 2. Apply $\boldsymbol{\theta}$ to the k-th column of \mathbf{R} and \mathbf{c}; update signatures $j_k, j_{\mathbf{c}}$ following \mathbf{j}_2 (possible sign change). |
| GRCR: | <p>Givens Row and Column Rotations Apply only if $j_k = j_{k+1}$:</p> <ol style="list-style-type: none"> 1. Compute Givens row rotation \mathbf{q} such that $\mathbf{q}^H \begin{bmatrix} c_k \\ c_{k+1} \end{bmatrix} = \begin{bmatrix} 0 \\ * \end{bmatrix}$; 2. Apply \mathbf{q}^H to rows $(k, k+1)$ of $[\mathbf{R}, \mathbf{c}]$; apply \mathbf{q} to columns $(k, k+1)$ of \mathbf{Q}; 3. Compute Givens column rotation $\boldsymbol{\theta}$ such that $[r_{k,k} \ r_{k,k+1}]\boldsymbol{\theta} = [* \ 0]$; 4. Apply $\boldsymbol{\theta}$ to columns $(k, k+1)$ of \mathbf{R} (no sign change). |
| GRR: | <p>Givens Row Rotations to zero $r_{k,k+1}$ Apply only if $c_k = c_{k+1} = 0$:</p> <ol style="list-style-type: none"> 1. Compute Givens row rotation \mathbf{q} such that $\mathbf{q}^H \begin{bmatrix} r_{k,k+1} \\ r_{k+1,k+1} \end{bmatrix} = \begin{bmatrix} 0 \\ * \end{bmatrix}$; 2. Apply \mathbf{q}^H to rows $(k, k+1)$ of $[\mathbf{R}, \mathbf{c}]$; apply \mathbf{q} to columns $(k, k+1)$ of \mathbf{Q}. |

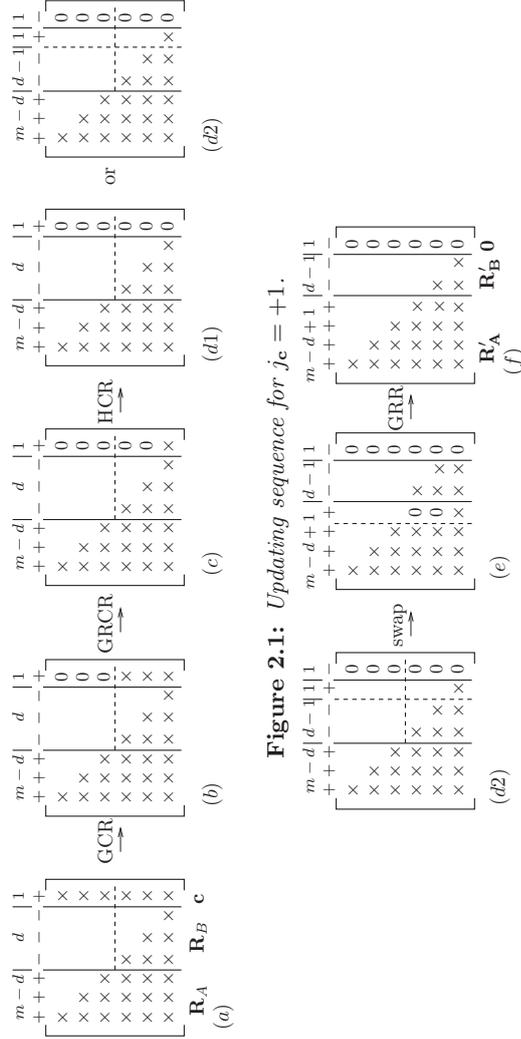


Figure 2.1: Updating sequence for $j_c = +1$.

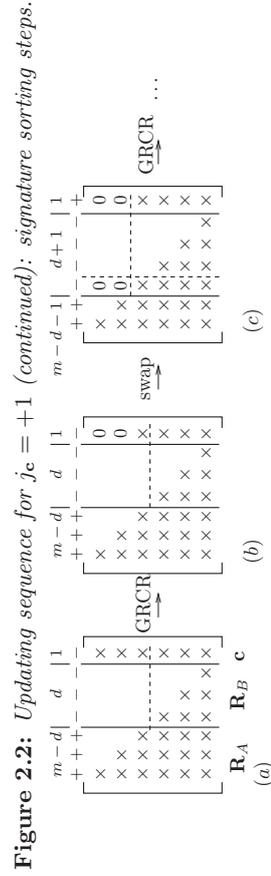


Figure 2.2: Updating sequence for $j_c = +1$ (continued): signature sorting steps.

Figure 2.3: Updating sequence for $j_c = -1$. Except for dimensions, (c) corresponds with Fig. 2.1(b), and the algorithm continues at that place.

- If $j_{\mathbf{c}} = +1$, then

(a) For $k = 1, \dots, m-d$, we have $j_{\mathbf{c}} = j_k = +1$. Thus, we cancel c_k against $r_{k,k}$ using GCR elementary rotations. The result is shown in Fig. 2.1(b). (Note that for $k > m-d$, the signature are unequal and GCR rotations are not applicable.)

If $d = 0$, then we are done, else we continue:

(b) For $k = m-d+1, \dots, m-1$, we have $j_{\mathbf{c}} = +1$, $j_k = -1$. Thus, we cancel c_k using GRCR. The result is in Fig. 2.1(c).

(c) At this point, only c_m is nonzero, and has to be cancelled against $r_{m,m}$. Since $j_{\mathbf{c}} = +1$ and $j_m = -1$, the signatures differ and a HCR is applied to cancel c_m .

If during the HCR the signatures stay constant ($|r_{m,m}| > |c_m|$, see Fig. 2.1(d1)), then we are done without rank changes ($d' = d$).² On the other hand, if $|r_{m,m}| < |c_m|$, see Fig. 2.1(d2), then the signatures are reversed and we continue: (see Fig. 2.2(d2))

(d2) Now $j_m = +1$ and $d' = d-1$: a rank decrease. As seen in Fig. 2.2(d2), the signatures are not sorted, as the last column of \mathbf{R} has a $+1$ signature and belongs to $\mathbf{R}_{\mathbf{A}}$. Thus move the last column \mathbf{r}_m of \mathbf{R} to fit at the end of $\mathbf{R}_{\mathbf{A}}$ and update the signatures in a corresponding way: $\mathbf{J}' = \mathbf{I}_{m-d'} \oplus -\mathbf{I}_{d'}$. See Fig. 2.2(e).

(e) Restore the lower triangular structure of \mathbf{R} using GRR (see Table 2.2), for $k = m-1$ down to $m-d+1$ (or $m-d'$). See Fig. 2.2(f). This concludes the update.

- If $j_{\mathbf{c}} = -1$ and $d < m$ (see Fig. 2.3(a)), then

(a) For $k = 1, \dots, m-d-1$, we have $j_k = +1$, whereas $j_{\mathbf{c}} = -1$. Thus, to avoid hyperbolic rotations, we cancel c_k against c_{k+1} using GRCR. (Entry c_{m-d} cannot yet be cancelled as it would lead to a hyperbolic rotation.) See Fig. 2.3(b).

(b) Swap column \mathbf{r}_{m-d} of \mathbf{R} with \mathbf{c} , also swap the corresponding signatures, and set $d := d+1$. This potential rank increase can later become undone. See Fig. 2.1(c).

Now continue as in the case $j_{\mathbf{c}} = +1$, step (b). In step (c) we apply an HCR, the rank may decrease again in which case $d' = d$ (and we continue with step (d2)), else we remain with $d' = d+1$ and we are done.

- If $j_{\mathbf{c}} = -1$ and $d = m$, then

(a) For $k = 1, \dots, m$, we have $j_k = -1$ and also $j_{\mathbf{c}} = -1$. Cancel c_k against $r_{k,k}$ using GCR. There are no rank changes.

²See later for the exceptional case where $|r_{m,m}| = |c_m|$.

The only hyperbolic rotation in the algorithm appears in Step (c). Depending on the sign change in this rotation (which occurs if $|r_{m,m}| < |c_m|$) and on the value of $j_{\mathbf{c}}$, the rank d of the subspace increases, decreases or remains constant. This directly corresponds to the number of singular values that are larger or smaller than the selected threshold. If at this stage $|r_{m,m}| = |c_m|$, i.e., a singular value is equal to the threshold, then the rank decision is arbitrary. In the algorithm, $\boldsymbol{\theta}$ will be unbounded but the result $[r_{m,m} \ c_m]\boldsymbol{\theta} = [0 \ 0]$ is well defined, and as $\boldsymbol{\theta}$ is not stored or used anyway, the result is numerically stable. This works because we took care that the hyperbolic rotation $\boldsymbol{\theta}$ acts on the last column of \mathbf{R} and \mathbf{c} , which are zero except for their last entries $r_{m,m}$ and c_m ; after the rotation both columns are $\mathbf{0}$. This is unlike the algorithm proposed [38], which used up to 3 hyperbolic rotations per update, acting on nonzero columns, where each of these rotations could lead to unbounded results. (The numerical stability is further studied in Section 2.5.)

2.4.3 Initialization

Prior to any updates, the algorithm is initialized as follows:

$$d = 0; \quad \mathbf{R} = \mathbf{0}_{m \times m}; \quad \mathbf{Q} = \mathbf{I}_m.$$

If $\mathbf{N} = \gamma\mathbf{I}$ is selected, i.e., it is needed to compute the principal subspace of \mathbf{X} in the sense of (2.3), then the first m updates after this initialization are using the columns of \mathbf{N} , and the result is

$$d = 0; \quad \mathbf{R} = \gamma\mathbf{I}_m; \quad \mathbf{Q} = \mathbf{I}_m.$$

Hence, this is also a valid initialization for the case $\mathbf{N} = \gamma\mathbf{I}$. More generally, it is allowed to do initialization using $\mathbf{R} = \mathbf{R}_{\mathbf{N}}^{1/2}$, where $\mathbf{R}_{\mathbf{N}}^{1/2}$ is a lower triangular Cholesky factor of $\mathbf{R}_{\mathbf{N}} = \mathbf{N}\mathbf{N}^H$, which is a properly scaled noise covariance matrix in most applications.

2.4.4 Computational Complexity

The computational complexity per update vector is assessed as a fixed number of m^2 multiplications (for the initial transformation of \mathbf{n} or \mathbf{x} to \mathbf{c} by \mathbf{Q}), followed by $O(m^2)$ elementary rotations. More precise numbers are given in Table 2.3, where applying a 2×2 rotation to a 2×1 vector has been counted as 4 complex multiplications. The complexity depends on the sign $j_{\mathbf{c}}$ (update or downdate), the rank d , and potential rank changes. The worst case is attained for rank $d = m$, at a complexity of $O(21m^2)$ multiplications per update vector.

In the proposed algorithm, only the square matrix \mathbf{Q} : $m \times m$ and the lower triangular matrix \mathbf{R} : $m \times m$ need to be stored.

Table 2.3: Computational complexity (complex multiplications; only dominant terms shown)

| | $j_{\mathbf{c}} = +1$ case | $j_{\mathbf{c}} = -1$ case |
|-----------------------------|---|--|
| $\mathbf{Q}^H \mathbf{n}$: | m^2 | $\mathbf{Q}^H \mathbf{x}$: m^2 |
| (a): | $4m(m-d)$ | (a): $12m(m-d)$ |
| (b): | $8md + 4d^2$ | (b): $8md + 4d^2$ |
| (e): | $8md$ | (e): $8md$ |
| Total: | $5m^2 + 12md + 4d^2$ multiplications | Total: $13m^2 + 4md + 4d^2$ multiplications |

2.5 Updating the SSE-2

2.5.1 SSE-2 Definition and Properties

As mentioned at the end of section 2.3, the proposed HURV decomposition (2.20) is not unique, and hence, it is necessary to place additional constraints to reach desired properties. This is already the case for the original one-sided decomposition (2.15). At the same time, all valid subspace estimates have such a form as

$$\mathbf{U} = \mathbf{B} - \mathbf{A}\mathbf{M}, \quad \|\mathbf{M}\| \leq 1,$$

where $\mathbf{M} : m-d \times d$ is a contractive matrix that parametrizes all solutions. These freedoms are connected: Given a \mathbf{M} , it is always possible to transform (\mathbf{A}, \mathbf{B}) using additional rotations $\Theta_{\mathbf{M}}$ to a new $(\mathbf{A}', \mathbf{B}')$ such that $\text{ran}(\mathbf{B}') = \text{ran}(\mathbf{B} - \mathbf{A}\mathbf{M})$, i.e., the same subspace is obtained using \mathbf{B}' and a new parameter $\mathbf{M}' = \mathbf{0}$. The existence of this transformation is shown in the proof of Theorem 1 below.

For a given decomposition, define a block partitioning of the corresponding Θ as

$$\Theta = \begin{matrix} & \begin{matrix} n_1 & n_2 \end{matrix} \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix} \end{matrix}, \quad (2.22)$$

In [35], a specific choice for \mathbf{M} was taken, namely a function of Θ :

$$\mathbf{M}_{\Theta} := [\mathbf{I}_{m-d} \quad \mathbf{0}] \Theta_{11}^{-1} \Theta_{12} \begin{bmatrix} \mathbf{I}_d \\ \mathbf{0} \end{bmatrix}. \quad (2.23)$$

The corresponding subspace estimate $\mathbf{U}_{SSE-2} = \mathbf{B} - \mathbf{A}\mathbf{M}_{\Theta}$ was called the SSE-2 subspace estimate, and was shown in [35] to enjoy important properties that make it numerically stable and “unbiased” in the sense that $\text{ran}(\mathbf{U}_{SSE-2}) \subset \text{ran}(\mathbf{X})$. Specifically:

Lemma 1. *Given an HURV decomposition (2.20), consider $\mathbf{B}' = \mathbf{B} - \mathbf{A}\mathbf{M}_{\Theta}$, with \mathbf{M}_{Θ} given by (2.23). Then $\text{ran}(\mathbf{B}') \subset \text{ran}(\mathbf{X})$, $\|\mathbf{B}'\| \leq \|\mathbf{X}\|$.*

Proof. The proof is a straightforward generalization of Lemma 3.4 in [35], which derived this for $\mathbf{N} = \epsilon \mathbf{I}$. \square

It was also shown in simulations [35] that the resulting subspace estimates $\text{ran}(\mathbf{B}')$ are quite close to the principal subspace estimate that follows from the SVD, which is in fact a special case of an SSE-2 subspace (see Appendix 2.A). Unfortunately, the direct computation of $\mathbf{B}' = \mathbf{B} - \mathbf{A}\mathbf{M}_\Theta$ with Θ given by (2.23) is quite cumbersome, as it requires the inversion of a submatrix of Θ , which itself grows as we add more updates. Also, it is possible that Θ is unbounded (or very large) even if the related (\mathbf{A}, \mathbf{B}) are well defined.

2.5.2 The SURV Algorithm Provides an SSE-2 Decomposition

In this section, it is shown that the proposed SURV updating algorithm in Section 2.4 automatically leads to a \mathbf{B} that is already the desired SSE-2 subspace estimate. This is achieved by restricting the available freedom in Θ such that \mathbf{M}_Θ in (2.23) satisfies $\mathbf{M}_\Theta = \mathbf{0}$, i.e., the inversion form in (2.23) is avoided, storage of Θ is not needed, and $\mathbf{U}_{SSE-2} = \mathbf{B}$. The results are summarized as follows:³

Theorem 1 ([38]). *For given matrices $\mathbf{N} : m \times n_1$ and $\mathbf{X} : m \times n_2$, there exist matrices $\mathbf{Q} : m \times m$, $\Theta : (n_1 + n_2) \times (n_1 + n_2)$, $\mathbf{T} : n_1 \times n_1$ such that*

$$\mathbf{Q}^H \begin{bmatrix} \overset{n_1}{+} \mathbf{N} & \overset{n_2}{-} \mathbf{X} \end{bmatrix} \Theta = \begin{bmatrix} \overset{m-d}{+} \mathbf{R}_A & \overset{n_1-d}{m+d} \mathbf{0} & \overset{d}{-} \mathbf{R}_B & \overset{n_2-d}{-} \mathbf{0} \end{bmatrix}, \quad (2.24)$$

$$\mathbf{T} \begin{bmatrix} \overset{n_1}{+} \mathbf{I} & \overset{n_2}{-} \mathbf{0} \end{bmatrix} \Theta = \begin{bmatrix} \overset{m-d}{+} \mathbf{I} & \overset{n_1-d}{m+d} \mathbf{0} & \overset{d}{-} \mathbf{0} & \overset{n_2-d}{-} * \\ \overset{m-d}{n_1-m+d} \mathbf{0} & \overset{n_1-d}{m+d} \mathbf{I} & \overset{d}{-} * & \overset{n_2-d}{-} * \end{bmatrix}, \quad (2.25)$$

where $\mathbf{Q} = [\mathbf{Q}_A \ \mathbf{Q}_B]$ is unitary, Θ is \mathbf{J} -unitary, $[\mathbf{R}_A \ \mathbf{R}_B]$ is lower triangular, and \mathbf{T} is an invertible matrix.

Let $[\mathbf{A} \ \mathbf{B}] = \mathbf{Q}[\mathbf{R}_A \ \mathbf{R}_B]$. Then $\text{ran}(\mathbf{Q}_B) = \text{ran}(\mathbf{B})$ is an SSE-2 subspace estimate.

Proof. See Appendix 2.B. \square

Corollary 1. *For the decomposition in Theorem 1, Θ is bounded if $\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H$ is nonsingular. In any case, the following assertions are true:*

1. $\text{ran}(\mathbf{B}) \subset \text{ran}(\mathbf{X})$, $\|\mathbf{R}_B\| = \|\mathbf{B}\| \leq \|\mathbf{X}\|$;
2. $\text{ran}(\mathbf{A}) \subset \text{ran}(\mathbf{N})$, $\|\mathbf{R}_A\| = \|\mathbf{A}\| \leq \|\mathbf{N}\|$.

³The theorem was proposed in [38] without a complete proof.

Proof. The result for Θ follows from Theorem 3.1 in [35], which proved it for the more general HURV. Theorem 1 shows that for the present decomposition (constrained by (2.25)), $\mathbf{M}_\Theta = \mathbf{0}$. Thus, by Lemma 1, the range and norm properties for \mathbf{B} follow. Since $\mathbf{N}\mathbf{N}^H - \mathbf{A}\mathbf{A}^H = \mathbf{X}\mathbf{X}^H - \mathbf{B}\mathbf{B}^H \geq 0$, it follows $\|\mathbf{A}\| \leq \|\mathbf{N}\|$, and hence also $\text{ran}(\mathbf{A}) \subset \text{ran}(\mathbf{N})$ (see [58][p.471]). Finally, $\|\mathbf{R}_\mathbf{A}\| = \|\mathbf{Q}^H \mathbf{A}\| = \|\mathbf{A}\|$ and $\|\mathbf{R}_\mathbf{B}\| = \|\mathbf{Q}^H \mathbf{B}\| = \|\mathbf{B}\|$, so the same norm inequalities hold for $\mathbf{R}_\mathbf{A}$ and $\mathbf{R}_\mathbf{B}$. \square

Thus, it is seen that the results (\mathbf{A}, \mathbf{B}) of the decomposition are bounded by the inputs, even if Θ may be unbounded. Also the corresponding subspaces $\mathbf{Q}_\mathbf{A}, \mathbf{Q}_\mathbf{B}$ are well-defined.

It remains to show:

Theorem 2. *The SURV algorithm presented in Section 2.4 provides the decomposition in Theorem 1 (without explicitly computing or storing Θ, \mathbf{T} and the right hand side of (2.25)).*

Proof. See Appendix 2.C. \square

2.5.3 Numerical Stability

Generally, algorithms using hyperbolic rotations have an issue of numerical stability: such rotations may have a large norm. Some relevant literature can be found in [47–50, 59].

Regarding the numerical stability of the proposed SURV algorithm, it should be noted first of all that Corollary 1 states that the corresponding ‘‘SSE-2’’ decomposition in Theorem 1 is well behaved: the resulting outputs \mathbf{A}, \mathbf{B} are bounded by the inputs, and in the algorithm the potentially unbounded Θ is not explicitly computed or used. Furthermore, in the implementation, the algorithm has only one hyperbolic rotation per update vector \mathbf{c} . This hyperbolic rotation θ is computed from two entries $r := r_{m,m}$ and $x := c_m$, and is applied only to these two entries as the corresponding columns of \mathbf{R} and \mathbf{c} to which θ should be applied are already zero except for these two entries. Given the numerically stable forms for hyperbolic rotations in case 1 and case 2 of Table 2.1:

$$\text{Case 1 } (|r| > |x|) : \quad s = \frac{x}{r}; \quad \begin{bmatrix} r & x \end{bmatrix} \theta = \begin{bmatrix} r\sqrt{1-|s|^2} & 0 \end{bmatrix},$$

$$\text{Case 2 } (|r| < |x|) : \quad s = \frac{r}{x}; \quad \begin{bmatrix} r & x \end{bmatrix} \theta = \begin{bmatrix} 0 & x\sqrt{1-|s|^2} \end{bmatrix},$$

it is possible to compute the resulting output vector (or entry r' in $\begin{bmatrix} r & x \end{bmatrix} \theta = \begin{bmatrix} r' & 0 \end{bmatrix}$) directly, avoiding the potential singularity problem arising from the intermediate computation of θ in case $|r| \approx |x|$. Indeed, $|r'| < |r|$, resp. $|r'| < |x|$, which is also seen as a particular case of Corollary 1. All other operations in the SURV

algorithm are ordinary Givens rotations (Case 3 in Table 2.1), which are known to be numerically stable [60]. Thus, it can be concluded that the proposed SURV algorithm is also stable.

If it happens that $|r| = |x|$ (this can occur only if $\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H$ is singular, e.g., for $\mathbf{N} = \gamma\mathbf{I}$ if a singular value of \mathbf{X} is precisely equal to the threshold γ), then the numerically stable forms given above result in $r' = 0$. A decision will have to be made on the corresponding output signature ($\mathbf{j}_2 = \mathbf{j}_1$ or $\mathbf{j}_2 = -\mathbf{j}_1$); this decision is arbitrary and corresponds to a rank decision on \mathbf{X} when a singular value is precisely at the threshold. The numerical properties of the algorithm do not depend on this decision.

2.6 Simulation Results

In this section, the numerical stability of the proposed SURV algorithm is demonstrated by some simulations, and compare its performance with the original HURV algorithm [38] and the URV [37]. For the simulations, a generic subspace tracking data model will be used as follows:

$$\mathbf{x}(t) = \mathbf{H}(t)\mathbf{s}(t) + \mathbf{n}(t), \quad t = 1, 2, \dots, N \quad (2.26)$$

where $\mathbf{x}(t)$: $m \times 1$, $\mathbf{s}(t)$: $d \times 1$ ($d \leq m$) is formed by i.i.d. Gaussian random variables with zero mean and standard deviation $\sigma_s = 1$, and $\mathbf{n}(t)$: $m \times 1$ is formed by i.i.d. Gaussian noise with zero mean and standard deviation σ_n . The mixing matrix $\mathbf{H}(t)$: $m \times d$ has d singular values all set to 1. The complete data matrix is formed by $\mathbf{X}_N = [\mathbf{x}(1), \dots, \mathbf{x}(N)]$, with size $m \times N$.

The subspace tracking uses a sliding window of size n on \mathbf{X}_N to compute the subspace of the corresponding data matrix $\mathbf{X}^{(i)} = [\mathbf{x}(i), \dots, \mathbf{x}(i+n-1)]$, for $i = 1, 2, \dots$. Starting from the decomposition for $\mathbf{X}^{(i-1)}$, this is implemented by applying an update by $\mathbf{x}(i+n-1)$ followed by a downdate by $\mathbf{x}(i-1)$ (i.e., an update by $\mathbf{x}(i-1)$ with an opposite signature). The SURV and HURV are initialized by setting $\mathbf{R} = \gamma\mathbf{I}_m$, where γ is the selected threshold (see below), followed by updates by $\mathbf{x}(1), \dots, \mathbf{x}(n)$.

For reference, each $\mathbf{X}^{(i)}$ is also processed by an SVD to generate the ‘‘optimal’’ rank estimate \hat{d}_{svd} corresponding to the threshold γ , and the principal subspace $\hat{\mathbf{U}}_{\text{svd}}$ formed by the singular vectors corresponding to the \hat{d}_{svd} largest singular values. These are called the SVD estimates. Rank errors in the SURV occur when estimates $\hat{d} \neq \hat{d}_{\text{svd}}$ (compared with the SVD) or $\hat{d} \neq d$ (the true rank), depending on the criterion.

The ‘‘rank error rate’’ (RER) is defined by

$$\text{RER} = \frac{\text{Number of rank errors}}{\text{Total number of tests}}. \quad (2.27)$$

The error $e_{\hat{\mathbf{U}}}$ of a principal subspace estimate $\hat{\mathbf{U}}$ (where $\hat{\mathbf{U}}$ is a matrix whose columns form an orthonormal basis of the estimated subspace) is defined by the

subspace angle as follows,

$$\text{If } \hat{d} = d, \text{ then } e_{\hat{\mathbf{U}}} = \|\mathbf{P}_{\hat{\mathbf{U}}}^\perp \mathbf{U}_{\mathbf{H}}\|; \text{ else } e_{\hat{\mathbf{U}}} = 1, \quad (2.28)$$

where $\mathbf{P}_{\hat{\mathbf{U}}}^\perp = \mathbf{I} - \hat{\mathbf{U}}\hat{\mathbf{U}}^H$, and columns of $\mathbf{U}_{\mathbf{H}}$ are the orthonormal basis of $\text{ran}(\mathbf{H})$.

For SURV and HURV, the ‘‘factorization error’’ is defined for each i as

$$e_{\mathbf{f}}^{(i)} = \|(\gamma^2 \mathbf{I} - \mathbf{X}^{(i)} \mathbf{X}^{(i)H}) - \mathbf{Q} \mathbf{R} \mathbf{J} \mathbf{R}^H \mathbf{Q}^H\|, \quad (2.29)$$

where \mathbf{Q} , \mathbf{R} are defined as in (2.20). For URV, which computes a decomposition $\mathbf{X}^{(i)} = \mathbf{U} \mathbf{R} \mathbf{V}^H$, the ‘‘factorization error’’ is defined as

$$e_{\mathbf{f}}^{(i)} = \|\mathbf{X}^{(i)} \mathbf{X}^{(i)H} - \mathbf{U} \mathbf{R} \mathbf{R}^H \mathbf{U}^H\|. \quad (2.30)$$

For the SVD, when computes a decomposition $\mathbf{X}^{(i)} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^H$, the ‘‘factorization error’’ is defined as

$$e_{\mathbf{f}}^{(i)} = \|\mathbf{X}^{(i)} \mathbf{X}^{(i)H} - \mathbf{U} \mathbf{\Sigma}^2 \mathbf{U}^H\|. \quad (2.31)$$

The signal to noise ratio (SNR) is defined as $\text{SNR} = 10 \log(\sigma_s^2 / \sigma_n^2)$. The same rank threshold

$$\gamma = 1.24\alpha \quad \text{where} \quad \alpha = \sigma_n(1 + \sqrt{m/n})\sqrt{n}$$

is applied to all algorithms. In fact, α is an estimate of the largest singular value of a ‘‘noise only’’ data matrix $\mathbf{N} : m \times n$ [61–63], and γ is a scaled version that usually results in correct rank estimates (no ‘‘false alarm’’).

In the simulations, the choices of m and n are $m = 16$ and $n = 20$ unless specified otherwise. ‘‘HURV’’ refers to the previously proposed algorithm in [38] for computing the same decomposition (Theorem 1) as the SURV. The ‘‘URV’’ algorithm is based on the MATLAB code taken from UTV-tools [37], which has been slightly modified so that it can run in nonstationary cases.

Fig. 2.4 shows the boxplot of the factorization error of SURV, HURV and URV on random matrices of size 16×20 over 10^5 Monte Carlo runs at given SNRs. The factorization error of the SVD is used as the reference. It is seen from Fig. 2.4 that HURV sometimes is not as stable as SURV and URV. The reason for this is that HURV uses at most three hyperbolic rotations per vector update (and also tracks two entries of Θ), and a situation that two large hyperbolic rotations cancel each other might happen. However, SURV uses at most one hyperbolic rotation per vector update and produces a stable factorization.

Fig. 2.5 shows the factorization error of SURV, HURV and URV for tracking in stationary and nonstationary cases (single run), where $N \approx 10^5$. In Fig. 2.5(a), $d = 2$. In Fig. 2.5(b), d switches between 2 and 4 every 150 samples. In Fig. 2.5(c), $m = n = 16$ and d switches between 8 and 16 every 150 samples at very high SNR (SNR = 250dB). It is seen that URV encounters breakdowns frequently and HURV encounters breakdowns at high SNR while such breakdowns never occur for SURV.

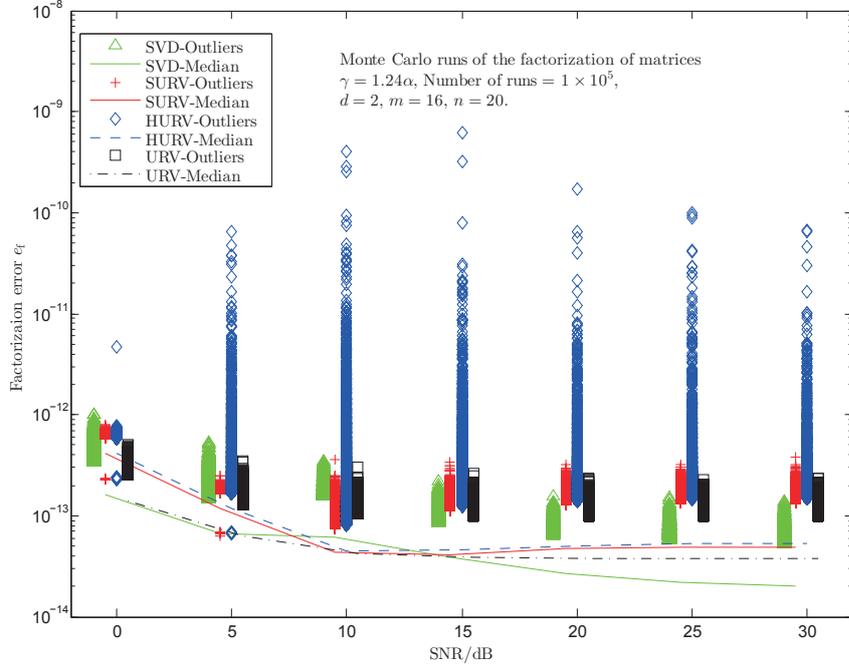


Figure 2.4: Factorization error of SURV, HURV and URV on random matrices.

Fig. 2.6 shows the subspace tracking performance of SURV, HURV and URV in a stationary case as a function of SNR. Fig. 2.6(a), Fig. 2.6(b) and Fig. 2.6(c) show the rank error rate in comparison with the real number of signals d , the rank error rate in comparison with \hat{d}_{SVD} , and the averaged subspace error of the estimated principal subspace over $2 \cdot 10^4$ updates per run ($N \approx 10^4$) and 10 runs, respectively. It is seen that the rank estimate of URV is unreliable at low SNR, while SURV and HURV always give rank estimates consistent with the SVD. This implies that URV cannot track the rank of the subspace well in nonstationary cases. The subspace estimates computed from SURV and HURV stay closer to $\text{ran}(\mathbf{H})$ than the others.

Fig. 2.7 shows the subspace tracking performance of SURV, HURV and URV in a nonstationary case as a function of SNR, with the true rank d switching between 2 and 4 sources every 150 samples. Fig. 2.7 is similar to Fig. 2.6 except that in Fig. 2.7(a) and Fig. 2.7(c) the performance statistics are only for the stationary parts, omitting the size- n transient parts during rank changes. It is seen from Fig. 2.7 that SURV and HURV always give rank estimates consistent with the SVD, as well as good estimates of the true principal (signal) subspace $\text{ran}(\mathbf{H})$. However, URV gives many rank errors and encounters breakdowns even at high SNR.

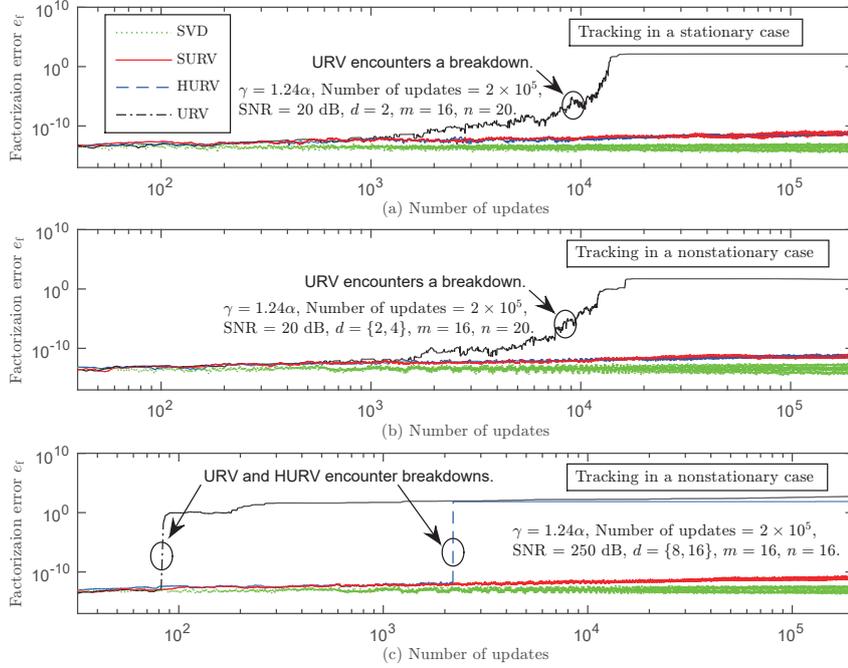


Figure 2.5: Factorization error of SURV, HURV and URV for tracking in stationary and nonstationary cases (single run).

In the following, a typical case will be provided where the subspace estimates computed from the compared algorithms are applied for direction of arrival (DOA) estimation where a classic algorithm, ESPRIT, is used (relevant results can be found in [54]). A linear antenna array of $m = 8$ elements with separation of 0.5 wavelength is used. The channel matrix $\mathbf{H}(t, \theta)$ now consists of steering vectors, which are complex vectors nonorthogonal to each other. A nonstationary scenario is chosen, where d switches between 2 (two sources with DOA $\boldsymbol{\theta} = [-20^\circ, 20^\circ]^T$) and 4 (four sources with DOA $\boldsymbol{\theta} = [40^\circ, -50^\circ, 0^\circ, 70^\circ]^T$) every 150 samples. The source signal is BPSK modulated (real-valued symbols from the alphabet $\{+1, -1\}$). All sources have equal signal power. Due to the limitation that URV codes from [37] runs only on real-valued data matrices, it is needed to split the real and imaginary parts of the complex matrix $\mathbf{X}^{(i)}$ and to stack them up to form a new real-valued data matrix

$$\mathbf{X}_{stack}^{(i)} = \begin{bmatrix} \text{Re}(\mathbf{X}^{(i)}) \\ \text{Im}(\mathbf{X}^{(i)}) \end{bmatrix}. \quad (2.32)$$

The subspace estimate $\hat{\mathbf{U}} = \begin{bmatrix} \hat{\mathbf{U}}_1 \\ \hat{\mathbf{U}}_2 \end{bmatrix}$ from each of the tracking algorithms is split

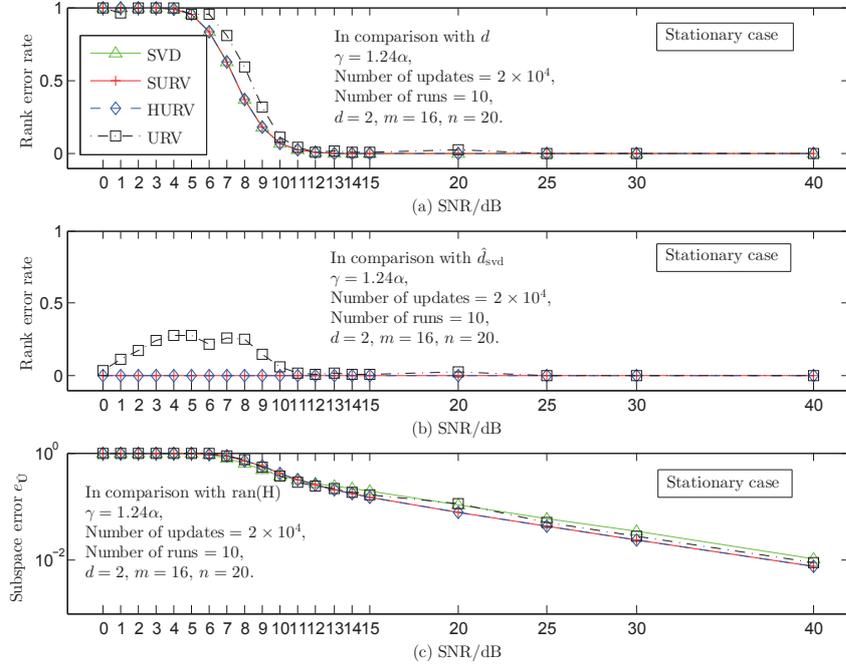


Figure 2.6: Subspace tracking performance of SURV, HURV and URV in a stationary case as a function of SNR.

into two submatrices to form the complex matrix

$$\hat{\mathbf{U}}_{\text{unstack}} = \hat{\mathbf{U}}_1 + j\hat{\mathbf{U}}_2, \quad (2.33)$$

which is passed to ESPRIT. It is better to select $2m \leq n$ due to the problem that URV [37] refuses to run when $\mathbf{X}_{\text{stack}}^{(i)}$ is tall. In contrast, SURV and HURV do not have this limitation and still work well on any tall matrices. For fair comparison, SURV, HURV, and the SVD are applied to the same $\mathbf{X}_{\text{stack}}^{(i)}$. The performance measure of DOA estimation is the root-mean-square error (RMSE)

$$\text{RMSE} = \sqrt{\text{avg}\left(\sum_{k=1}^{10} \sum_i \min_{\mathbf{\Pi}_1 \text{ or } \mathbf{\Pi}_2} \text{avg} \left\| \mathbf{\Pi}_1 \hat{\boldsymbol{\theta}}_{ik} - \mathbf{\Pi}_2 \boldsymbol{\theta}_{ik} \right\|_E^2\right)}, \quad (2.34)$$

where $\|\cdot\|_E$ denotes the Euclidean norm, $\hat{\boldsymbol{\theta}}_{ik}$ and $\boldsymbol{\theta}_{ik}$ are the DOA estimates and the true DOA for $\mathbf{X}^{(i)}$ in the k -th run, respectively, the length of $\hat{\boldsymbol{\theta}}_{ik}$ is limited to at most 4 (as the length of $\boldsymbol{\theta}_{ik}$ is at most 4), and $\mathbf{\Pi}_1$ and $\mathbf{\Pi}_2$ are row selection matrices with compatible size to $\hat{\boldsymbol{\theta}}_{ik}$ and $\boldsymbol{\theta}_{ik}$ to generate all possible row selections. $\mathbf{\Pi}_1 = \mathbf{I}$, when the length of $\hat{\boldsymbol{\theta}}_{ik}$ is shorter than $\boldsymbol{\theta}_{ik}$; $\mathbf{\Pi}_2 = \mathbf{I}$, otherwise.

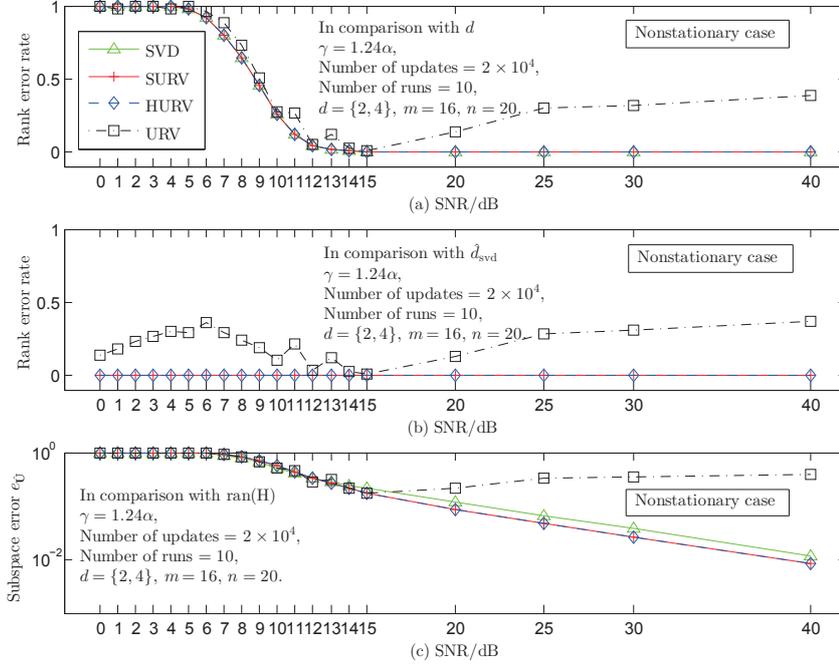


Figure 2.7: Subspace tracking performance of SURV, HURV and URV in a nonstationary case as a function of SNR.

Fig. 2.8 shows the performance of DOA estimation using subspace estimates computed from SURV, HURV and URV tracking in the above nonstationary case. The performance of the SVD is used as the reference. The rank threshold is set to $\gamma = 1.28\alpha$ due to the change of m . Fig. 2.8(a) and Fig. 2.8(c) provide the collection of the performance statistics only for the stationary parts like that in Fig. 2.7. SURV and HURV again show very good tracking performance that the rank estimates are exact and the subspace estimates are good and useful. The averaged DOA performance of SURV and HURV stays quite close to and even better than the SVD. URV shows unstable performance, resulting in bad DOA estimates, as it frequently encounters breakdowns during tracking, where the rank can be overestimated.

2.7 Conclusions

This chapter proposed a Hyperbolic URV decomposition and its updating algorithm (SURV) for subspace tracking. It is a non-iterative algorithm that provides exact rank estimates and very good principal subspace estimates in comparison with the SVD with the same rank threshold. The subspace is provided in terms

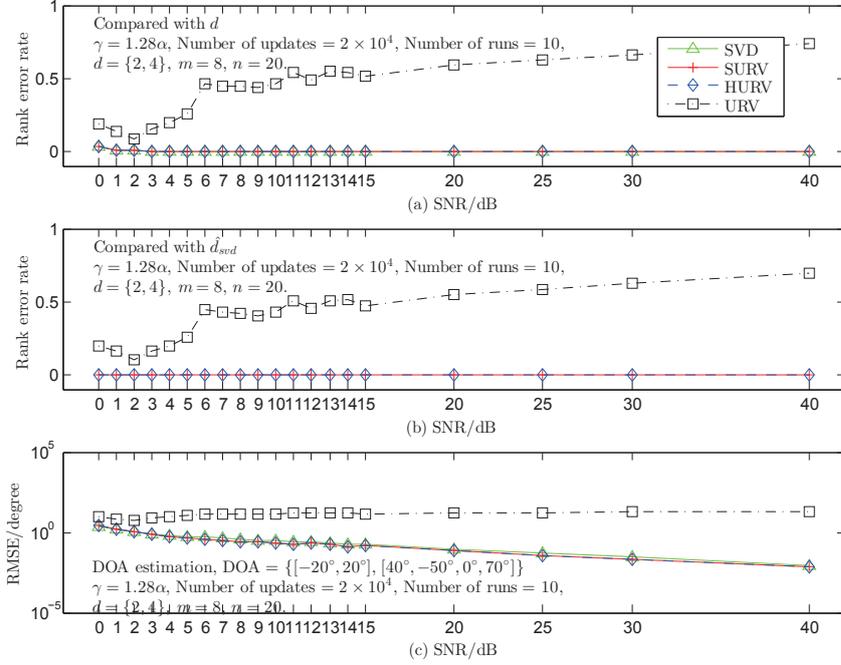


Figure 2.8: Performance of DOA estimation using subspace estimates computed from SURV, HURV and URV tracking in a nonstationary case as a function of SNR.

of an orthogonal basis \mathbf{Q}_B . The proposed updating algorithm uses at most one hyperbolic rotation per vector update and provides a numerically stable computation of the “SSE-2” Schur subspace estimator (unlike the previously proposed algorithm in [38]).

The proposed updating algorithm has a computational complexity of $O(m^2)$ per size- m update vector (similar to a QR update), and has constant $O(m^2)$ memory requirements. The algorithm is based on a sliding-window update scheme, but is easily tailored to exponential windowing by scaling matrix \mathbf{R} after every update. All operations in the proposed updating algorithm are local, consisting of elementary rotations and column permutations, and the computational flow is non-iterative, thus facilitating parallel implementations.

The proposed algorithm can replace an SVD in cases where the subspace is the only point of interest, and a threshold on the noise power for the splitting the subspace is given. The computational complexity is higher than some of the tracking algorithms with knowledge of the rank, such as PAST, but the advantage is that the estimate is “exact” with known properties and does not rely on convergence.

2.A The Principal Subspace is an SSE-2

Here it will be shown that for a given matrix \mathbf{X} , there is an SSE-2 decomposition of the required form (2.24)–(2.25) such that the “SSE-2 subspace” provided by \mathbf{Q}_B is equal to the left principal subspace of \mathbf{X} as provided by the SVD.

Suppose for simplicity of notation that \mathbf{X} is square, and has d singular values larger than d . Let the SVD of \mathbf{X} be defined by $\mathbf{X} = \mathbf{U}_1 \boldsymbol{\Sigma}_1 \mathbf{V}_1^H + \mathbf{U}_2 \boldsymbol{\Sigma}_2 \mathbf{V}_2^H$, where \mathbf{U}_1 and \mathbf{V}_1 have d columns, \mathbf{U}_2 and \mathbf{V}_2 have $m - d$ columns, $\boldsymbol{\Sigma}_1 : d \times d$ and $\boldsymbol{\Sigma}_2 : (m - d) \times (m - d)$ are diagonal matrices, and $\boldsymbol{\Sigma}_1 > \gamma$ and $\boldsymbol{\Sigma}_2 < \gamma$. Define $\mathbf{Q}, \boldsymbol{\Theta}, \mathbf{T}$ as

$$\begin{aligned} \mathbf{Q} &= [\mathbf{U}_2 \quad \mathbf{U}_1], \\ \boldsymbol{\Theta} &= \left[\begin{array}{cc|cc} \mathbf{U}_2 \boldsymbol{\Sigma}_2 & \mathbf{U}_1 \boldsymbol{\Sigma}_1 & -\mathbf{U}_1 \boldsymbol{\Sigma}_1 & -\mathbf{U}_2 \boldsymbol{\Sigma}_2 \\ -\mathbf{V}_2 \boldsymbol{\Sigma}_2 & -\mathbf{V}_1 \boldsymbol{\Sigma}_1 & \mathbf{V}_1 \boldsymbol{\Sigma}_1 & \mathbf{V}_2 \boldsymbol{\Sigma}_2 \end{array} \right]^{-1/2} \\ &\quad \left[\begin{array}{ccc} \gamma^2 - \boldsymbol{\Sigma}_2^2 & & \\ & \boldsymbol{\Sigma}_1^2 - \gamma^2 & \\ & & \boldsymbol{\Sigma}_1^2 - \gamma^2 \\ & & & \gamma^2 - \boldsymbol{\Sigma}_2^2 \end{array} \right], \\ \mathbf{T} &= \boldsymbol{\Theta}_{11}^{-1} = \left[\begin{array}{c} (\gamma^2 - \boldsymbol{\Sigma}_2^2)^{1/2} \gamma^{-1} \mathbf{U}_2^H \\ (\boldsymbol{\Sigma}_1^2 - \gamma^2)^{1/2} \boldsymbol{\Sigma}_1^{-1} \mathbf{U}_1^H \end{array} \right]. \end{aligned}$$

It is readily verified that \mathbf{Q} is unitary, $\boldsymbol{\Theta}$ is \mathbf{J} -unitary, and that

$$\begin{aligned} \mathbf{Q}^H [\gamma \mathbf{I} \quad \mathbf{X}] \boldsymbol{\Theta} &= \left[\begin{array}{cc|cc} (\gamma^2 - \boldsymbol{\Sigma}_2^2)^{1/2} & 0 & 0 & 0 \\ 0 & 0 & (\boldsymbol{\Sigma}_1^2 - \gamma^2)^{1/2} & 0 \end{array} \right], \\ \mathbf{T} [\mathbf{I} \quad \mathbf{0}] \boldsymbol{\Theta} &= \left[\begin{array}{cc|cc} \mathbf{I} & \mathbf{0} & \mathbf{0} & -\gamma^{-1} \boldsymbol{\Sigma}_2 \\ \mathbf{0} & \mathbf{I} & -\boldsymbol{\Sigma}_1^{-1} & \mathbf{0} \end{array} \right], \end{aligned}$$

so that (2.24)–(2.25) hold. Since $\mathbf{Q}_B = \mathbf{U}_1$, the SSE-2 subspace is equal to the principal subspace obtained by an SVD.

2.B Proof for Theorem 1 in Section 2.5

The proof is technical and consists of several steps (lemmas).

Lemma 2. *For given matrices $\mathbf{A} : m \times (m - d)$, $\mathbf{B} : m \times d$, and a \mathbf{J} -unitary matrix $\boldsymbol{\Theta} : (n_1 + n_2) \times (n_1 + n_2)$, consider a transformation by any \mathbf{J} -unitary matrix $\boldsymbol{\Theta}_M$:*

$$[\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \boldsymbol{\Theta}_M = [\mathbf{A}' \quad \mathbf{0} \mid \mathbf{B}' \quad \mathbf{0}], \quad (2.35)$$

$$\boldsymbol{\Theta} \boldsymbol{\Theta}_M = \boldsymbol{\Theta}', \quad (2.36)$$

where $\boldsymbol{\Theta}_M$ only acts on the columns of \mathbf{A}, \mathbf{B} (and corresponding columns of $\boldsymbol{\Theta}$). Define \mathbf{M}_Θ as in (2.23), and likewise for $\mathbf{M}_{\Theta'}$.

Then $\text{ran}(\mathbf{B} - \mathbf{A} \mathbf{M}_\Theta) = \text{ran}(\mathbf{B}' - \mathbf{A}' \mathbf{M}_{\Theta'})$, i.e., the SSE-2 subspace is invariant under $\boldsymbol{\Theta}_M$.

Proof. Use the block partitioning (2.22) of Θ , and note that

$$[\Theta_{11} \ \Theta_{12}] \begin{bmatrix} -\Theta_{11}^{-1}\Theta_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} = \mathbf{0}, \quad (2.37)$$

$$[\Theta'_{11} \ \Theta'_{12}] \begin{bmatrix} -\Theta'^{-1}_{11}\Theta'_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} = \mathbf{0}, \quad (2.38)$$

Also note that the right hand side matrix factors are full rank complements with equal column dimensions n_2 . From $\Theta\Theta_M = \Theta'$ it is obtained that

$$[\Theta_{11} \ \Theta_{12}] \Theta_M = [\Theta'_{11} \ \Theta'_{12}]. \quad (2.39)$$

Inserting in (2.38) gives

$$[\Theta_{11} \ \Theta_{12}] \Theta_M \begin{bmatrix} -\Theta'^{-1}_{11}\Theta'_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} = \mathbf{0}. \quad (2.40)$$

Comparing (2.40) with (2.37) and using the full rank complement property reveals that there exists a $n_2 \times n_2$ invertible matrix \mathbf{V} such that

$$\begin{aligned} \Theta_M \begin{bmatrix} -\Theta'^{-1}_{11}\Theta'_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} &= \begin{bmatrix} -\Theta_{11}^{-1}\Theta_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} \mathbf{V} \\ &= \begin{bmatrix} -\Theta_{11}^{-1}\Theta_{12}\mathbf{V} \\ \mathbf{V} \end{bmatrix}. \end{aligned} \quad (2.41)$$

It is next shown that \mathbf{V} is a block-upper triangular matrix. Because Θ_M only acts on the columns of \mathbf{A} , \mathbf{B} , the block structure of Θ_M can be written as

$$\begin{aligned} \Theta_M &= \begin{matrix} & n_1 & n_2 \\ \begin{matrix} n_1 \\ n_2 \end{matrix} & \begin{bmatrix} (\Theta_M)_{11} & (\Theta_M)_{12} \\ (\Theta_M)_{21} & (\Theta_M)_{22} \end{bmatrix} \end{matrix} \\ &= \begin{matrix} & m-d & \begin{matrix} n_1-d \\ (m-d) \end{matrix} & d & n_2-d \\ \begin{matrix} m-d \\ n_1-(m-d) \\ d \\ n_2-d \end{matrix} & \left[\begin{array}{c|c} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} & \begin{bmatrix} \mathbf{P}_{12} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ \hline \begin{bmatrix} \mathbf{P}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} & \begin{bmatrix} \mathbf{P}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \end{array} \right] \end{matrix}, \end{matrix} \quad (2.42)$$

where the identity submatrices in Θ_M ensure that the columns outside \mathbf{A} , \mathbf{B} remain unchanged, and the submatrices \mathbf{P}_{ij} , $i, j = 1, 2$ transform the columns of \mathbf{A} , \mathbf{B} . Now, by using the expression for \mathbf{V} in the lower part of (2.41), and the partitioning of Θ_M , it is found that

$$\mathbf{V} = (\Theta_M)_{22} - (\Theta_M)_{21}\Theta'^{-1}_{11}\Theta'_{12} \quad (2.43)$$

$$\begin{aligned} &= \begin{bmatrix} \mathbf{P}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_2-d} \end{bmatrix} - \begin{bmatrix} \mathbf{P}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} (\Theta'^{-1}_{11}\Theta'_{12})_{11} & (\Theta'^{-1}_{11}\Theta'_{12})_{12} \\ (\Theta'^{-1}_{11}\Theta'_{12})_{21} & (\Theta'^{-1}_{11}\Theta'_{12})_{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{0} & \mathbf{V}_{22} \end{bmatrix}. \end{aligned} \quad (2.44)$$

As \mathbf{V} is invertible, this block structure also implies that \mathbf{V}_{11} is invertible. Furthermore, it follows that

$$\begin{aligned} \mathbf{B}' - \mathbf{A}'\mathbf{M}_{\Theta'} &= [\mathbf{B}' \quad \mathbf{0}] \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} - [\mathbf{A}' \quad \mathbf{0}]\Theta_{11}'^{-1}\Theta_{12}' \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \\ &= [\mathbf{A}' \quad \mathbf{0} \mid \mathbf{B}' \quad \mathbf{0}] \begin{bmatrix} -\Theta_{11}'^{-1}\Theta_{12}' \\ \mathbf{I}_{n_2} \end{bmatrix} \end{aligned} \quad (2.45)$$

$$\stackrel{(2.35)}{=} [\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \Theta_{\mathbf{M}} \begin{bmatrix} -\Theta_{11}'^{-1}\Theta_{12}' \\ \mathbf{I}_{n_2} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_d \\ \mathbf{0} \end{bmatrix} \quad (2.46)$$

$$\stackrel{(2.41)}{=} [\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \begin{bmatrix} -\Theta_{11}^{-1}\Theta_{12} \\ \mathbf{I}_{n_2} \end{bmatrix} \mathbf{V} \begin{bmatrix} \mathbf{I}_d \\ \mathbf{0} \end{bmatrix} \quad (2.47)$$

$$\stackrel{(2.44)}{=} ([\mathbf{B} \quad \mathbf{0}] - [\mathbf{A} \quad \mathbf{0}]\Theta_{11}^{-1}\Theta_{12}) \cdot \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{0} \end{bmatrix}. \quad (2.47)$$

Now, $\Theta_{11}^{-1}\Theta_{12}$ is partitioned into blocks as

$$\Theta_{11}^{-1}\Theta_{12} = \begin{matrix} & d & n_2-d \\ \begin{matrix} m-d \\ n_1-(m-d) \end{matrix} & \begin{bmatrix} (\Theta_{11}^{-1}\Theta_{12})_{11} & (\Theta_{11}^{-1}\Theta_{12})_{12} \\ (\Theta_{11}^{-1}\Theta_{12})_{21} & (\Theta_{11}^{-1}\Theta_{12})_{22} \end{bmatrix} \end{matrix}. \quad (2.48)$$

By using this partitioning, it is obtained that

$$\begin{aligned} \mathbf{B}' - \mathbf{A}'\mathbf{M}_{\Theta'} &= ([\mathbf{B} \quad \mathbf{0}] - [\mathbf{A} \quad \mathbf{0}]\Theta_{11}^{-1}\Theta_{12}) \cdot \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{0} \end{bmatrix} \\ &= [\mathbf{B} - \mathbf{A}(\Theta_{11}^{-1}\Theta_{12})_{11} \quad -\mathbf{A}(\Theta_{11}^{-1}\Theta_{12})_{12}] \cdot \begin{bmatrix} \mathbf{V}_{11} \\ \mathbf{0} \end{bmatrix} \\ &= (\mathbf{B} - \mathbf{A}\mathbf{M}_{\Theta})\mathbf{V}_{11}. \end{aligned} \quad (2.49)$$

Because \mathbf{V}_{11} is invertible, the desired conclusion is reached as:

$$\text{ran}(\mathbf{B}' - \mathbf{A}'\mathbf{M}_{\Theta'}) = \text{ran}(\mathbf{B} - \mathbf{A}\mathbf{M}_{\Theta}). \quad (2.50)$$

□

Lemma 3. *In the context of Lemma 2, there exists a transformation $\Theta_{\mathbf{M}}$ such that $\mathbf{M}_{\Theta'} = \mathbf{0}$, i.e., such that $\text{ran}(\mathbf{B}')$ is an SSE-2 subspace.*

Proof. First explicitly construct a suitable $\Theta_{\mathbf{M}}$ from \mathbf{M}_{Θ} . Since $\|\mathbf{M}_{\Theta}\| = \|(\Theta_{11}^{-1}\Theta_{12})_{11}\| \leq \| \Theta_{11}^{-1}\Theta_{12} \| < 1$ [35], \mathbf{M}_{Θ} is strictly contractive. It follows from Theorem 2.1 in [35] that there exists a \mathbf{J} -unitary matrix \mathbf{P} , with

$$\mathbf{J} = \begin{bmatrix} \mathbf{I}_{m-d} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_d \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix},$$

and an invertible matrix $\mathbf{Y} : (m-d) \times (m-d)$ such that

$$\begin{bmatrix} \mathbf{I}_{m-d} & \mathbf{M}_{\Theta} \end{bmatrix} \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & \mathbf{0}_{(m-d) \times d} \end{bmatrix}. \quad (2.51)$$

Equation (2.51) implies

$$\mathbf{P}_{12} + \mathbf{M}_{\Theta} \mathbf{P}_{22} = \mathbf{P}_{12} + (\Theta_{11}^{-1} \Theta_{12})_{11} \mathbf{P}_{22} = \mathbf{0}. \quad (2.52)$$

A suitable $\Theta_{\mathbf{M}}$ related to \mathbf{M}_{Θ} is now obtained by augmenting \mathbf{P} to a full-size \mathbf{J} -unitary matrix $\Theta_{\mathbf{M}}$ as in (2.42). After defining $\Theta' = \Theta \Theta_{\mathbf{M}}$, it is needed to prove the assertion that $\mathbf{M}_{\Theta'} = \mathbf{0}$. By using the block partitioning of Θ as in (2.22), it follows that

$$\begin{bmatrix} \Theta'_{11} & \Theta'_{12} \end{bmatrix} = \begin{bmatrix} \Theta_{11} & \Theta_{12} \end{bmatrix} \Theta_{\mathbf{M}} \quad (2.53)$$

and hence

$$\Theta_{11}^{-1} \begin{bmatrix} \Theta'_{11} & \Theta'_{12} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \Theta_{11}^{-1} \Theta_{12} \end{bmatrix} \Theta_{\mathbf{M}} \quad (2.54)$$

After completion of the matrix multiplication and insertion of (2.52), it follows that

$$\begin{aligned} \Theta_{11}^{-1} \Theta'_{11} &= \begin{bmatrix} \mathbf{P}_{11} & \\ & \mathbf{I} \end{bmatrix} + \Theta_{11}^{-1} \Theta_{12} \begin{bmatrix} \mathbf{P}_{21} & \\ & \mathbf{I} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{P}_{11} + (\Theta_{11}^{-1} \Theta_{12})_{11} \mathbf{P}_{21} & \mathbf{0} \\ (\Theta_{11}^{-1} \Theta_{12})_{21} \mathbf{P}_{21} & \mathbf{I} \end{bmatrix}, \end{aligned} \quad (2.55)$$

$$\begin{aligned} \Theta_{11}^{-1} \Theta'_{12} &= \begin{bmatrix} \mathbf{P}_{12} & \\ & \mathbf{I} \end{bmatrix} + \Theta_{11}^{-1} \Theta_{12} \begin{bmatrix} \mathbf{P}_{22} & \\ & \mathbf{I} \end{bmatrix} \\ &\stackrel{(2.52)}{=} \begin{bmatrix} \mathbf{0} & (\Theta_{11}^{-1} \Theta_{12})_{12} \\ (\Theta_{11}^{-1} \Theta_{12})_{21} \mathbf{P}_{22} & (\Theta_{11}^{-1} \Theta_{12})_{22} \end{bmatrix}. \end{aligned} \quad (2.56)$$

The block triangular structure of these matrices directly implies that

$$\begin{aligned} \mathbf{M}_{\Theta'} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \Theta_{11}^{-1} \Theta'_{12} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} (\Theta_{11}^{-1} \Theta'_{11})^{-1} (\Theta_{11}^{-1} \Theta'_{12}) \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \\ &\stackrel{(2.55)(2.56)}{=} \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} * & \mathbf{0} \\ * & * \end{bmatrix} \begin{bmatrix} \mathbf{0} & * \\ * & * \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} = \mathbf{0}, \end{aligned} \quad (2.57)$$

and hence

$$\text{ran}(\mathbf{B}' - \mathbf{A}' \mathbf{M}_{\Theta'}) = \text{ran}(\mathbf{B}').$$

□

Hence, it has been confirmed that there exists a matrix $\Theta_{\mathbf{M}}$ (defined explicitly from \mathbf{M}_{Θ} via (2.51)) which transforms any Θ to $\Theta' = \Theta\Theta_{\mathbf{M}}$, such that after the transformation, $\mathbf{M}_{\Theta'} = \mathbf{0}$, implying that the corresponding \mathbf{B}' can simply be taken to get the desired SSE-2 subspace. With the above conclusion, there are easier ways to find this transformation. Suppose Θ is partitioned into

$$\Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}$$

$$= \begin{array}{c} m-d \\ n_1-(m-d) \\ d \\ n_2-d \end{array} \left[\begin{array}{cc|cc} m-d & n_1-(m-d) & d & n_2-d \\ (\Theta_{11})_{11} & (\Theta_{11})_{12} & (\Theta_{12})_{11} & (\Theta_{12})_{12} \\ * & * & * & * \\ \hline * & * & * & * \\ * & * & * & * \end{array} \right]. \quad (2.58)$$

To have $\mathbf{M}_{\Theta'} = \mathbf{0}$, it is only needed to find a transformation on Θ such that the resulting $\Theta'_{11}{}^{-1}\Theta'_{12}$ has a zero (11)-block. This will be the case, for example, if both $(\Theta'_{11})_{12} = \mathbf{0}$ and $(\Theta'_{12})_{11} = \mathbf{0}$. Note that, by definition, $\Theta_{\mathbf{M}}$ is not allowed to change the columns of $(\Theta_{11})_{12}$, for else the range of $\mathbf{B} - \mathbf{A}\mathbf{M}_{\Theta}$ will not stay invariant. Thus, it is inappropriate to use Θ to zero this block. However, it is possible to apply any invertible transformation $\mathbf{T} : n_1 \times n_1$ to the *rows* of $[\Theta_{11} \ \Theta_{12}]$:

$$[\Theta'_{11} \ \Theta'_{12}] = \mathbf{T}[\Theta_{11} \ \Theta_{12}],$$

because $\Theta'_{11}{}^{-1}\Theta'_{12} = \Theta_{11}^{-1}\Theta_{12}$ is invariant under \mathbf{T} . This motivates the specified decomposition (2.24), (2.25) in Theorem 1, whose existence is proven below:

Lemma 4. *Given matrices $\mathbf{A} : m \times (m-d)$, $\mathbf{B} : m \times d$, and a \mathbf{J} -unitary matrix $\Theta : (n_1 + n_2) \times (n_1 + n_2)$. Then there exists a \mathbf{J} -unitary matrix $\Theta_{\mathbf{M}}$ (only acting on the columns of \mathbf{A}, \mathbf{B}) and an invertible matrix \mathbf{T} such that*

$$[\mathbf{A} \ \mathbf{0} \mid \mathbf{B} \ \mathbf{0}]\Theta_{\mathbf{M}} = [\mathbf{A}' \ \mathbf{0} \mid \mathbf{B}' \ \mathbf{0}] \quad (2.59)$$

$$\Theta\Theta_{\mathbf{M}} = \Theta' \quad (2.60)$$

$$\mathbf{T}[\Theta'_{11} \ \Theta'_{12}] = \begin{array}{c} m-d \\ n_1-m+d \end{array} \left[\begin{array}{cc|cc} m-d & n_1-m+d & d & n_2-d \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & * \\ \mathbf{0} & \mathbf{I} & * & * \end{array} \right], \quad (2.61)$$

hence such that $\mathbf{M}_{\Theta'} = \mathbf{0}$.

Proof. The proof is by construction. Given Θ , first form $\mathbf{M}_{\Theta} = [\mathbf{I} \ \mathbf{0}]\Theta_{11}^{-1}\Theta_{12} \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}$, and then take $\Theta_{\mathbf{M}}$ as in the previous lemma, equation (2.51). After forming Θ' ,

take $\mathbf{T} = \Theta'_{11}{}^{-1}$. Now, (2.57) implies that

$$\begin{aligned} \mathbf{T} [\Theta'_{11} \quad \Theta'_{12}] &= [\mathbf{I} \quad \Theta'_{11}{}^{-1} \Theta'_{12}] \\ &= \begin{matrix} m-d & n_1-d & d & n_2-d \\ + & + & - & - \end{matrix} \\ &= \begin{matrix} m-d \\ n_1-(m-d) \end{matrix} \left[\begin{array}{cc|cc} \mathbf{I} & \mathbf{0} & \mathbf{0} & * \\ \mathbf{0} & \mathbf{I} & * & * \end{array} \right], \end{aligned} \quad (2.62)$$

which has the desired structure and guarantees $\mathbf{M}'_{\Theta} = \mathbf{0}$. \square

For the proof of Theorem 1, it remains to insert the QR decomposition $\mathbf{Q}[\mathbf{R}_A \quad \mathbf{R}_B] = [\mathbf{A} \quad \mathbf{B}]$.

2.C Proof of the Updating Algorithm for SURV

Here, it is proven that the proposed SURV algorithm for updating (2.24) automatically satisfies the structural requirements represented by (2.25), viz.

$$\mathbf{Q}^H_m \begin{bmatrix} n_1 & n_2 \\ \mathbf{N} & \mathbf{X} \end{bmatrix} \Theta = \begin{matrix} m-d & n_1-d & d & n_2-d \\ \mathbf{R}_A & \boxed{\mathbf{0}} & \mathbf{R}_B & \boxed{\mathbf{0}} \end{matrix}, \quad (2.63)$$

$$\mathbf{T}_{n_1} \begin{bmatrix} n_1 & n_2 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \Theta = \begin{matrix} m-d & n_1-d & d & n_2-d \\ \mathbf{I} & \boxed{\mathbf{0}} & \boxed{\mathbf{0}} & * \\ \mathbf{0} & \mathbf{I} & * & * \end{matrix}. \quad (2.64)$$

In these equations, it is important to show that the required “zero” structure (represented by $\boxed{\mathbf{0}}$) is satisfied, in particular for (2.64), as SURV was derived to satisfy (2.63). First, the equations valid for updating an existing decomposition by a single new vector will be written down. Then it will be shown that only 1 rotation per update, the HCR hyperbolic rotation, can destroy the zero structure in (2.64), but that the SURV algorithm maintains the structure without additional operations (this is not valid for general algorithms that compute (2.63), e.g., in [38] an additional hyperbolic rotation was needed). Finally, it will be also shown that additional storage of matrices in (2.25) is not needed.

The proof is technical and it will handle 4 cases separately, corresponding to different values of the pair $(j_c = \pm 1, j'_c = \pm 1)$.

2.C.1 $j_c = +1$ and $j'_c = +1$

Consider the update of an existing decomposition by a new vector \mathbf{n} with signature +1. For the present case where the output signature is also +1, the rank is

constant ($d' = d$), and the required decomposition after the update is

$$\mathbf{Q}'^H_m \left[\begin{array}{c|cc} n_1 & 1 & n_2 \\ \mathbf{N} & \mathbf{n} & \mathbf{X} \end{array} \right] \Theta' = m \left[\begin{array}{cc|cc} m-d & 1 & n_1-d & d & n_2-d \\ \mathbf{R}'_A & \mathbf{0} & \mathbf{0} & \mathbf{R}'_B & \mathbf{0} \end{array} \right], \quad (2.65)$$

$$\mathbf{T}'^{n_1}_1 \left[\begin{array}{c|cc} n_1 & 1 & n_2 \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline & 1 & \mathbf{0} \end{array} \right] \Theta' = \begin{array}{c} m-d \\ 1 \\ n_1-m+d \end{array} \left[\begin{array}{cc|cc} m-d & 1 & n_1-d & d & n_2-d \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & * \\ \hline & 1 & & * & * \\ & & \mathbf{I} & * & * \end{array} \right]. \quad (2.66)$$

From the initial decomposition (2.63), (2.64), after the rows and columns that do not play a role are dropped because they will not change, more compact *update* equations can be obtained as follows:

$$\mathbf{Q}'^H_m \left[\begin{array}{c|cc} m-d & 1 & d \\ \mathbf{R}_A & \mathbf{c} & \mathbf{R}_B \end{array} \right] \Theta_c = m \left[\begin{array}{cc|c} m-d & 1 & d \\ \mathbf{R}'_A & \mathbf{0} & \mathbf{R}'_B \end{array} \right], \quad (2.67)$$

$$\mathbf{T}'_c \begin{array}{c} m-d \\ 1 \end{array} \left[\begin{array}{c|cc} m-d & 1 & d \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline & 1 & \mathbf{0} \end{array} \right] \Theta_c = \begin{array}{c} m-d \\ 1 \end{array} \left[\begin{array}{cc|c} m-d & 1 & d \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline & 1 & * \end{array} \right]. \quad (2.68)$$

Here, $\mathbf{c} = \mathbf{Q}^H \mathbf{n}$, and \mathbf{Q}_c , \mathbf{T}_c and Θ_c are the updates of \mathbf{Q} , \mathbf{T} and Θ , respectively (using proper extensions to make dimensions fit).

The SURV algorithm acts on equation (2.67) and nulls \mathbf{c} using the rotations described in Section 2.4. A difference with the algorithm description is that, now, the columns of the matrices $[\mathbf{R}_A \ \mathbf{c} \ | \ \mathbf{R}_B]$ have been written sorted according to signature, whereas in Section 2.4 this was written $[\mathbf{R}_A \ \mathbf{R}_B \ \mathbf{c}]$ for convenience in the algorithm description there. Also the ordering of the resulting columns is a bit different. Nevertheless, there is an obvious one-to-one match of the rotation operations in both cases—the algorithm operations are the same.

We will follow the algorithm operations step by step and investigate the additional updates required to satisfy (2.68) as well. To facilitate the discussion, equation (2.68) is written as $\mathbf{T}_c \mathbf{E} \Theta_c = \mathbf{E}'$, with \mathbf{E} defined as

$$\mathbf{E} := \left[\begin{array}{c|cc} \mathbf{E}_+ & & \mathbf{E}_- \end{array} \right] = \left[\begin{array}{cc|c} \mathbf{E}_{11} & \mathbf{E}_{12} & \mathbf{E}_{13} \\ \mathbf{E}_{21} & \mathbf{E}_{22} & \mathbf{E}_{23} \end{array} \right] = \begin{array}{c} m-d \\ 1 \end{array} \left[\begin{array}{c|cc} \mathbf{I} & & \mathbf{0} \\ \hline & 1 & \mathbf{0} \end{array} \right]$$

(This subpartitioning of \mathbf{E} will be used to identify its various subblocks throughout the algorithm.)

- (a) GCR: The SURV operations belonging to Θ_c for this step consist of column rotations acting on $[\mathbf{R}_A \ \mathbf{c}]$, and on \mathbf{E}_+ . They destroy the identity matrices in \mathbf{E}_+ , but these are easily recovered by matching rotations in \mathbf{T}_c . Since the entries in the \mathbf{E}_- blocks are zero, these do not change. Thus, $\mathbf{E} = [\mathbf{I} \ \mathbf{0}]$ is still true after this update.

- (b) GRCR: The column rotations in the GRCR operations only act on \mathbf{E}_- . Since $\mathbf{E}_- = \mathbf{0}$, it is not changed.
- (c) HCR: The HCR operation is a hyperbolic rotation $\boldsymbol{\theta}$ acting on \mathbf{c} and a column of \mathbf{R}_B , and on corresponding columns of \mathbf{E}_+ and \mathbf{E}_- . This creates a fill-in in subblock \mathbf{E}_{23} , but that matches the required structure as seen from (2.68). The entry \mathbf{E}_{22} is now unequal to 1 but it is easily scaled back to 1 by a row scaling operation in \mathbf{T}_c . At this point, the required decompositions (2.67), (2.68) have been achieved.

Thus, it has been seen that the required structure (2.68) is obtained by certain operations \mathbf{T}_c ; no additional operations $\boldsymbol{\Theta}_c$ are needed beyond those of the SURV algorithm. Since the resulting decomposition \mathbf{E}' is known (the last row with arbitrary entries $*$ is not needed in future and can be dropped), it is not needed to store \mathbf{E} and \mathbf{T}_c , and these operations do not actually have to be performed as they do not influence (2.67). As mentioned, this is a special feature of SURV and not valid for other more general updating algorithms to achieve (2.67).

2.C.2 $j_c = +1$ and $j'_c = -1$

In this case, the update of \mathbf{N} decreases the rank ($d' = d - 1$). Consequently, the size of \mathbf{R}_A will be increased by adding one column, and \mathbf{R}_B will drop by one column. For brevity, after some details are skipped, the update equations are directly written down as:

$$\mathbf{Q}_c^H \begin{matrix} m-d & 1 & d \\ \mathbf{R}_A & \mathbf{c} & \mathbf{R}_B \end{matrix} \boldsymbol{\Theta}_c = \begin{matrix} m-d & 1 & d-1 & 1 \\ \mathbf{R}'_A & \mathbf{a} & \mathbf{R}'_B & \boxed{\mathbf{0}} \end{matrix};$$

$$\mathbf{R}'_A = [\mathbf{R}''_A \ \mathbf{a}], \quad (2.69)$$

$$\mathbf{T}_c \begin{matrix} m-d & 1 & d \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ 1 & 1 & \mathbf{0} \end{matrix} \boldsymbol{\Theta}_c = \begin{matrix} m-d & 1 & d-1 & 1 \\ \mathbf{I} & \boxed{\mathbf{0}} & \boxed{\mathbf{0}} & * \\ 1 & \boxed{\mathbf{0}} & \boxed{\mathbf{0}} & * \end{matrix}. \quad (2.70)$$

- (a), (b) As before, these SURV operations consist of rotations only acting on \mathbf{E}_+ and \mathbf{E}_- , respectively, and are easily matched with corresponding operations in \mathbf{T} to restore the structure of \mathbf{E} .
- (c) The HCR operation acting on \mathbf{c} and the last column of \mathbf{R}_B creates a fill-in in the last column of \mathbf{E} , but that matches the required structure as seen from (2.70). The entry \mathbf{E}_{22} is now unequal to 1 but it is easily scaled back to 1 without affecting the structure. Thus, the required decompositions (2.69), (2.70) have been achieved without additional operations in $\boldsymbol{\Theta}_c$, no additional operations are needed beyond those of the SURV algorithm.

2.C.3 $j_{\mathbf{c}} = -1$ and $j'_{\mathbf{c}} = -1$

In this case, after \mathbf{X} is updated by \mathbf{x} , the rank remains constant ($d' = d$). The update equations are (with $\mathbf{c} = \mathbf{Q}^H \mathbf{x}$):

$$\mathbf{Q}_c^H \begin{matrix} m-d & d & 1 \\ m & & \end{matrix} \left[\mathbf{R}_A \mid \mathbf{R}_B \mid \mathbf{c} \right] \Theta_c = \begin{matrix} m-d & d & 1 \\ m & & \end{matrix} \left[\mathbf{R}'_A \mid \mathbf{R}'_B \mid \boxed{\mathbf{0}} \right], \quad (2.71)$$

$$\mathbf{T}_c \begin{matrix} m-d & d & 1 \\ m-d & & \end{matrix} \left[\mathbf{I} \mid \mathbf{0} \mid \mathbf{0} \right] \Theta_c = \begin{matrix} m-d & d & 1 \\ m-d & & \end{matrix} \left[\mathbf{I} \mid \boxed{\mathbf{0}} \mid * \right]. \quad (2.72)$$

As before, the SURV operations that act only on the “+” part or on the “-” part of \mathbf{E} can be easily inverted by corresponding rotations in \mathbf{T}_c to maintain the same structure of \mathbf{E} . The HCR operation acting on \mathbf{c} and the last column of \mathbf{R}_A creates a fill-in in the last column of \mathbf{E} , but that matches the required structure. Subsequently, the last row of \mathbf{E} is scaled by \mathbf{T}_c and the required \mathbf{E}' is obtained. No additional operations in Θ_c are needed beyond those of the SURV algorithm.

2.C.4 $j_{\mathbf{c}} = -1$ and $j'_{\mathbf{c}} = +1$

In this case, after \mathbf{X} is updated, the rank will increase ($d' = d + 1$). This implies that \mathbf{R}_A shrinks and \mathbf{R}_B grows by one column. The update equations are:

$$\mathbf{Q}_c^H \begin{matrix} m-d & d & 1 \\ m & & \end{matrix} \left[\mathbf{R}_A \mid \mathbf{R}_B \mid \mathbf{c} \right] \Theta_c = \begin{matrix} m-d' & 1 & d & 1 \\ m & & & \end{matrix} \left[\mathbf{R}'_A \mid \boxed{\mathbf{0}} \mid \mathbf{R}''_B \mid \mathbf{b} \right];$$

$$\mathbf{R}'_B = [\mathbf{R}''_B \mid \mathbf{b}], \quad (2.73)$$

$$\mathbf{T}_c \begin{matrix} m-d & d & 1 \\ m-d & & \end{matrix} \left[\mathbf{I} \mid \mathbf{0} \mid \mathbf{0} \right] \Theta_c = \begin{matrix} m-d' & 1 & d & 1 \\ m-d' & & & \end{matrix} \left[\begin{array}{c|c|c|c} \mathbf{I} & \boxed{\mathbf{0}} & \boxed{\mathbf{0}} & \boxed{\mathbf{0}} \\ \hline & 1 & * & * \end{array} \right]. \quad (2.74)$$

The first steps are as before and the structure of \mathbf{E} is maintained. The HCR operation acting on \mathbf{c} and the last column of \mathbf{R}_A zeroes the last column, and creates a fill-in in the last column of \mathbf{E} (corresponding to \mathbf{c}). However, that fill-in occurs in the last row, which will be dropped as $m - d'$ is one less than $m - d$: it matches the structure in (2.74). No additional operations in Θ_c are needed beyond those of the SURV algorithm.

Chapter 3

Blind Beamforming Techniques for Separating Partially Overlapping Data Packets

3.1 Introduction

Co-channel interference is a growing concern in wireless communication applications. One approach for interference mitigation is to use an antenna array. Beamforming techniques allow to receive the target signals and suppress the interference signals, assuming the array response vector of each of the signals is known. Blind beamforming techniques aim to estimate these array response vectors.

In many cases, the interference is intermittent and unsynchronized. For example, inter-cell interference reduces channel capacity in MIMO cellular networks [64, 65]. Also, ad-hoc communication systems or wireless sensor networks where devices transmit whenever data is available result in multiple partially overlapping data packets at the receiver. Other examples are RFID systems with multiple tags, the automatic identification system (AIS) for ships, wherein transponders at vessels periodically report their locations [6, 23], the secondary surveillance radar (SSR) [22, 66] and similar ADS-B transponder systems for airplanes. Another example is multiple unsynchronized WLAN systems in the same service area.

In this chapter the separation of partially overlapping data packets using blind beamforming techniques is considered. Within an analysis window, the target signals could also be continuously present, but the interference signals are assumed intermittent with arbitrary arrival times. Some methods to combat this type of interference are based on known training sequences [67, 68]. Much attention will

be paid to applications where the target cannot be simply identified by a unique training code (as in CDMA), or the target can be identified by this training code only after interference is suppressed.

The approach is to collect a block of data from an analysis window (this could actually be a sliding window). Then the data block is split into two sub-blocks, and the subspaces present in respective blocks are compared. Specifically, a generalized eigenvalue decomposition allows to match basis vectors within the subspaces with each other, and target/interference signal classification is based on detecting differences in signal power between the two blocks. Subspace intersection allows to find the common and difference subspaces. After subspace estimation, oblique projections are used to separate the target signals from the interference signals.

The objective of this chapter is not to separate each individual target signal. If the resulting subspace contains multiple target signals, then other properties should provide further separation, such as constant modulus properties (the ACMA algorithm [12]) or related algorithms based on higher-order statistics (JADE [69] and MUK [70]). Such algorithms typically cannot handle intermittent interference or signals with non-stationary properties, and the algorithms in this chapter can serve as a preprocessing step both to filter out the intermittent signals and to arrive at a more synchronous scenario.

This chapter has two parts. In the first part, some generic algorithms are proposed based on the generalized SVD (GSVD) [11] and the generalized eigenvalue decomposition (GEVD). In the second part, the details of these algorithms are investigated, and then the improved algorithms are proposed based on a new tool—the signed URV (SURV) algorithm (See Chapter 2) [14, 35]. This leads to computationally efficient techniques that allow for tracking and improved noise processing. Simulations are provided to confirm the results.

Interference cancellation using oblique projections has been studied in [71, 72]. Here, this chapter focuses on the estimation of the required subspace information by oblique projections so that these tools can be applied. Not many literatures consider intermittent interference cancellation based on subspace techniques. For the blind separation of partially overlapping SSR signals, Petrochilos *et al.* proposed a block-based tracking algorithm [73] based on detecting and projecting out rank-1 components representing time segments where only a single source is present. The existence of such segments can be considered as a simplified special case of the scenarios that are considered in this chapter.

3.2 Data Model

3.2.1 Signals

Consider discrete-time intermittent signals $s_i[k]$ where i is the signal index and k is the time index. Each signal consists of a stretch of N_p nonzero values, preceded and followed by zeros. For concise notation, all intermittent signals will have the

same packet length N_p (this will be generalized at a later stage). There are d signals, and they are stacked in a vector $\mathbf{s}[k] = [s_1[k], \dots, s_d[k]]^T$.

Assume that the receiver has an antenna array of M antennas, and stack the (complex-valued) antenna signals into a vector $\mathbf{x}[k] \in \mathbb{C}^M$. In a narrowband scenario, the received signal vector is

$$\mathbf{x}[k] = \mathbf{h}_1 s_1[k] + \dots + \mathbf{h}_d s_d[k] + \mathbf{n}[k] = \mathbf{H}\mathbf{s}[k] + \mathbf{n}[k] \quad (3.1)$$

where the vectors \mathbf{h}_i , $i = 1, \dots, d$ are the channel vectors (array response vectors) corresponding to each signal, $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_d] \in \mathbb{C}^{M \times d}$ is the channel matrix, and $\mathbf{n} \in \mathbb{C}^M$ is the noise vector.

Also assume that the channel matrix \mathbf{H} has full column rank. This also implies that $d \leq M$. In the applications in this chapter, there is a scaling indeterminacy between signals and channel vectors; without loss of generality, assume that the channel vectors are all scaled to $\|\mathbf{h}_i\| = 1$. The noise is assumed to be i.i.d. zero mean Gaussian vectors with covariance matrix $\mathbf{R}_n = \mathbb{E}\{\mathbf{n}\mathbf{n}^H\} = \sigma^2 \mathbf{I}$. Also assume that the noise power σ^2 is known.

After collecting N_s samples $\mathbf{x}[k]$, one can collect these into a matrix $\mathbf{X} = [\mathbf{x}[1], \dots, \mathbf{x}[N_s]]$, and similarly for the signals and the noise. The corresponding data model is

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{N}. \quad (3.2)$$

The sample covariance matrix is $\hat{\mathbf{R}}_x = \frac{1}{N_s} \mathbf{X}\mathbf{X}^H$.

3.2.2 Stationary Data Model

The source signals are nonstationary. The proposed algorithms will be based on second-order statistics, i.e., the sample covariance matrix $\hat{\mathbf{R}}_x$. For the development of algorithms, it is necessary to replace the *model* for the received data vector $\mathbf{x}[k]$ with an “equivalent” stationary model $\mathbf{y}[k]$

$$\mathbf{y}[k] = \mathbf{H}\mathbf{s}[k] + \mathbf{n}[k] \quad (3.3)$$

wherein the source signals are modeled as (block) wide-sense stationary random vectors and with “the same” second-order statistics. We develop algorithms for $\mathbf{y}[k]$, but use the available data $\mathbf{x}[k]$ in the actual application. There are subtle differences between these models, which have to do with asymptotic properties and cross-correlations of signals and will be discussed this in Section 3.8.

For the model $\mathbf{y}[k]$, assume that the signals $s_i[k]$ and the noise $n_i[k]$ are independent zero mean complex random variables with covariance matrices \mathbf{R}_s and \mathbf{R}_n , respectively. In this case, the data model for $\mathbf{R}_y = \mathbb{E}\{\mathbf{y}[k]\mathbf{y}^H[k]\}$ is

$$\mathbf{R}_y = \mathbf{H}_s \mathbf{R}_s \mathbf{H}_s^H + \mathbf{R}_n.$$

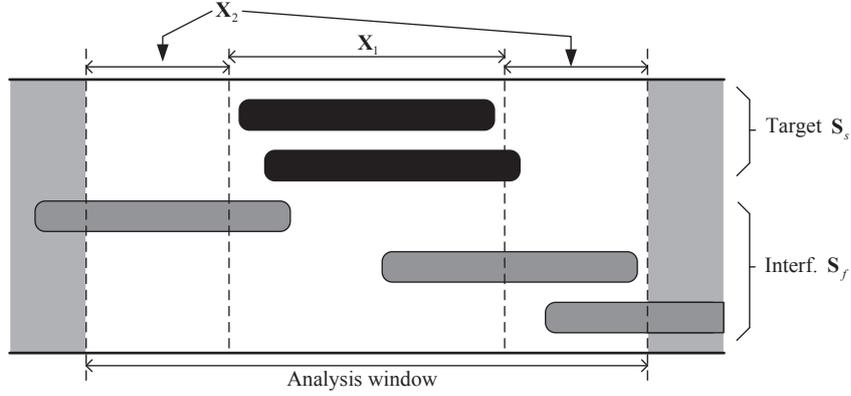


Figure 3.1: Scenario 1: target signals are present mostly in \mathbf{X}_1 .

3.2.3 Separation Scenarios

Assume that N_s samples of data corresponding to an “analysis window” have been obtained. The proposed algorithms are based on splitting this window into parts and comparing the subspaces determined by corresponding part. This can be done in several ways, each corresponding to different definitions of target signals and interference signals, as explained below.

As a refinement of Section 3.2.2, now assume that there are d_s target signals and d_f interference signals. The channel vectors of the target signals are collected in a matrix \mathbf{H}_s , and those of the interference signals in \mathbf{H}_f . For convenience, define $\mathbf{H} = [\mathbf{H}_s, \mathbf{H}_f]$.

In Scenario 1, the analysis window is split into three blocks. The middle block contains mostly the target signals, and the data matrix is denoted by \mathbf{X}_1 . The first and the third block contain mostly the interference signals. The latter two blocks are combined into a single block \mathbf{X}_2 as shown in Fig. 3.1. The corresponding data model is

$$\begin{aligned}\mathbf{X}_1 &= \mathbf{H}_s \mathbf{S}_{s1} + \mathbf{H}_f \mathbf{S}_{f1} + \mathbf{N}_1 \\ \mathbf{X}_2 &= \mathbf{H}_s \mathbf{S}_{s2} + \mathbf{H}_f \mathbf{S}_{f2} + \mathbf{N}_2.\end{aligned}\quad (3.4)$$

In general, \mathbf{X}_1 contains N_1 samples, \mathbf{X}_2 has N_2 samples, and $N_s = N_1 + N_2$ is the number of samples in the analysis window. For concise notation, initially assume that both data matrices have the same number of samples $N = N_s/2$, and also that $N = N_p$, the number of samples in a data packet. The generalization will be presented in Section 3.8.

For properly modeling, each nonstationary matrix \mathbf{X}_i can be modeled with a stationary data model \mathbf{Y}_i which consists of zero mean complex random variables with covariance matrices

$$\begin{aligned}\mathbf{R}_1 &= \mathbf{H}_s \mathbf{R}_{s1} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f1} \mathbf{H}_f^H + \mathbf{R}_{n1} \\ \mathbf{R}_2 &= \mathbf{H}_s \mathbf{R}_{s2} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f2} \mathbf{H}_f^H + \mathbf{R}_{n2}.\end{aligned}\quad (3.5)$$

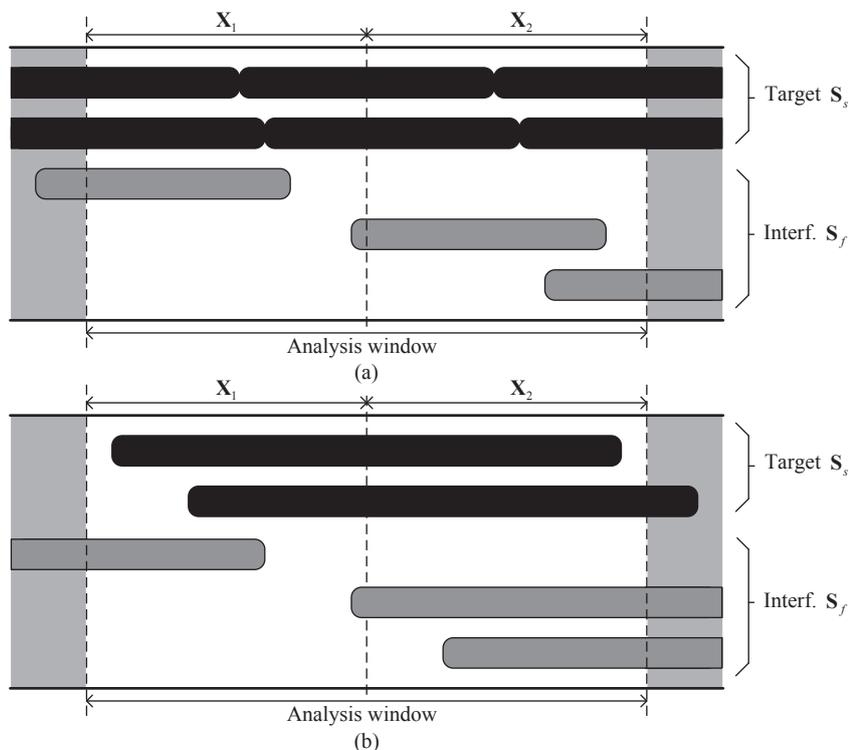


Figure 3.2: Scenario 2: target signals are equally present in both data blocks, whereas each interference signal is concentrated in only one of these blocks.

Here, \mathbf{R}_{s1} and \mathbf{R}_{s2} are covariance matrices of the target signals, with size $d_s \times d_s$, and \mathbf{R}_{f1} and \mathbf{R}_{f2} are covariance matrices of the interference signals, with size $d_f \times d_f$. Also assume $\mathbf{R}_{n1} = \mathbf{R}_{n2} = \sigma^2 \mathbf{I}$.

The distinction between target signals and interference signals in Scenario 1 is described by

$$\mathbf{R}_{s1} > \mathbf{R}_{s2}, \quad \mathbf{R}_{f1} < \mathbf{R}_{f2}. \quad (3.6)$$

(This will be refined this by introducing a scaling factor α .) This defines that the target signals are stronger (larger in power) in the first data block than in the second data block. Similarly, in the second data block the interference signals are stronger than in the first data block.

In Scenario 2 (see Fig. 3.2), assume that a target signal fills the complete analysis window. The window is split into two consecutive blocks, and the assumption is that each target signal is equally present in each block, whereas each interference signal is concentrated mostly in one of the blocks, but not equally strong in both. The data samples of each block are collected into two corresponding matrices

ces \mathbf{X}_1 and \mathbf{X}_2 , that have the same data model (3.4) as before. Each data block consists of N samples. For the development of the algorithms, the nonstationary data model \mathbf{X}_i , $i = 1, 2$ is replaced by a stationary model \mathbf{Y}_i with a covariance model as in (3.5). The distinction between target signals and interference signals is now based on

$$\mathbf{R}_{s1} \approx \mathbf{R}_{s2}, \quad \mathbf{R}_{f1} \neq \mathbf{R}_{f2}. \quad (3.7)$$

More precisely, a threshold has to be defined that specifies when two signal powers are approximately equal.

3.2.4 Objective

The objective of this chapter is to compute a single separating beamforming matrix \mathbf{W} of size $M \times d_s$, such that

$$\hat{\mathbf{S}}_{s1} = \mathbf{W}^H \mathbf{X}_1, \quad \hat{\mathbf{S}}_{s2} = \mathbf{W}^H \mathbf{X}_2$$

are estimates of the target signals in each block, with the interference signals suppressed. The purpose is not to separate the individual target signals but to allow for an arbitrary linear combination \mathbf{M}_s , an unknown full rank matrix of size $d_s \times d_s$. Thus, the special attention is paid to

$$\mathbf{W}^H \mathbf{H}_s = \mathbf{M}_s, \quad \mathbf{W}^H \mathbf{H}_f = \mathbf{0}.$$

3.3 Tools from Linear Algebra

3.3.1 Singular Value Decomposition (SVD)

The “skinny” SVD of a matrix $\mathbf{Y} : M \times N$ with $M \leq N$ is given by $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$, where $\mathbf{U} : M \times M$ is unitary, $\mathbf{V} : N \times M$ is semi-unitary ($\mathbf{V}^H\mathbf{V} = \mathbf{I}$), and $\mathbf{\Sigma} : M \times M$ is square and diagonal with nonnegative entries. For a given threshold $\epsilon \geq 0$, the singular values are sorted and the matrices are partitioned as

$$\mathbf{Y} = [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{bmatrix} \mathbf{\Sigma}_1 & \\ & \mathbf{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix},$$

where $\mathbf{\Sigma}_1 > \epsilon \mathbf{I}$ and $\mathbf{\Sigma}_2 < \epsilon \mathbf{I}$. A low-rank approximation of \mathbf{Y} is then given by the truncated SVD (TSVD),

$$\hat{\mathbf{Y}} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^H, \quad (3.8)$$

which is such that $\|\hat{\mathbf{Y}} - \mathbf{Y}\| < \epsilon$. The columns of \mathbf{U}_1 form a basis for the “signal subspace” $\text{ran}(\hat{\mathbf{Y}})$ of \mathbf{Y} , with threshold ϵ . An orthogonal projection onto this subspace is $\mathbf{P} = \mathbf{U}_1 \mathbf{U}_1^H$.

3.3.2 Generalized Singular Value Decomposition (GSVD)

The GSVD of two matrices $\mathbf{Y}_1, \mathbf{Y}_2$ (each of size $M \times N$ with $M \leq N$) is denoted by

$$\text{GSVD}(\mathbf{Y}_1, \mathbf{Y}_2) \Leftrightarrow \begin{cases} \mathbf{Y}_1 = \mathbf{F}\mathbf{C}\mathbf{U}^H \\ \mathbf{Y}_2 = \mathbf{F}\mathbf{S}\mathbf{V}^H \end{cases}$$

where $\mathbf{F} : M \times M$ is an invertible matrix, \mathbf{C} and \mathbf{S} are square positive diagonal matrices, and \mathbf{U}, \mathbf{V} are semi-unitary matrices of size $N \times M$. Without loss of generality, each column of \mathbf{F} is scaled to norm 1 by exchanging factors with \mathbf{C} and \mathbf{S} .

For a given threshold $\epsilon \geq 0$, the entries of \mathbf{C} and \mathbf{S} can be further sorted and then the matrices can be partitioned as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & & & \\ & \mathbf{C}_2 & & \\ & & \mathbf{C}_3 & \\ & & & \mathbf{C}_4 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} \mathbf{S}_1 & & & \\ & \mathbf{S}_2 & & \\ & & \mathbf{S}_3 & \\ & & & \mathbf{S}_4 \end{bmatrix}$$

where the partitioning is defined by

$$\begin{array}{lll} \mathbf{C}_1 > \epsilon \mathbf{I}, & \mathbf{S}_1 > \epsilon \mathbf{I}, & \mathbf{C}_3 < \epsilon \mathbf{I}, \quad \mathbf{S}_3 > \epsilon \mathbf{I}, \\ \mathbf{C}_2 > \epsilon \mathbf{I}, & \mathbf{S}_2 < \epsilon \mathbf{I}, & \mathbf{C}_4 < \epsilon \mathbf{I}, \quad \mathbf{S}_4 < \epsilon \mathbf{I}. \end{array}$$

A corresponding partitioning of \mathbf{F} is

$$\mathbf{F} = [\mathbf{F}_1 \quad \mathbf{F}_2 \quad \mathbf{F}_3 \quad \mathbf{F}_4].$$

This provides the following subspace information:

- $\text{ran}(\mathbf{F}_1)$ contains the common column span (with tolerance ϵ), i.e., $\text{ran}(\mathbf{Y}_1) \cap \text{ran}(\mathbf{Y}_2)$
- $\text{ran}(\mathbf{F}_2)$ is the subspace of the columns in $\text{ran}(\mathbf{Y}_1)$ but not in $\text{ran}(\mathbf{Y}_2)$,
- Similarly, $\text{ran}(\mathbf{F}_3)$ is the subspace of the columns in $\text{ran}(\mathbf{Y}_2)$ but not in $\text{ran}(\mathbf{Y}_1)$,
- $\text{ran}(\mathbf{F}_4)$ is a common left null space.

Thus, the GSVD provides a generic technique for subspace intersection, and this has been exploited in source separation applications, see e.g., [74, 75]. Note that \mathbf{F} is invertible but generally not unitary, so that these subspaces are not necessarily orthogonal to each other.

3.3.3 Generalized Eigenvalue Decomposition

After the data matrices in the previous subsection are “squared”, a related decomposition for positive definite matrices $\mathbf{R}_1, \mathbf{R}_2$ can be obtained as:

$$\text{GEVD}(\mathbf{R}_1, \mathbf{R}_2) \quad \Leftrightarrow \quad \begin{cases} \mathbf{R}_1 &= \frac{1}{N} \mathbf{F} \mathbf{D} \mathbf{F}^H \\ \mathbf{R}_2 &= \frac{1}{N} \mathbf{F} \mathbf{K} \mathbf{F}^H \end{cases}$$

where \mathbf{F} is invertible and \mathbf{D}, \mathbf{K} are diagonal and positive. If $\mathbf{R}_1 = \frac{1}{N} \mathbf{Y}_1 \mathbf{Y}_1^H$ and $\mathbf{R}_2 = \frac{1}{N} \mathbf{Y}_2 \mathbf{Y}_2^H$, then the GEVD is related to the GSVD via $\mathbf{D} = \mathbf{C}^2$ and $\mathbf{K} = \mathbf{S}^2$. Therefore, the same partitioning of the GEVD into four subspaces as for the GSVD can be defined. For symmetric matrices, the GEVD is sure to exist if $\mathbf{R}_1 > 0$ or $\mathbf{R}_2 > 0$; otherwise the existence of the decomposition is unclear (\mathbf{D} or \mathbf{K} may become complex).

3.3.4 Oblique Projections

A square matrix \mathbf{P} is an orthogonal projection if $\mathbf{P}^2 = \mathbf{P}$ and $\mathbf{P}^H = \mathbf{P}$. If \mathbf{G} is a “tall” matrix of full column rank, then

$$\mathbf{P}_{\mathbf{G}} = \mathbf{G} \mathbf{G}^\dagger = \mathbf{G} (\mathbf{G}^H \mathbf{G})^{-1} \mathbf{G}^H$$

is an orthogonal projection onto the column span \mathcal{G} of \mathbf{G} . The projection onto \mathcal{G} is independent of the choice of basis: If $\tilde{\mathbf{G}} = \mathbf{G} \mathbf{M}$ is any other basis of the span \mathcal{G} (\mathbf{M} is an invertible matrix), then $\mathbf{P}_{\tilde{\mathbf{G}}} = \mathbf{P}_{\mathbf{G}}$. The projection onto the orthogonal complement is $\mathbf{P}_{\mathbf{G}}^\perp := \mathbf{I} - \mathbf{P}_{\mathbf{G}}$.

A square matrix \mathbf{E} is an oblique projection if $\mathbf{E}^2 = \mathbf{E}$. Let $\mathbf{H} = [\mathbf{F} \ \mathbf{G}]$ be of full column rank, then

$$\mathbf{E}_{\mathbf{F}, \mathbf{G}} := \mathbf{H} \begin{bmatrix} \mathbf{I} & \\ & \mathbf{0} \end{bmatrix} \mathbf{H}^\dagger \quad (3.9)$$

is an oblique projection such that $\mathbf{E} \mathbf{F} = \mathbf{F}$ and $\mathbf{E} \mathbf{G} = \mathbf{0}$, so that it projects onto the column span \mathcal{F} of \mathbf{F} and projects away the column span \mathcal{G} of \mathbf{G} . Also \mathbf{E} is independent of the choice of basis for these subspaces. It was shown by Behrens and Scharf [71] that

$$\mathbf{H}^\dagger = \begin{bmatrix} (\mathbf{F}^H \mathbf{P}_{\mathbf{G}}^\perp \mathbf{F})^{-1} \mathbf{F}^H \mathbf{P}_{\mathbf{G}}^\perp \\ (\mathbf{G}^H \mathbf{P}_{\mathbf{F}}^\perp \mathbf{G})^{-1} \mathbf{G}^H \mathbf{P}_{\mathbf{F}}^\perp \end{bmatrix}.$$

Therefore, another expression for $\mathbf{E}_{\mathbf{F}, \mathbf{G}}$ is

$$\mathbf{E}_{\mathbf{F}, \mathbf{G}} = \mathbf{F} (\mathbf{F}^H \mathbf{P}_{\mathbf{G}}^\perp \mathbf{F})^{-1} \mathbf{F}^H \mathbf{P}_{\mathbf{G}}^\perp. \quad (3.10)$$

3.3.5 Subspace Intersection

Consider two subspaces \mathcal{A} and \mathcal{B} , with orthogonal bases $\mathbf{Q}_A, \mathbf{Q}_B$. The intersection of \mathcal{A} and \mathcal{B} is obtained by computing the SVD

$$[\mathbf{Q}_A \ \mathbf{Q}_B] = [\mathbf{U}_c \ \mathbf{U}_d] \begin{bmatrix} \Sigma_c & \\ & \Sigma_d \end{bmatrix} \begin{bmatrix} \mathbf{V}_c^H \\ \mathbf{V}_d^H \end{bmatrix} \quad (3.11)$$

where the singular values are sorted such that $\Sigma_c > \xi \mathbf{I}$, $\Sigma_d < \xi \mathbf{I}$, and ξ is a detection threshold ($1 < \xi < \sqrt{2}$). The common subspace is related to singular values close to $\sqrt{2}$. The corresponding left singular vectors are collected into \mathbf{U}_c and form a basis for the common subspace.

An alternative is to use the GSVD, as in Section 3.3.3. In this case, orthogonal bases are not needed. Starting from two matrices $\mathbf{Y}_1, \mathbf{Y}_2$, the matrix \mathbf{F}_1 resulting from GSVD($\mathbf{Y}_1, \mathbf{Y}_2$) is a (nonorthogonal) basis of the intersection. Also here, a detection threshold should be specified.

3.4 Separation Using a Single Data Block

For reference, an algorithm for interference suppression using a single data block is first presented. At this time, additional information is required: either the target signal subspace and the interference subspace must be known, or one of them as well as the covariance matrix of the measurement data. The algorithm is based on oblique projections. Scharf and McCloud [72] have given an elaborate overview of the different cases and have distinguished between deterministic and stochastic signal models. Here, only the deterministic approach will be discussed.

Consider a stationary data model of the form (the sample index is ignored in the notation)

$$\mathbf{y} = \mathbf{H}_s \mathbf{s}_s + \mathbf{H}_f \mathbf{s}_f + \mathbf{n}.$$

For a deterministic signal model, only the noise is considered as a random variable, and it is complex Gaussian distributed with zero mean and covariance matrix $\mathbf{R}_n = \sigma^2 \mathbf{I}$. The signals are unknown and no statistical assumptions are made on them. $\mathbf{H} = [\mathbf{H}_s, \mathbf{H}_f]$ is further assumed to be “tall” and of full column rank $d = d_s + d_f$, where \mathbf{H}_s has d_s columns and \mathbf{H}_f has d_f columns.

If both \mathbf{H}_s and \mathbf{H}_f are known, then the deterministic signal model is sufficient. In this case, the interference signals can be suppressed by an oblique projection $\mathbf{E}_{s,f} = \mathbf{H}_s (\mathbf{H}_s^H \mathbf{P}_f^\perp \mathbf{H}_s)^{-1} \mathbf{H}_s^H \mathbf{P}_f^\perp$ as in (3.9). A related “zero-forcing” beamformer to suppress the interference signals and to estimate \mathbf{s}_s is

$$\mathbf{W}^H = [\mathbf{I} \quad \mathbf{0}] [\mathbf{H}_s \quad \mathbf{H}_f]^\dagger = (\mathbf{H}_s^H \mathbf{P}_f^\perp \mathbf{H}_s)^{-1} \mathbf{H}_s^H \mathbf{P}_f^\perp. \quad (3.12)$$

The corresponding target signal estimate is

$$\hat{\mathbf{s}}_s = \mathbf{W}^H \mathbf{y} = \mathbf{s}_s + \mathbf{W}^H \mathbf{n}.$$

This beamformer is obtained through the following process. First the interference signals are cancelled by applying an orthogonal projection \mathbf{P}_f^\perp to \mathbf{y} , leading to projected data $\tilde{\mathbf{y}} := \mathbf{P}_f^\perp \mathbf{y}$ with model

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}_s \mathbf{s}_s + \tilde{\mathbf{n}} \quad (3.13)$$

where $\tilde{\mathbf{H}}_s = \mathbf{P}_f^\perp \mathbf{H}_s$ and $\tilde{\mathbf{n}} = \mathbf{P}_f^\perp \mathbf{n}$. Subsequently, from $\tilde{\mathbf{y}}$ a least squares (or maximum likelihood) estimation of \mathbf{s}_s is done, leading to a zero-forcing beamformer on the projected data

$$\tilde{\mathbf{W}}^H := (\tilde{\mathbf{H}}_s^H \tilde{\mathbf{H}}_s)^{-1} \tilde{\mathbf{H}}_s^H = (\mathbf{H}_s^H \mathbf{P}_f^\perp \mathbf{H}_s)^{-1} \mathbf{H}_s^H \mathbf{P}_f^\perp. \quad (3.14)$$

Comparison with (3.12) shows that $\mathbf{W}^H = \tilde{\mathbf{W}}^H \mathbf{P}_f^\perp = \tilde{\mathbf{W}}^H$.

Suppose now that only subspace information is known, i.e., only the arbitrary bases $\bar{\mathbf{H}}_s$ and $\bar{\mathbf{H}}_f$ for the column spans of \mathbf{H}_s and \mathbf{H}_f are known. Then $\bar{\mathbf{H}}_s = \mathbf{H}_s \mathbf{M}_s$ and $\bar{\mathbf{H}}_f = \mathbf{H}_f \mathbf{M}_f$ for unknown square matrices $\mathbf{M}_s, \mathbf{M}_f$. The beamformer

$$\bar{\mathbf{W}}^H = [\mathbf{I} \ \mathbf{0}] [\bar{\mathbf{H}}_s \ \bar{\mathbf{H}}_f]^\dagger = [\mathbf{M}_s^{-1} \ \mathbf{0}] [\mathbf{H}_s \ \mathbf{H}_f]^\dagger = \mathbf{M}_s^{-1} \mathbf{W}^H$$

will result in a target signal estimate

$$\hat{\mathbf{s}}_s = \mathbf{M}_s^{-1} \hat{\mathbf{s}}_s.$$

The overbar in $\hat{\mathbf{s}}_s$ denotes some unknown linear combination of the target signals. This is reasonable, as there is no information to further separate these signals at this point. Since \mathbf{M}_s is unknown anyway, it is allowed to drop any invertible factor at the left in (3.14), and it is sufficient to consider

$$\bar{\mathbf{W}}^H = \bar{\mathbf{H}}_s^H \mathbf{P}_f^\perp$$

as a separating beamformer. This is an interference projection followed by a subspace matched filter.

3.5 Separation Using Two Data Blocks

For the applications, the preceding single-block algorithm is not directly applicable, as it requires subspace information that is not yet available. In this section, the algorithms to estimate these subspaces will be developed.

Recall that the covariance model for $\mathbf{Y}_1, \mathbf{Y}_2$ is

$$\begin{aligned} \mathbf{R}_1 &= \mathbf{H}_s \mathbf{R}_{s1} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f1} \mathbf{H}_f^H + \mathbf{R}_{n1} \\ \mathbf{R}_2 &= \mathbf{H}_s \mathbf{R}_{s2} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f2} \mathbf{H}_f^H + \mathbf{R}_{n2}, \end{aligned} \quad (3.15)$$

where $[\mathbf{H}_s, \mathbf{H}_f]$ is of full column rank, with columns normalized to norm 1.

3.5.1 Scenario 1

In Scenario 1, the distinction between target signals and interference signals is based on

$$\mathbf{R}_{s1} > \alpha^2 \mathbf{R}_{s2}, \quad \mathbf{R}_{f1} < \alpha^2 \mathbf{R}_{f2}, \quad (3.16)$$

where in comparison with (3.7) the parameter α ($\alpha > 1$) is used as a way to control the threshold on detecting a target. The choice of α will be discussed later in Section 3.8. Consider first the noiseless case. In this case, it follows that

$$\begin{aligned} \mathbf{R}_1 &= [\mathbf{H}_s \ \mathbf{H}_f] \begin{bmatrix} \mathbf{R}_{s1} & \mathbf{R}_{f1} \\ \mathbf{R}_{s2} & \mathbf{R}_{f2} \end{bmatrix} \begin{bmatrix} \mathbf{H}_s^H \\ \mathbf{H}_f^H \end{bmatrix}, \\ \mathbf{R}_2 &= [\mathbf{H}_s \ \mathbf{H}_f] \begin{bmatrix} \mathbf{R}_{s1} & \mathbf{R}_{f1} \\ \mathbf{R}_{s2} & \mathbf{R}_{f2} \end{bmatrix} \begin{bmatrix} \mathbf{H}_s^H \\ \mathbf{H}_f^H \end{bmatrix}. \end{aligned} \quad (3.17)$$

A GSVD of $(\mathbf{Y}_1, \mathbf{Y}_2)$ or equivalently a GEVD of $(\mathbf{R}_1, \mathbf{R}_2)$ can be computed. This leads to

$$\begin{cases} \mathbf{R}_1 = \frac{1}{N} \mathbf{F} \mathbf{D} \mathbf{F}^H \\ \mathbf{R}_2 = \frac{1}{N} \mathbf{F} \mathbf{K} \mathbf{F}^H \end{cases}$$

For a given threshold $\epsilon \geq 0$, the generalized eigenvalues can be sorted, and then \mathbf{F} , \mathbf{D} , \mathbf{K} ($\mathbf{F} = [\mathbf{F}_1 \ \mathbf{F}_2 \ \mathbf{F}_3]$) can be partitioned as:

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & & \\ & \mathbf{D}_2 & \\ & & \mathbf{D}_3 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & & \\ & \mathbf{K}_2 & \\ & & \mathbf{K}_3 \end{bmatrix} \quad (3.18)$$

where the partitioning is defined such that

$$\begin{aligned} \mathbf{D}_1 &> \epsilon^2 \mathbf{I} & \mathbf{K}_2 &> \epsilon^2 \mathbf{I} \\ \mathbf{D}_3 &< \epsilon^2 \mathbf{I} & \mathbf{K}_3 &< \epsilon^2 \mathbf{I} \end{aligned}$$

and moreover

$$\mathbf{D}_1 > \alpha^2 \mathbf{K}_1, \quad \mathbf{D}_2 < \alpha^2 \mathbf{K}_2. \quad (3.19)$$

By comparing this decomposition with (3.16)–(3.17) and by using the uniqueness of the GEVD, it is immediately found that

$$\text{ran}(\mathbf{F}_1) = \text{ran}(\mathbf{H}_s), \quad \text{ran}(\mathbf{F}_2) = \text{ran}(\mathbf{H}_f).$$

Thus, the GEVD of $(\mathbf{R}_1, \mathbf{R}_2)$, or the GSVD of $(\mathbf{Y}_1, \mathbf{Y}_2)$, gives directly the required subspace information. By using \mathbf{F} , the oblique projector can be constructed to cancel the interference signals in several ways. The starting point is the following equation

$$\mathbf{E} = [\mathbf{F}_1 \ \mathbf{F}_2] \begin{bmatrix} \mathbf{I} & \\ & \mathbf{0} \end{bmatrix} [\mathbf{F}_1 \ \mathbf{F}_2]^\dagger \quad (3.20)$$

and the corresponding separating beamformer is

$$\mathbf{W}^H = [\mathbf{I} \ \mathbf{0}] [\mathbf{F}_1 \ \mathbf{F}_2]^\dagger = (\mathbf{F}_1^H \mathbf{P}_2^\perp \mathbf{F}_1)^{-1} \mathbf{F}_1^H \mathbf{P}_2^\perp \quad (3.21)$$

where $\mathbf{P}_2^\perp = \mathbf{I} - \mathbf{F}_2 \mathbf{F}_2^\dagger = \mathbf{I} - \mathbf{F}_2 (\mathbf{F}_2^H \mathbf{F}_2)^{-1} \mathbf{F}_2^H$. It is sufficient to keep $\mathbf{W}^H = \mathbf{F}_1^H \mathbf{P}_2^\perp$. Alternatively, the starting point is selected as

$$\mathbf{E} = \mathbf{F} \begin{bmatrix} \mathbf{I} & & \\ & \mathbf{0} & \\ & & \mathbf{0} \end{bmatrix} \mathbf{F}^{-1}$$

and the corresponding separating beamformer is

$$\mathbf{W}^H = [\mathbf{I} \quad \mathbf{0} \quad \mathbf{0}] \mathbf{F}^{-1}. \quad (3.22)$$

This is equivalent to (3.21) if $\text{ran}(\mathbf{F}_3)$ is orthogonal to $\text{ran}([\mathbf{F}_1, \mathbf{F}_2])$. Although (3.22) is more general, the beamformer in (3.21) is better conditioned (leading to less noise enhancement).

This result is valid also when the $\mathbf{R}_{s_i}, \mathbf{R}_{f_i}$ are not diagonal (i.e., target signals and/or interference signals have nonzero cross-correlations). However, for independent signals, if a generalized eigenvalue $(\mathbf{D})_{ii}/(\mathbf{K})_{ii}$ is unique, then the corresponding vector \mathbf{f}_i is a column of \mathbf{H}_s or \mathbf{H}_f . If all ratios are unique, then $\mathbf{H}_s = \mathbf{F}_1$ and $\mathbf{H}_f = \mathbf{F}_2$. In that case, the beamformer directly provides the individual signals. The algorithm so far is in essence identical to the second order blind identification (SOBI) algorithm [76], which uses similar properties of covariance matrices for various time lags to separate sources with differing temporal correlation structures.

Now the case to be considered is that both \mathbf{Y}_1 and \mathbf{Y}_2 are contaminated by white noise, and $\mathbf{R}_{n1} = \mathbf{R}_{n2} = \sigma^2 \mathbf{I}$. In this case, the computation of the GSVD of $(\mathbf{Y}_1, \mathbf{Y}_2)$ or the GEVD of $(\mathbf{R}_1, \mathbf{R}_2)$ will lead to the change of \mathbf{F} , being unlike the noiseless case. This is also unlike the case of the SVD or EVD for a single matrix, where the addition of white noise will shift the singular values (eigenvalues) but leave the singular vectors (eigenvectors) intact. For significant noise power, the matrix \mathbf{F} as computed in this way will tend to lose its information. For reasonable noise power, the algorithm seems to perform reasonably well.

With the known noise power σ^2 , it is also possible to compute the GEVD of $(\mathbf{R}_1 - \sigma^2 \mathbf{I}, \mathbf{R}_2 - \sigma^2 \mathbf{I})$. This can give the same correct results as in the noiseless case. Also, computing the GEVD $(\hat{\mathbf{R}}_1 - \sigma^2 \mathbf{I}, \hat{\mathbf{R}}_2 - \sigma^2 \mathbf{I})$ based on the data covariance estimates $\hat{\mathbf{R}}_1 = \frac{1}{N} \mathbf{Y}_1 \mathbf{Y}_1^H$, $\hat{\mathbf{R}}_2 = \frac{1}{N} \mathbf{Y}_2 \mathbf{Y}_2^H$ will lead to reasonable results only if at least one of the resulting matrices is still positive definite (otherwise it may easily happen that the generalized eigenvalues become complex, and meaningless for the applications). However, it is unlikely that the components corresponding to the noise subspace \mathbf{F}_3 will remain positive after noise power shifting. Therefore, the noise subspace must be removed by using a preprocessing step.

The proposed algorithm is given in Table 3.1. In the algorithm, step 1) is a preprocessing (rank reduction, or projecting out the noise subspace) using a threshold ϵ on the singular values of \mathbf{Y} . This matrix contains $2N$ samples. For a sufficiently large number of samples, it could be appropriate to take $\epsilon = \sigma \sqrt{2N}$. But with finite samples, random matrix theory (see Appendix 3.C) has shown that the expected value of the largest singular value due to the noise is $\sigma(\sqrt{2N} + \sqrt{M})$. The variance on the singular values should be considered, so it is appropriate to set

$$\epsilon = \beta_1 \sigma (\sqrt{2N} + \sqrt{M}) \quad (3.23)$$

where the ‘‘Tracy-Widom’’ factor $\beta_1 > 1$ depends on M and N , and is derived in detail in Appendix 3.C, equation (3.62). Typically, this is a small correction

factor, e.g., $\beta_1 < 2$ for reasonable values of N .

After this preprocessing step, the term \mathbf{F}_3 in the subsequent GSVD/GEVD step should be absent as the noise subspace has been removed. This simplifies the algorithm at that stage (note that there is no literature on threshold selection for the noise subspace in GSVD/GEVD, and by this preprocessing step this threshold selection is avoided).

In the algorithm, $\alpha > 1$ is an input parameter that controls the threshold on detecting a target (see Section 3.8).

3.5.2 Scenario 2

In Scenario 2, the data model (3.15) is still used. However, for the target signals, $\mathbf{R}_{s1} \approx \mathbf{R}_{s2}$ is considered, whereas for the interference signals, $\mathbf{R}_{f1} \neq \mathbf{R}_{f2}$ is considered. To formulate this more precisely, a parameter α is again introduced to control how “equal” two values of signal power should be identified as the ones for the same target signal. Specifically, a target signal ought to satisfy

$$\frac{1}{\alpha^2} < \frac{(\mathbf{R}_{s1})_{ii}}{(\mathbf{R}_{s2})_{ii}} < \alpha^2 \quad (3.24)$$

where $\alpha > 1$. Consider first the noiseless case. Then a GEVD of $(\mathbf{R}_1, \mathbf{R}_2)$ gives the decomposition

$$\begin{cases} \mathbf{R}_1 &= \frac{1}{N} \mathbf{F} \mathbf{D} \mathbf{F}^H \\ \mathbf{R}_2 &= \frac{1}{N} \mathbf{F} \mathbf{K} \mathbf{F}^H \end{cases} \quad (3.25)$$

which after sorting can lead to decompositions: $\mathbf{F} = [\mathbf{F}_1 \ \mathbf{F}_2 \ \mathbf{F}_3]$,

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & & \\ & \mathbf{D}_2 & \\ & & \mathbf{D}_3 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_1 & & \\ & \mathbf{K}_2 & \\ & & \mathbf{K}_3 \end{bmatrix} \quad (3.26)$$

such that

$$\begin{aligned} \mathbf{D}_1 &> \epsilon^2 \mathbf{I}, & \mathbf{K}_1 &> \epsilon^2 \mathbf{I} \\ \mathbf{D}_3 &< \epsilon^2 \mathbf{I}, & \mathbf{K}_3 &< \epsilon^2 \mathbf{I} \end{aligned}$$

and

$$\frac{1}{\alpha^2} < \frac{(\mathbf{D}_1)_{ii}}{(\mathbf{K}_1)_{ii}} < \alpha^2 \quad (3.27)$$

Note that $\mathbf{D}_2, \mathbf{K}_2$ collect the diagonal entries that do not satisfy any of the conditions in (3.27). Comparing (3.27) with (3.24), it can be seen that \mathbf{F}_1 corresponds to the target signal subspace and \mathbf{F}_2 to the interference signal subspace. The resulting algorithm is the same as that for Scenario 1 (see Table 3.1), except that now the partitioning (3.26)–(3.27) is used. The parameter α can be used to tune the definition of a target signal, i.e., how “equal” the power in both blocks has to be identified as the ones for the same target signal. Its choice will be discussed in more detail in Section 3.8.

Table 3.1: Source separation algorithm using GSVD or GEVD for Scenario 1 and 2.

-
1. *Rank reduction:* compute the SVD:

$$[\mathbf{Y}_1 \ \mathbf{Y}_2] = [\mathbf{U}_1 \ \mathbf{U}_2] \begin{bmatrix} \boldsymbol{\Sigma}_1 & \\ & \boldsymbol{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix}$$

where $\boldsymbol{\Sigma}_1 > \epsilon \mathbf{I}$ and $\boldsymbol{\Sigma}_2 < \epsilon \mathbf{I}$, and ϵ is a noise threshold, e.g., $\epsilon = \sigma(\sqrt{2N} + \sqrt{M})$ or slightly larger by a factor β as discussed in (3.62).

Then apply a rank and dimension reduction:

$$\tilde{\mathbf{Y}}_1 = \mathbf{U}_1^H \mathbf{Y}_1, \quad \tilde{\mathbf{Y}}_2 = \mathbf{U}_1^H \mathbf{Y}_2$$

2. Estimation of signal subspaces \mathbf{F} :

- *Version using GSVD:* Compute

$$\text{GSVD}(\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2) \Rightarrow \begin{cases} \tilde{\mathbf{Y}}_1 = \mathbf{F}\mathbf{C}\mathbf{U}^H \\ \tilde{\mathbf{Y}}_2 = \mathbf{F}\mathbf{S}\mathbf{V}^H \end{cases}$$

- *Version using GEVD and noise power shifting:* Compute

$$\tilde{\mathbf{R}}_1 = \frac{1}{N} \tilde{\mathbf{Y}}_1 \tilde{\mathbf{Y}}_1^H, \quad \tilde{\mathbf{R}}_2 = \frac{1}{N} \tilde{\mathbf{Y}}_2 \tilde{\mathbf{Y}}_2^H$$

$$\text{GEVD}(\tilde{\mathbf{R}}_1 - \sigma^2 \mathbf{I}, \tilde{\mathbf{R}}_2 - \sigma^2 \mathbf{I}) \Rightarrow \begin{cases} \tilde{\mathbf{R}}_1 - \sigma^2 \mathbf{I} = \frac{1}{N} \mathbf{F}\mathbf{D}\mathbf{F}^H \\ \tilde{\mathbf{R}}_2 - \sigma^2 \mathbf{I} = \frac{1}{N} \mathbf{F}\mathbf{K}\mathbf{F}^H \end{cases}$$

3. Using α , sort the entries of $\mathbf{C}^2, \mathbf{S}^2$ or \mathbf{D}, \mathbf{K} and correspondingly partition \mathbf{F} for Scenario 1 according to (3.18)–(3.19), and for Scenario 2 according to (3.26)–(3.27).

4. The separating beamformer is

$$\mathbf{W}^H = [\mathbf{I} \ \mathbf{0}][\mathbf{F}_1 \ \mathbf{F}_2]^{-1} \mathbf{U}_1^H$$

3.6 Signed URV Decomposition

The GEVD-based algorithm in Table 3.1 is a block-based algorithm, and involves “squaring the data”, which numerically worsens the dynamic range. Generally, it is preferable to work directly with the data, i.e., the GSVD. However, the noise power cannot be subtracted by using the GSVD. Subtracting the noise power by using the GEVD may make the matrices become indefinite. In this section, several algorithms in which the GSVD is replaced by a different subspace estimator will be proposed. They work directly on the data, compensate for the noise power, and are suitable for efficient subspace tracking for sliding-window updates. They are based on the Schur subspace estimator (SSE) introduced in [35, 54]. Some of the definitions in Chapter 2 are repeated here for easy reading.

3.6.1 Definition of the SSE

For two given matrices $\mathbf{N} : M \times N_1$ and $\mathbf{X} : M \times N_2$, with $N_1 + N_2 \geq M$, the SSE is obtained from the decomposition

$$[\mathbf{N} \mid \mathbf{X}] \Theta = [\mathbf{A} \ \mathbf{0} \mid \mathbf{B} \ \mathbf{0}] \quad (3.28)$$

where “|” denotes a matrix partitioning, $\mathbf{A} : M \times d_{\mathbf{A}}$, $\mathbf{B} : M \times d_{\mathbf{B}}$, and Θ is a \mathbf{J} -unitary matrix:

$$\Theta \mathbf{J} \Theta^H = \mathbf{J}, \quad \Theta^H \mathbf{J} \Theta = \mathbf{J}, \quad \mathbf{J} = \left[\begin{array}{c|c} \mathbf{I}_{N_1} & \mathbf{0} \\ \hline \mathbf{0} & -\mathbf{I}_{N_2} \end{array} \right].$$

\mathbf{J} is called a signature matrix. The partitioning of \mathbf{J} (and hence of Θ) follows the partitioning indicated by “|” in (3.28), and then \mathbf{N} and \mathbf{A} have a positive signature, whereas \mathbf{X} and \mathbf{B} have a negative signature. The matrix $[\mathbf{A}, \mathbf{B}]$ is square or tall and has full column rank, and the $\mathbf{0}$ -blocks augment \mathbf{A} and \mathbf{B} to N_1 and N_2 columns, respectively. After “ \mathbf{J} -squaring” the data, it follows from (3.28) that

$$\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H = \mathbf{A}\mathbf{A}^H - \mathbf{B}\mathbf{B}^H$$

and it is seen that \mathbf{A} and \mathbf{B} capture the positive and negative parts of $\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H$, respectively, using factors of minimal dimensions. Assume that there is no “neutral subspace”, i.e., $\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H$ is of full rank; in that case the dimensions of \mathbf{A}, \mathbf{B} satisfy $d_{\mathbf{A}} + d_{\mathbf{B}} = M$, i.e., $[\mathbf{A}, \mathbf{B}]$ is a square matrix.

The decomposition (3.28) always exists [35], but \mathbf{A}, \mathbf{B} and Θ are not unique. The dimensions $d_{\mathbf{A}}$ and $d_{\mathbf{B}}$ are always the same, by Sylvester’s law of inertia. An example of this decomposition can be obtained from the eigenvalue decomposition of $\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H$, i.e.,

$$\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H = [\mathbf{U}_+ \ \mathbf{U}_-] \begin{bmatrix} \Lambda_+ & \\ & -\Lambda_- \end{bmatrix} [\mathbf{U}_+ \ \mathbf{U}_-]^H$$

which provides $\mathbf{A} = \mathbf{U}_+ \mathbf{\Lambda}_+^{1/2}$, $\mathbf{B} = \mathbf{U}_- \mathbf{\Lambda}_-^{1/2}$ satisfying $\mathbf{A} \perp \mathbf{B}$. An option would be the signed Cholesky decomposition,

$$\mathbf{N}\mathbf{N}^H - \mathbf{X}\mathbf{X}^H = \mathbf{R}\mathbf{J}\mathbf{R}^H$$

where $\mathbf{R} = [\mathbf{A}, \mathbf{B}]$ is lower or upper triangular, corresponding to a Hyperbolic QR decomposition in (3.18). However, this decomposition does not always exist [35]: the triangular shape is too restricting. So this is necessary to introduce in (3.28) a QR-factorization of $[\mathbf{A}, \mathbf{B}]$:

$$[\mathbf{A} \ \mathbf{B}] = \mathbf{Q}[\mathbf{R}_A \ \mathbf{R}_B]$$

where \mathbf{Q} is unitary and $[\mathbf{R}_A, \mathbf{R}_B]$ is lower (or upper) triangular, giving rise to a two-sided decomposition of (\mathbf{N}, \mathbf{X}) as

$$\mathbf{Q}^H [\mathbf{N} \ \mathbf{X}] \mathbf{\Theta} = [\mathbf{R}_A \ \mathbf{0} | \mathbf{R}_B \ \mathbf{0}] \quad (3.29)$$

that always exists. This “hyperbolic URV” decomposition is reminiscent of the URV [77, 78], except that it involves a \mathbf{J} -unitary matrix $\mathbf{\Theta}$. Such matrices are related to hyperbolic rotations that can have large norms and thus may lead to numerical problems if not treated carefully.

Some of the freedom in the decomposition (3.29) can be restricted by putting further constraints on $\mathbf{\Theta}$. One restriction is given by the so-called SSE-2 [35], which poses a rather technical structural constraint on $\mathbf{\Theta}$ that induces favorable properties, in particular

$$\mathbf{A}\mathbf{A}^H \leq \mathbf{N}\mathbf{N}^H, \quad \mathbf{B}\mathbf{B}^H \leq \mathbf{X}\mathbf{X}^H. \quad (3.30)$$

This restriction prevents the introduction of large common components in \mathbf{A} and \mathbf{B} that cancel each other in $\mathbf{A}\mathbf{A}^H - \mathbf{B}\mathbf{B}^H$. A numerically stable algorithm to compute the SSE-2 was recently proposed in [14], simply called the “signed URV” algorithm (See Chapter 2). The algorithm is non-iterative and easily updated for new columns of \mathbf{N} and \mathbf{X} , with complexity similar to that of a QR-update. Dnndating (removing a column of \mathbf{N} or \mathbf{X}) is elegantly achieved by updating \mathbf{X} or \mathbf{N} , respectively. This makes SURV very suitable for sliding window tracking of subspaces.

3.6.2 Properties

Here, some properties of the SSE-2 (or SURV algorithm) are listed to reflect its potential for subspace tracking. Consider $\mathbf{N} = \epsilon \mathbf{I}$, where ϵ is a threshold, and introduce the SVD of \mathbf{X} as

$$\mathbf{X} = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^H + \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^H$$

where $\Sigma_1 > \epsilon \mathbf{I}$ and $\Sigma_2 < \epsilon \mathbf{I}$. Assume that Σ_1 has size $d \times d$ (i.e., \mathbf{X} has d singular values larger than ϵ ; here assume none are equal to ϵ). Then the SSE is obtained as

$$[\epsilon \mathbf{I} \mid \mathbf{X}] \Theta = [\mathbf{A} \ \mathbf{0} \mid \mathbf{B} \ \mathbf{0}] \quad (3.31)$$

which corresponds to

$$\mathbf{X} \mathbf{X}^H - \epsilon^2 \mathbf{I} = \mathbf{B} \mathbf{B}^H - \mathbf{A} \mathbf{A}^H. \quad (3.32)$$

It is seen that the spectrum of $\mathbf{X} \mathbf{X}^H$ is shifted by ϵ^2 , which will introduce $M - d$ negative eigenvalues. Thus, \mathbf{B} has d columns and \mathbf{A} has $M - d$ columns. By means of this decomposition, it was shown in [35] that $(\mathbf{A}, \mathbf{B}, \Theta)$ can be used to parametrize all rank- d approximants $\hat{\mathbf{X}}$ such that

$$\|\mathbf{X} - \hat{\mathbf{X}}\| < \epsilon. \quad (3.33)$$

In particular, it was shown that the column span of any such $\hat{\mathbf{X}}$ is parametrized as $\text{ran}(\mathbf{B}')$ with

$$\mathbf{B}' = \mathbf{B} - \mathbf{A} \mathbf{M}, \quad \|\mathbf{M}\| < 1 \quad (3.34)$$

where the matrix $\mathbf{M} : d_{\mathbf{A}} \times d_{\mathbf{B}}$ contains free parameters, reflecting the non-uniqueness in the decomposition (3.29).

It was shown in [14] that the TSVD (3.8) is a special case of an approximant satisfying (3.33), corresponding to an SSE-2 decomposition with

$$\mathbf{B} = \mathbf{U}_1 (\Sigma_1^2 - \epsilon^2 \mathbf{I})^{1/2}, \quad \mathbf{A} = \mathbf{U}_2 (\epsilon^2 \mathbf{I} - \Sigma_2^2)^{1/2} \quad (3.35)$$

and a specific Θ . In this case the column span of \mathbf{B} is equal to the principal subspace obtained from the SVD of \mathbf{X} .

More in general, consider matrices \mathbf{N} , \mathbf{X} , each with N columns, and the SSE-2 decomposition $[\mathbf{N} \mid \mathbf{X}] \Theta = [\mathbf{A} \ \mathbf{0} \mid \mathbf{B} \ \mathbf{0}]$ such that $\mathbf{N} \mathbf{N}^H = \epsilon^2 \mathbf{R}_n$. Then a direct generalization of (3.33) shows that all low-rank approximants $\hat{\mathbf{X}}$ such that

$$\|\mathbf{R}_n^{-1/2} (\mathbf{X} - \hat{\mathbf{X}})\| < \epsilon$$

have a column span parametrized by $\mathbf{B} - \mathbf{A} \mathbf{M}$, $\|\mathbf{M}\| < 1$.

Now the relation of the SSE-2 in (3.28) to the GSVD can be described as follows. The GSVD is firstly constructed as

$$\begin{cases} \mathbf{N} = \mathbf{F} \mathbf{C} \mathbf{U}^H = [\mathbf{F}_1 & \mathbf{F}_2] \begin{bmatrix} \mathbf{C}_1 & \\ & \mathbf{C}_2 \end{bmatrix} \mathbf{U}^H \\ \mathbf{X} = \mathbf{F} \mathbf{S} \mathbf{V}^H = [\mathbf{F}_1 & \mathbf{F}_2] \begin{bmatrix} \mathbf{S}_1 & \\ & \mathbf{S}_2 \end{bmatrix} \mathbf{V}^H \end{cases} \quad (3.36)$$

where the sorting and partitioning is such that $\mathbf{C}_1 > \mathbf{S}_1$, $\mathbf{C}_2 < \mathbf{S}_2$ (for concise notation, here assume there is no common null space: \mathbf{F}_3 is missing). Equivalently,

the GEVD is obtained as

$$\begin{cases} \mathbf{NN}^H = \mathbf{FDF}^H = [\mathbf{F}_1 & \mathbf{F}_2] \begin{bmatrix} \mathbf{D}_1 & \\ & \mathbf{D}_2 \end{bmatrix} [\mathbf{F}_1 & \mathbf{F}_2]^H \\ \mathbf{XX}^H = \mathbf{FKF}^H = [\mathbf{F}_1 & \mathbf{F}_2] \begin{bmatrix} \mathbf{K}_1 & \\ & \mathbf{K}_2 \end{bmatrix} [\mathbf{F}_1 & \mathbf{F}_2]^H \end{cases}$$

with a partitioning such that $\mathbf{D}_1 > \mathbf{K}_1$, $\mathbf{D}_2 < \mathbf{K}_2$. Then

$$\begin{aligned} \mathbf{NN}^H - \mathbf{XX}^H &= \mathbf{F}(\mathbf{D} - \mathbf{K})\mathbf{F}^H \\ &= \mathbf{F}_1(\mathbf{D}_1 - \mathbf{K}_1)\mathbf{F}_1^H - \mathbf{F}_2(\mathbf{K}_2 - \mathbf{D}_2)\mathbf{F}_2^H. \end{aligned} \quad (3.37)$$

But squaring (3.28) provides

$$\mathbf{NN}^H - \mathbf{XX}^H = \mathbf{AA}^H - \mathbf{BB}^H$$

then it follows that

$$\begin{aligned} \mathbf{A} &= \mathbf{F}_1(\mathbf{D}_1 - \mathbf{K}_1)^{1/2} = \mathbf{F}_1(\mathbf{C}_1^2 - \mathbf{S}_1^2)^{1/2} \\ \mathbf{B} &= \mathbf{F}_2(\mathbf{K}_2 - \mathbf{D}_2)^{1/2} = \mathbf{F}_2(\mathbf{S}_2^2 - \mathbf{C}_2^2)^{1/2} \end{aligned} \quad (3.38)$$

as one possible solution to (3.28)—recall that this decomposition is not unique. This solution provides, in particular, $\text{ran}(\mathbf{A}) = \text{ran}(\mathbf{F}_1)$, $\text{ran}(\mathbf{B}) = \text{ran}(\mathbf{F}_2)$. Thus, the GSVD provides a special case of the SSE-2 decomposition.

Then it has been shown that the GSVD provides a special case of an SSE-2, where the pair $(\mathbf{F}_1, \mathbf{F}_2)$ in the GSVD is directly related to the pair (\mathbf{A}, \mathbf{B}) in the SSE-2. Conversely, a suggestion will be put forward in the next section about replacing the GSVD by the SURV algorithm in the source separation algorithm, with the hope that the resulting subspace estimates are sufficiently close to that of the GSVD. Simulations to confirm this conclusion are in Section 3.9. The advantage of using SURV is that it is a simple non-iterative algorithm which is easily updated and thus enables sliding window tracking.

3.7 Source Separation Using SURV

3.7.1 Application to Scenario 1

In Scenario 1, a pair of data matrices $(\mathbf{Y}_1, \mathbf{Y}_2)$ are used, each with N samples, with sample covariance matrices $(\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2)$ converging to covariance matrices $(\mathbf{R}_1, \mathbf{R}_2)$ that satisfy the data model (3.15). Instead of the GSVD or GEVD, the SURV is performed

$$[\mathbf{Y}_1 \mid \alpha \mathbf{Y}_2] \Theta = [\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \quad (3.39)$$

where the factor $\alpha > 1$ controls the signal classification (see Section 3.8). Consider first the noiseless case, and assume that $[\mathbf{H}_s, \mathbf{H}_f]$ is square (M columns) and of full rank (this can be ensured by an initial rank reduction). Squaring (3.39) gives

$$\hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 = \frac{1}{N} (\mathbf{AA}^H - \mathbf{BB}^H). \quad (3.40)$$

The GSVD or GEVD provides a special case of this decomposition (cf. (3.37)–(3.38)), i.e.,

$$\hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 = \frac{1}{N} \left(\mathbf{F}_1 (\mathbf{D}_1 - \alpha^2 \mathbf{K}_1) \mathbf{F}_1^H - \mathbf{F}_2 (\alpha^2 \mathbf{K}_2 - \mathbf{D}_2) \mathbf{F}_2^H \right)$$

(the partitioning of \mathbf{F} is such that $\mathbf{D}_1 - \alpha^2 \mathbf{K}_1 > 0$ and $\alpha^2 \mathbf{K}_2 - \mathbf{D}_2 > 0$) whereas the covariance model of Scenario 1 provides

$$\mathbf{R}_1 - \alpha^2 \mathbf{R}_2 = \mathbf{H}_s (\mathbf{R}_{s1} - \alpha^2 \mathbf{R}_{s2}) \mathbf{H}_s^H - \mathbf{H}_f (\alpha^2 \mathbf{R}_{f2} - \mathbf{R}_{f1}) \mathbf{H}_f^H.$$

Asymptotically (for sufficiently large N such that $\hat{\mathbf{R}}_1 \rightarrow \mathbf{R}_1$ and $\hat{\mathbf{R}}_2 \rightarrow \mathbf{R}_2$), it follows that $\text{ran}(\mathbf{A}) \approx \text{ran}(\mathbf{F}_1) = \text{ran}(\mathbf{H}_s)$ and $\text{ran}(\mathbf{B}) \approx \text{ran}(\mathbf{F}_2) = \text{ran}(\mathbf{H}_f)$. Generally, for finite N , it follows that

$$\begin{cases} \text{ran}(\mathbf{A}) \approx \text{ran}(\mathbf{F}_1) \approx \text{ran}(\mathbf{H}_s) \\ \text{ran}(\mathbf{B}) \approx \text{ran}(\mathbf{F}_2) \approx \text{ran}(\mathbf{H}_f) \end{cases}$$

Thus, the decomposition SURV($\mathbf{Y}_1, \alpha \mathbf{Y}_2$) directly gives estimates of $(\mathbf{F}_1, \mathbf{F}_2)$ that are used to construct a separating beamformer \mathbf{W} .

Consider now the case with noise. The GSVD of $(\mathbf{Y}_1, \mathbf{Y}_2)$, or equivalently the GEVD of $(\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2)$, and its data model are

$$\begin{aligned} \hat{\mathbf{R}}_1 &= \frac{1}{N} \mathbf{FDF}^H, & \mathbf{R}_1 &= \mathbf{H}_s \mathbf{R}_{s1} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f1} \mathbf{H}_f^H + \mathbf{R}_{n1} \\ \hat{\mathbf{R}}_2 &= \frac{1}{N} \mathbf{FKF}^H, & \mathbf{R}_2 &= \mathbf{H}_s \mathbf{R}_{s2} \mathbf{H}_s^H + \mathbf{H}_f \mathbf{R}_{f2} \mathbf{H}_f^H + \mathbf{R}_{n2} \end{aligned}$$

(the parameter α defines the partitioning of \mathbf{F}), and

$$\begin{aligned} \mathbf{R}_1 - \alpha^2 \mathbf{R}_2 &= \mathbf{H}_s (\mathbf{R}_{s1} - \alpha^2 \mathbf{R}_{s2}) \mathbf{H}_s^H \\ &\quad - \mathbf{H}_f (\alpha^2 \mathbf{R}_{f2} - \mathbf{R}_{f1}) \mathbf{H}_f^H + (\mathbf{R}_{n1} - \alpha^2 \mathbf{R}_{n2}). \end{aligned} \quad (3.41)$$

As mentioned before, the presence of \mathbf{R}_{n1} and \mathbf{R}_{n2} will cause $\text{ran}(\mathbf{F}_1)$ to deviate from $\text{ran}(\mathbf{H}_s)$ and $\text{ran}(\mathbf{F}_2)$ from $\text{ran}(\mathbf{H}_f)$ in the GSVD/GEVD. However, it can be found that by looking at $\mathbf{R}_1 - \mathbf{R}_2$ when $\alpha = 1$ that the two noise covariance matrices \mathbf{R}_{n1} and \mathbf{R}_{n2} will cancel each other in (3.41): automatic noise power compensation! Thus, it is possible that SURV($\mathbf{Y}_1, \alpha \mathbf{Y}_2$) produces (asymptotically) the same result as in the noiseless case. In reality, the SSE-2 is not unique and therefore one can encounter anything between the noiseless result and the GSVD with noise.

For $\alpha > 1$, the noise term $\mathbf{R}_{n1} - \alpha^2 \mathbf{R}_{n2}$ in (3.41) is not zero. Although the SURV decomposition gives more reasonable results, it is still possible to further improve results by means of the compensation of the noise power by adding the term $\tau \mathbf{I}$ to $\hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2$. The resulting data model for $\mathbf{R}_1 - \alpha^2 \mathbf{R}_2 + \tau \mathbf{I}$ has the noise power term $\mathbf{R}_{n1} - \alpha^2 \mathbf{R}_{n2} + \tau \mathbf{I}$. It is clear that the noise power cancellation occurs when

$$\tau = \sigma^2 (\alpha^2 - 1), \quad \mathbf{R}_{n1} = \mathbf{R}_{n2} = \sigma^2 \mathbf{I} \quad (3.42)$$

Note that $\tau > 0$. Now the SURV is performed

$$[\gamma \mathbf{I} \quad \mathbf{Y}_1 \mid \alpha \mathbf{Y}_2] \Theta = [\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \quad (3.43)$$

with $\gamma = \sqrt{N\tau}$, and then it follows that

$$\hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 + \frac{\gamma^2}{N} \mathbf{I} = \hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 + \tau \mathbf{I} = \frac{1}{N} (\mathbf{A} \mathbf{A}^H - \mathbf{B} \mathbf{B}^H)$$

resulting in the noise power compensation, asymptotically.

Several refinements are in order. For finite samples, the resulting noise power $\hat{\mathbf{R}}_{n1} - \alpha^2 \hat{\mathbf{R}}_{n2} + \tau \mathbf{I}$ will not be exactly zero, but some indefinite matrix. For optimal accuracy of \mathbf{A} , it is better to ensure that this matrix is small but negative definite. As shown in Appendix 3.D, this is achieved by setting

$$\tau = \sigma^2 \left(\alpha^2 \left(1 - \frac{\sqrt{M}}{\sqrt{N}} \right)^2 - \left(1 + \frac{\sqrt{M}}{\sqrt{N}} \right)^2 \right) \quad (3.44)$$

which replaces (3.42) by a slightly smaller τ . If $\tau > 0$, then the SURV in (3.43) can be directly performed with $\gamma = \sqrt{N\tau}$; otherwise, the following SURV can be performed with $\gamma = \sqrt{N|\tau|}$:

$$[\mathbf{Y}_1 \mid \gamma \mathbf{I} \quad \alpha \mathbf{Y}_2] \Theta = [\mathbf{A} \quad \mathbf{0} \mid \mathbf{B} \quad \mathbf{0}] \quad (3.45)$$

After this equation is squared, it follows that

$$\hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 - \frac{\gamma^2}{N} \mathbf{I} = \hat{\mathbf{R}}_1 - \alpha^2 \hat{\mathbf{R}}_2 + \tau \mathbf{I} = \frac{1}{N} (\mathbf{A} \mathbf{A}^H - \mathbf{B} \mathbf{B}^H)$$

which is the required expression for noise compensation. Appendix 3.D also lists refinements for matrices $\mathbf{Y}_1, \mathbf{Y}_2$ of unequal dimensions, and a more accurate result using Tracy-Widom factors.

In the original GSVD/GEVD algorithm in Table 3.1, the first step required an initial SVD of $[\mathbf{Y}_1, \mathbf{Y}_2]$. This SVD can be replaced by an SURV as this case is similar to what is shown at the beginning of Section 3.6.2. Here, the threshold requirement is $\epsilon > \sigma(\sqrt{2N} + \sqrt{M})$, (taking into account that there are $2N$ samples), and a Tracy-Widom factor β_1 can be used to make this more accurate, as discussed in Appendix 3.C, Eq. (3.62). After rank reduction, the algorithm continues with the reduced matrices $(\tilde{\mathbf{Y}}_1, \tilde{\mathbf{Y}}_2)$. However, this initial rank reduction can also be omitted, as discussed below.

Both versions of the algorithm are shown in Table 3.2. In step 1) of the version with rank reduction, the fact that $[\mathbf{A}, \mathbf{B}] = \mathbf{U}[\mathbf{L}_A, \mathbf{L}_B]$ and $[\mathbf{L}_A, \mathbf{L}_B]$ is lower triangular is used: in that case \mathbf{U}_B is an orthogonal basis of the column span of \mathbf{B} . The rank reduction is achieved by premultiplying with \mathbf{U}_B^H .

Step 2) is the essence of the algorithm. It is needed to compute the SURV decomposition (3.48) or (3.49), and to define $[\mathbf{A}, \mathbf{B}] = \mathbf{Q}[\mathbf{R}_A, \mathbf{R}_B]$. Using the fact

Table 3.2: Source separation algorithms for Scenario 1 using SURV (with or without rank reduction)

SURV1-S1—Version 1 for Scenario 1, with initial rank reduction:

1. Rank reduction: compute the SURV:

$$\mathbf{U}^H[\epsilon\mathbf{I} \mid \mathbf{Y}_1 \ \mathbf{Y}_2]\boldsymbol{\Theta} = [\mathbf{L}_A \ \mathbf{0} \mid \mathbf{L}_B \ \mathbf{0}] \quad (3.46)$$

where $\epsilon > \beta_1\sigma(\sqrt{2N} + \sqrt{M})$ is a noise threshold, $\mathbf{U} = [\mathbf{U}_A \ \mathbf{U}_B]$ is unitary, and $[\mathbf{L}_A \ \mathbf{L}_B]$ is lower triangular. Apply a rank and dimension reduction:

$$\tilde{\mathbf{Y}}_1 = \mathbf{U}_B^H \mathbf{Y}_1, \quad \tilde{\mathbf{Y}}_2 = \mathbf{U}_B^H \mathbf{Y}_2 \quad (3.47)$$

2. Compute τ as in (3.44). Set $\gamma = \sqrt{N|\tau|}$. If $\tau > 0$, then compute the SURV

$$\mathbf{Q}^H[\gamma\mathbf{I} \ \tilde{\mathbf{Y}}_1 \mid \alpha\tilde{\mathbf{Y}}_2]\boldsymbol{\Theta} = [\mathbf{R}_A \ \mathbf{0} \mid \mathbf{R}_B \ \mathbf{0}] \quad (3.48)$$

else compute the SURV

$$\mathbf{Q}^H[\tilde{\mathbf{Y}}_1 \mid \gamma\mathbf{I} \ \alpha\tilde{\mathbf{Y}}_2]\boldsymbol{\Theta} = [\mathbf{R}_A \ \mathbf{0} \mid \mathbf{R}_B \ \mathbf{0}] \quad (3.49)$$

$\mathbf{Q} = [\mathbf{Q}_A \ \mathbf{Q}_B]$ is unitary, and $[\mathbf{R}_A \ \mathbf{R}_B]$ is lower triangular.

3. The separating beamformer is given by

$$\mathbf{W}^H = \mathbf{Q}_A^H \mathbf{U}_B^H \quad (3.50)$$

SURV2-S1—Version 2 for Scenario 1, without rank reduction:

1. Compute τ as in (3.44). Set $\gamma = \sqrt{N|\tau|}$. Compute one SURV:

$$\text{(if } \tau > 0) \quad \mathbf{Q}^H[\gamma\mathbf{I} \ \tilde{\mathbf{Y}}_1 \mid \alpha\tilde{\mathbf{Y}}_2]\boldsymbol{\Theta} = [\mathbf{R}_A \ \mathbf{0} \mid \mathbf{R}_B \ \mathbf{0}] \quad (3.51)$$

$$\text{(if } \tau < 0) \quad \mathbf{Q}^H[\tilde{\mathbf{Y}}_1 \mid \gamma\mathbf{I} \ \alpha\tilde{\mathbf{Y}}_2]\boldsymbol{\Theta} = [\mathbf{R}_A \ \mathbf{0} \mid \mathbf{R}_B \ \mathbf{0}] \quad (3.52)$$

2. The separating beamformer is given by

$$\mathbf{W}^H = \mathbf{Q}_A^H \quad (3.53)$$

that $[\mathbf{R}_A, \mathbf{R}_B]$ is lower triangular, it can be found that \mathbf{Q}_B is an orthogonal basis of the estimated interference subspace $\tilde{\mathcal{H}}_f = \text{ran}(\mathbf{B})$. Further, \mathbf{Q}_A is orthogonal to \mathbf{Q}_B .

At this point, the separating beamformer (acting on the rank-reduced $\tilde{\mathbf{Y}}$) is

$$\tilde{\mathbf{W}}^H = [\mathbf{I} \quad \mathbf{0}][\mathbf{A} \quad \mathbf{B}]^\dagger = (\mathbf{A}^H \mathbf{P}_B^\perp \mathbf{A})^{-1} \mathbf{A}^H \mathbf{P}_B^\perp.$$

The factor $(\mathbf{A}^H \mathbf{P}_B^\perp \mathbf{A})^{-1}$ is invertible and can be omitted without changing the separation by the beamformer. Since $\mathbf{A} = \mathbf{Q} \mathbf{R}_A$ and $\mathbf{P}_B^\perp = \mathbf{Q}_A \mathbf{Q}_A^H$, it follows that

$$\mathbf{A}^H \mathbf{P}_B^\perp = \mathbf{R}_A^H \mathbf{Q}^H \mathbf{Q}_A \mathbf{Q}_A^H = \mathbf{R}_A^H \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{Q}_A^H = \mathbf{R}'_A \mathbf{Q}_A^H. \quad (3.54)$$

\mathbf{R}'_A is square and invertible, and can be dropped. Therefore, if $\tilde{\mathbf{W}} = \mathbf{Q}_A$ is selected, then the overall beamformer is $\mathbf{W}^H = \mathbf{Q}_A^H \mathbf{U}_B^H$.

In the version of the algorithm without rank reduction, the rank reduction step is gone before going to the SURV of $(\mathbf{Y}_1, \mathbf{Y}_2)$. This version applies proper noise power compensation to a small but negative amount. This ensures that the noise subspace (which is present if $[\mathbf{H}_s, \mathbf{H}_f]$ has $d < M$ columns) is part of the negative subspace. The column span of \mathbf{Q}_A is orthogonal to the estimated interference subspace \mathcal{H}_f and the noise subspace. The required beamformer is now simply $\mathbf{W}^H = \mathbf{Q}_A^H$. Thus, the algorithm becomes particularly simple, and is suitable for tracking as the SURV is easily used in a sliding window update setting [14].

3.7.2 Application to Scenario 2

In Scenario 2, it is assumed that $[\mathbf{H}_s, \mathbf{H}_f]$ is square and of full rank, then the data matrices $(\mathbf{Y}_1, \mathbf{Y}_2)$ have sample covariances $(\hat{\mathbf{R}}_1, \hat{\mathbf{R}}_2)$ and corresponding covariance model $(\mathbf{R}_1, \mathbf{R}_2)$. The GEVD in (3.25) is suitable for the latter, with the sorting of the eigenvalues defined by (3.27). Specifically, for a choice of $\alpha > 1$ that determines the signal classification (see Sec. 3.8), the eigenvalues of the target signals satisfy

$$\frac{1}{\alpha^2} < \frac{(\mathbf{R}_{s1})_{ii}}{(\mathbf{R}_{s2})_{ii}} < \alpha^2 \quad \Leftrightarrow \quad \frac{1}{\alpha^2} < \frac{(\mathbf{D}_1)_{ii}}{(\mathbf{K}_1)_{ii}} < \alpha^2.$$

For the target signals, the following two conditions need to be satisfied simultaneously:

$$\begin{cases} \alpha^2 \mathbf{R}_{s1} - \mathbf{R}_{s2} > \mathbf{0} & \Leftrightarrow & \alpha^2 \mathbf{D}_1 - \mathbf{K}_1 > \mathbf{0} \\ \mathbf{R}_{s1} - \alpha^2 \mathbf{R}_{s2} < \mathbf{0} & \Leftrightarrow & \mathbf{D}_1 - \alpha^2 \mathbf{K}_1 < \mathbf{0} \end{cases} \quad (3.55)$$

Each interference signal satisfies only one of the conditions, but not both, that is, the interference signals can be split into two groups:

$$\mathbf{R}_{f1} = \begin{bmatrix} \mathbf{R}_{f1,1} & \\ & \mathbf{R}_{f1,2} \end{bmatrix}, \quad \mathbf{R}_{f2} = \begin{bmatrix} \mathbf{R}_{f2,1} & \\ & \mathbf{R}_{f2,2} \end{bmatrix}$$

Table 3.3: *Source separation algorithm for Scenario 2 using SURV (with initial rank reduction)*

SURV1-S2—Version 1 for Scenario 2, with rank reduction:

1. *Rank reduction:* compute the SURV:

$$\mathbf{U}^H [\epsilon \mathbf{I} \mid \mathbf{Y}_1 \ \mathbf{Y}_2] \Theta = [\mathbf{L}_A \ \mathbf{0} \mid \mathbf{L}_B \ \mathbf{0}]$$

where $\epsilon > \beta\sigma(\sqrt{2N} + \sqrt{M})$ is a noise threshold, $\mathbf{U} = [\mathbf{U}_A \ \mathbf{U}_B]$ is unitary, and $[\mathbf{L}_A \ \mathbf{L}_B]$ is lower triangular. Apply a rank and dimension reduction:

$$\tilde{\mathbf{Y}}_1 = \mathbf{U}_B^H \mathbf{Y}_1, \quad \tilde{\mathbf{Y}}_2 = \mathbf{U}_B^H \mathbf{Y}_2$$

2. Compute the two SURVs:

$$\mathbf{Q}_1^H [\alpha \tilde{\mathbf{Y}}_1 \mid \tilde{\mathbf{Y}}_2 \ \gamma \mathbf{I}] \Theta = [\mathbf{R}_{A_1} \ \mathbf{0} \mid \mathbf{R}_{B_1} \ \mathbf{0}]$$

$$\mathbf{Q}_2^H [\alpha \tilde{\mathbf{Y}}_2 \mid \tilde{\mathbf{Y}}_1 \ \gamma \mathbf{I}] \Theta = [\mathbf{R}_{A_2} \ \mathbf{0} \mid \mathbf{R}_{B_2} \ \mathbf{0}]$$

where γ is given in (3.56), $\mathbf{Q}_1 = [\mathbf{Q}_{A_1} \ \mathbf{Q}_{B_1}]$ and $\mathbf{Q}_2 = [\mathbf{Q}_{A_2} \ \mathbf{Q}_{B_2}]$ are unitary, and $[\mathbf{R}_{A_1} \ \mathbf{R}_{B_1}]$ and $[\mathbf{R}_{A_2} \ \mathbf{R}_{B_2}]$ are lower triangular.

3. Compute an orthonormal basis of the common subspace \mathbf{U}_c of \mathbf{Q}_{A_1} and \mathbf{Q}_{A_2} via subspace intersection, using an SVD as in (3.11).
4. The separating beamformer is given by

$$\mathbf{W}^H = \mathbf{U}_c^H \mathbf{U}_B^H$$

with

$$\begin{cases} \alpha^2 \mathbf{R}_{f1,1} - \mathbf{R}_{f2,1} > 0 & \text{and } \mathbf{R}_{f1,1} - \alpha^2 \mathbf{R}_{f2,1} > 0 \\ \alpha^2 \mathbf{R}_{f1,2} - \mathbf{R}_{f2,2} < 0 & \text{and } \mathbf{R}_{f1,2} - \alpha^2 \mathbf{R}_{f2,2} < 0 \end{cases} .$$

The two SURVs will be used to compute subspaces corresponding to each condition, and then use an SVD to intersect them, as stated below.

The first SURV is

$$[\alpha \mathbf{Y}_1 \mid \mathbf{Y}_2 \quad \gamma \mathbf{I}] \Theta_1 = [\mathbf{A}_1 \quad \mathbf{0} \mid \mathbf{B}_1 \quad \mathbf{0}]$$

where $\gamma \mathbf{I}$ implements a noise power compensation similar to Scenario 1. Appendix 3.D derives a suitable value for γ as

$$\gamma = \sqrt{N\tau}, \quad \tau = \sigma^2 \left[\alpha^2 \left(1 + \frac{\sqrt{M}}{\sqrt{N}} \right)^2 - \left(1 - \frac{\sqrt{M}}{\sqrt{N}} \right)^2 \right] > 0. \quad (3.56)$$

This decomposition provides

$$\alpha^2 \hat{\mathbf{R}}_1 - \hat{\mathbf{R}}_2 - \frac{\gamma^2}{N} \mathbf{I} = \frac{1}{N} (\mathbf{A}_1 \mathbf{A}_1^H - \mathbf{B}_1 \mathbf{B}_1^H)$$

whereas the data model is

$$\begin{aligned} \alpha^2 \mathbf{R}_1 - \mathbf{R}_2 - \frac{\gamma^2}{N} \mathbf{I} &= \mathbf{H}_s (\alpha^2 \mathbf{R}_{s1} - \mathbf{R}_{s2}) \mathbf{H}_s^H \\ &+ \mathbf{H}_f (\alpha^2 \mathbf{R}_{f1} - \mathbf{R}_{f2}) \mathbf{H}_s^H - \left(\frac{\gamma^2}{N} - \sigma^2 (\alpha^2 - 1) \right) \mathbf{I}. \end{aligned}$$

By the definition of a target signal (the first condition in (3.55)), it is known that $\alpha^2 \mathbf{R}_{s1} - \mathbf{R}_{s2} > 0$ for the target signals, but $\alpha^2 \mathbf{R}_{f1} - \mathbf{R}_{f2}$ is indefinite: the “first” group of interference signals gives a positive power difference. Hence the dimension of the positive subspace can be larger than d_s . The choice of γ in (3.56) is such that $\frac{\gamma^2}{N} - \sigma^2 (\alpha^2 - 1)$ is small but positive, even for finite samples. In that case, \mathbf{A}_1 is an estimate of a basis of the subspace spanned by both \mathbf{H}_s and the part of \mathbf{H}_f corresponding to the first group of interference signals.

The second SURV is

$$[\alpha \mathbf{Y}_2 \mid \mathbf{Y}_1 \quad \gamma \mathbf{I}] \Theta_2 = [\mathbf{A}_2 \quad \mathbf{0} \mid \mathbf{B}_2 \quad \mathbf{0}]. \quad (3.57)$$

where α and γ are the same as before. This provides

$$\alpha^2 \hat{\mathbf{R}}_2 - \hat{\mathbf{R}}_1 - \frac{\gamma^2}{N} \mathbf{I} = \frac{1}{N} (\mathbf{A}_2 \mathbf{A}_2^H - \mathbf{B}_2 \mathbf{B}_2^H)$$

whereas the data model is

$$\begin{aligned} \alpha^2 \mathbf{R}_2 - \mathbf{R}_1 - \frac{\gamma^2}{N} \mathbf{I} &= \mathbf{H}_s (\alpha^2 \mathbf{R}_{s2} - \mathbf{R}_{s1}) \mathbf{H}_s^H \\ &+ \mathbf{H}_f (\alpha^2 \mathbf{R}_{f2} - \mathbf{R}_{f1}) \mathbf{H}_s^H - \left(\frac{\gamma^2}{N} - \sigma^2 (\alpha^2 - 1) \right) \mathbf{I}. \end{aligned}$$

By the definition (the second condition in (3.55)), it is known that $\alpha^2 \mathbf{R}_{s2} - \mathbf{R}_{s1} > 0$ but $\alpha^2 \mathbf{R}_{f2} - \mathbf{R}_{f1}$ is indefinite: the “second” group of interference signals gives

a positive difference. As before, γ is chosen to make $\frac{\gamma^2}{N} - \sigma^2(\alpha^2 - 1)$ small but positive. In that case, \mathbf{A}_2 is an estimate for a basis of the subspace spanned by both \mathbf{H}_s and part of \mathbf{H}_f corresponding to the second group of interference signals.

The subspace \mathcal{H}_s is obtained from the intersection of the subspaces spanned by $\mathbf{P}_{\mathbf{B}_1}^\perp \mathbf{A}_1$ and $\mathbf{P}_{\mathbf{B}_2}^\perp \mathbf{A}_2$. The computation of the SURVs will produce orthogonal bases of these subspaces as $\mathbf{Q}_{\mathbf{A}_1}$ and $\mathbf{Q}_{\mathbf{A}_2}$, respectively. Thus, it is possible to compute the intersection from an SVD of $[\mathbf{Q}_{\mathbf{A}_1}, \mathbf{Q}_{\mathbf{A}_2}]$, as discussed in Section 3.3.5. A threshold $1 < \xi < \sqrt{2}$ is required. An orthonormal basis \mathbf{U}_c of the intersection is produced by the SVD, and the separating beamformer is given by

$$\mathbf{W}^H = \mathbf{U}_c^H.$$

In the version of the algorithm with rank reduction, $(\mathbf{Y}_1, \alpha \mathbf{Y}_2)$ is replaced by the rank-reduced $(\tilde{\mathbf{Y}}_1, \alpha \tilde{\mathbf{Y}}_2)$. The data model for these matrices involves $\tilde{\mathbf{H}} = \mathbf{U}_{\mathbf{B}}^H [\mathbf{H}_s, \mathbf{H}_f]$ which is square and of full rank. The resulting separating beamformer is now

$$\mathbf{W}^H = \mathbf{U}_c^H \mathbf{U}_{\mathbf{B}}^H.$$

The version of the algorithm without rank reduction in Scenario 2 is not provided here because its performance seems not to be satisfactory.

3.8 Generalization of Algorithms for Nonstationary Data Model

The algorithms for a block-stationary data model $(\mathbf{Y}_1, \mathbf{Y}_2)$ were derived before, and applied to nonstationary data $(\mathbf{X}_1, \mathbf{X}_2)$ consisting of intermittent signals. In general, \mathbf{X}_1 and \mathbf{X}_2 can have an unequal number of samples N_1 and N_2 . In this section, the works to be done include the validity of the algorithms for the more general nonstationary case and the discussion of selecting the detection parameter α .

As shown in Appendix 3.E, for $(\mathbf{X}_1, \mathbf{X}_2)$, the sample correlation matrix converges faster to its model than it would for corresponding stationary signals $(\mathbf{Y}_1, \mathbf{Y}_2)$. This was shown for the correlation matrices of two signals that are partly zero, but the same holds for the more elaborate model $(\mathbf{X}_1, \mathbf{X}_2)$ of intermittent signals in noise. Thus, the derived algorithms are valid and perhaps have better performance for finite samples because of the faster convergence.

For the proper selection of α , consider first Scenario 1 and define $n_1^{(k)}$ and $n_2^{(k)}$ to be the number of nonzero samples of the k th source (packet) within \mathbf{X}_1 and \mathbf{X}_2 , respectively. In Scenario 1, a target signal is selected if $\mathbf{R}_{s1} > \alpha^2 \mathbf{R}_{s2}$. In the case of independent sources, \mathbf{R}_{s1} and \mathbf{R}_{s2} are diagonal, and it can be seen that the power (amplitude) of a source does not play a role, but only the ratio of its number of nonzero samples in block \mathbf{X}_1 and \mathbf{X}_2 . In particular, the k th source is considered as a target signal if $n_1^{(k)} > \alpha^2 n_2^{(k)}$. To design α , first choose a parameter n_1 that defines a target packet having more than n_1 nonzero samples

in \mathbf{X}_1 (and fewer than $n_2 = N_p - n_1$ samples in \mathbf{X}_2). Then set

$$\alpha^2 = \frac{n_1}{n_2}. \quad (3.58)$$

This choice of α is suitable for a large number of samples such that the cross-correlations among the independent sources tend to zero.

With a finite number of samples, α acts as a detection threshold parameter with regard to the definition of a target packet (defined by n_1, n_2), as it determines a certain probability of detection and false alarm. These probabilities as function of both α and the number of samples will be computed here. However, there exists a suitable α such that the probability of false alarm is zero. Assume, in a worst-case scenario, that there are M packets at a marginal position, i.e., with $n_1^{(k)} = n_1$ and $n_2^{(k)} = n_2$. Further assume without loss of generality that the sources have equal unit power, and that all sources have the same Gaussian distribution. For this scenario, the condition that the packets must be classified as interferers (i.e., no false alarm) is

$$n_1 \hat{\lambda}_1 < \alpha^2 n_2 \hat{\lambda}_M,$$

where $\hat{\lambda}_1$ is the largest eigenvalue of a sample covariance matrix generated by n_1 i.i.d. complex white Gaussian noise samples, and $\hat{\lambda}_M$ is the smallest eigenvalue of such a matrix, now for n_2 samples. Results from random matrix theory listed in Appendix 3.C allow to approximate these eigenvalues, and it is proper to set

$$\alpha^2 > \alpha_{\min}, \quad \alpha_{\min} := \frac{\beta_1^2(n_1)}{\beta_M^2(n_2)} \left(\frac{\sqrt{n_1} + \sqrt{M}}{\sqrt{n_2} - \sqrt{M}} \right)^2$$

where $\beta_1(n)$ and $\beta_M(n)$ are Tracy-Widom factors defined in (3.62), (3.63). This generalizes (3.58) for a finite number of samples.

For Scenario 2, similar considerations lead to two conditions

$$\alpha^2 > \frac{\beta_1^2(n_2)}{\beta_M^2(n_1)} \left(\frac{\sqrt{n_2} + \sqrt{M}}{\sqrt{n_1} - \sqrt{M}} \right)^2$$

$$\alpha^2 > \frac{\beta_1^2(n_1)}{\beta_M^2(n_2)} \left(\frac{\sqrt{n_1} + \sqrt{M}}{\sqrt{n_2} - \sqrt{M}} \right)^2.$$

The larger one of these two lower bounds on α should be used.

Finally, if the matrices $\mathbf{X}_1, \mathbf{X}_2$ of the two blocks in Scenario 1 and Scenario 2 have different lengths N_1, N_2 , then in the expression of the noise power threshold ϵ in (3.23) and similar equations, $N_1 + N_2 = N_s$ should be used in place of $2N$. The noise power shift parameter γ in (3.44), (3.56) and (3.57) also needs to be adjusted. Appendix 3.D lists the appropriate generalizations.

3.9 Simulation Results

3.9.1 Data Model with Stationary Sub-Blocks

To apply the algorithms to the data model derived above, their performance on “ \mathbf{Y} ” matrices, i.e., for stationary data matrices $(\mathbf{Y}_1, \mathbf{Y}_2)$ each of size $M \times N$ is firstly examined. Specifically, the data are generated according to

$$\begin{aligned}\mathbf{Y}_1 &= \mathbf{H}_s \mathbf{S}_{s1} + \sqrt{r} \mathbf{H}_f \mathbf{S}_{f1} + \mathbf{N}_1 \\ \mathbf{Y}_2 &= \sqrt{r} \mathbf{H}_s \mathbf{S}_{s2} + \mathbf{H}_f \mathbf{S}_{f2} + \mathbf{N}_2\end{aligned}\quad (3.59)$$

where the signal and interference data matrices \mathbf{S}_{s1} , \mathbf{S}_{s2} , \mathbf{S}_{f1} , and \mathbf{S}_{f2} correspond to unit-power QPSK sources with symbols taken from the alphabet $\{+1, +j, -1, -j\}$. The “mixing” parameter r ($0 \leq r < 1$) controls the signal to interference ratio (SIR), defined by $-10 \log_{10}(r)$ dB. The algorithms are tested for $\alpha = 1$, which determines the classification threshold.

To simplify the comparison, only one signal and one interference in (3.59) are considered. The columns of \mathbf{H} are generated, acting as the array response vectors of a uniform linear antenna array consisting of $M = 4$ elements with separation of 0.5 wavelength. The target signal direction of arrival (DOA) is fixed at 0° . The interference DOA is set at 20° unless specified otherwise. The noise matrices are generated from i.i.d. complex Gaussian sources with variance σ^2 . The SNR is defined as $\text{SNR} = 10 \log_{10}(1/\sigma^2)$. The SNR is set at 15 dB unless specified otherwise. The four algorithms are compared: *GSVD* and *GEVD* as shown in Table 3.1; *SURV1* (with rank reduction) and *SURV2* (without rank reduction), see Table 3.2.

Instead of beamformers, the underlying estimates of $\text{ran}(\mathbf{H}_s)$ and $\text{ran}(\mathbf{H}_f)$ for each algorithm are considered. the performance measure is considered though the subspace error $\mathcal{E}(\mathbf{U}, \mathbf{U}_e)$, defined for an orthonormal basis \mathbf{U} of the “true” subspace and an orthonormal basis \mathbf{U}_e of its estimate, as

$$\mathcal{E}(\mathbf{U}, \mathbf{U}_e) = \|(\mathbf{I} - \mathbf{U}_e \mathbf{U}_e^H) \mathbf{U}\|.$$

In Fig. 3.3(a), the subspace error of the algorithms is shown as a function of the mixing factor r at $\text{SNR} = 15$ dB. There is not much difference between the algorithms. It can be seen that the performance of all the algorithms drops when the mixing factor r goes towards 1 (SIR = 0 dB). At this marginal case, it is not expected to distinguish the subspaces.

Fig. 3.3(b) shows the performance for varying separation in DOA between both sources, for $r = 0.5$ (SIR = 3 dB). The target DOA is fixed at 0° and the interference DOA goes from 0° to 90° . All the algorithms give the same performance.

Fig. 3.3(c) shows the performance as a function of N , for $r = 0.5$. For very large N , the performance lines start to split, indicating that the algorithms, except *GEVD* with noise power subtraction, are not asymptotically unbiased, as expected for the *GSVD* in case of additive white noise.

Fig. 3.3(d) shows the performance as a function of SNR, for $r = 0.5$ (SIR = 3 dB). Unless the SNR is small, the performance of the algorithms is dominated by the non-zero finite sample cross-correlations of the sources.

Overall, it can be concluded that essentially there is no performance difference among GSVD, GEVD and SURV algorithms for the separation of target signals and interferences in the block-stationary data model (3.59).

3.9.2 Data Model with Partially Overlapping Data Packets

The proposed algorithms, as shown in Fig. 3.1 and Fig. 3.2, respectively, for separating partially overlapping data packets in Scenario 1 and Scenario 2 are compared. All the packets have the same length $N_p = 256$ symbols. The length of the analysis window is $N_s = 3N_p$. For Scenario 1, the length of \mathbf{X}_1 is $N_1 = N_p$ and the length of \mathbf{X}_2 is $N_2 = 2N_p$. For Scenario 2, the two lengths are taken to be the same: $N_1 = N_2 = 1.5N_p$.

The source data are QPSK symbols. A uniform linear antenna array of $M = 5$ elements with separation of 0.5 wavelength is used. Consider $d_s = 2$ target packets with DOAs $[-10^\circ, 40^\circ]$, and $d_f = 3$ partially overlapping interference packets with DOAs $[5^\circ, -30^\circ, -60^\circ]$.

The packets with arbitrary arrival times are first generated. To describe the extent of overlap, let $n_1^{(k)}$ be the number of nonzero samples of the k th packet in block \mathbf{X}_1 , and define the ‘‘overlapping ratio’’ r_k as $r_k = \frac{n_1^{(k)}}{N_p}$. In Scenario 1, two kinds of packets must be generated: the target packets with r_k randomly selected in the interval $[0.85, 1]$, and the interference packets with $r_k \in [0, 0.5]$. In Scenario 2, the target packets are transmitted continuously and fill up the analysis window. The interference packets have $r_k \in [0, 0.5]$.

The following algorithms are compared: *GSVD* and *GEVD* for Scenario 1 and Scenario 2, as in Table 3.1; *SURV1-S1* and *SURV2-S1* as in Table 3.2; and *SURV1-S2* as in Table 3.3.

After the interference suppressed by these algorithms, the two target packets are separated using ACMA [12]. The performance measure is the residual signal-to-interference-plus-noise ratio (SINR) at the output of the beamformers. The SINR of a single beamformer for source k is defined as

$$\text{sinr}(\mathbf{h}_k, \mathbf{w}) := \frac{\mathbf{w}^H (\mathbf{h}_k \sigma_k^2 \mathbf{h}_k^H) \mathbf{w}}{\mathbf{w}^H (\mathbf{H} \boldsymbol{\Sigma}^2 \mathbf{H}^H - \mathbf{h}_k \sigma_k^2 \mathbf{h}_k^H + \sigma^2 \mathbf{I}) \mathbf{w}},$$

where $\boldsymbol{\Sigma}^2$ is a diagonal matrix holding the power of signals, and σ_k^2 is the power of the k th signal. The overall output SINR is defined as the worst SINR among the beamformer SINRs, after the best assignment of beamformers is made to target source indices.

The values of the various parameters ϵ , γ and α in the algorithms are selected according to Section 3.8 and Appendix 3.D.

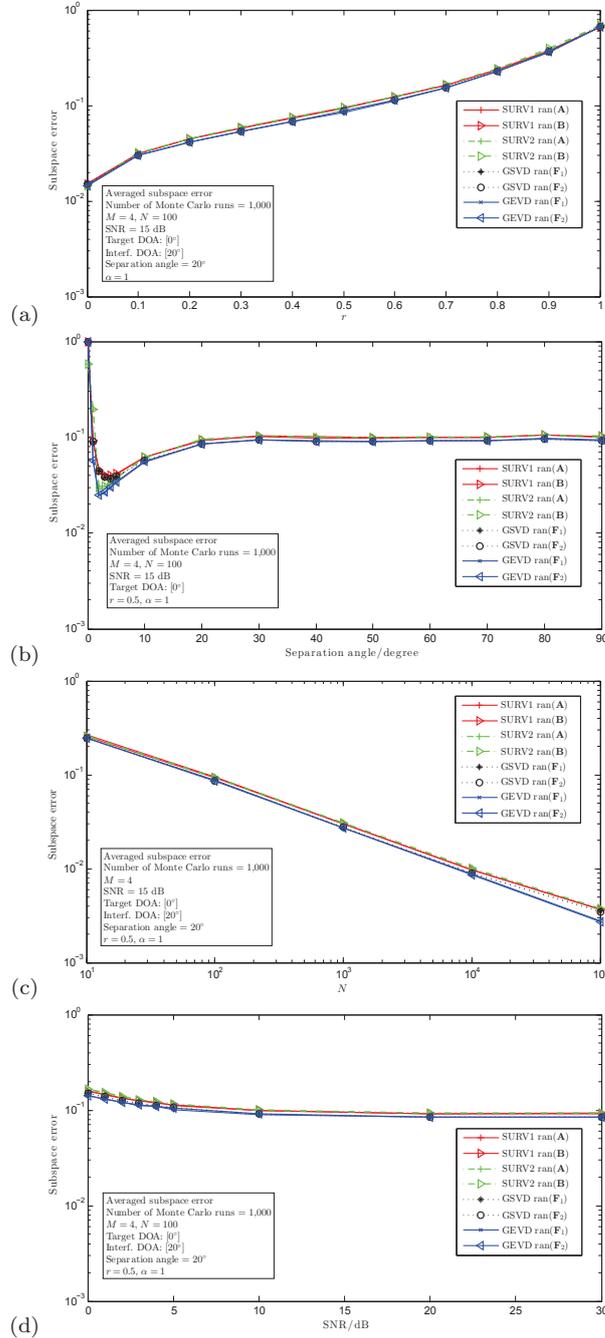


Figure 3.3: Separation performance for block-stationary data model in terms of the subspace error \mathcal{E} . (a) Varying SIR (parameter r) for SNR = 15 dB; (b) varying separation angle for $r = 0.5$; (c) varying N for $r = 0.5$; (d) varying SNR for $r = 0.5$.

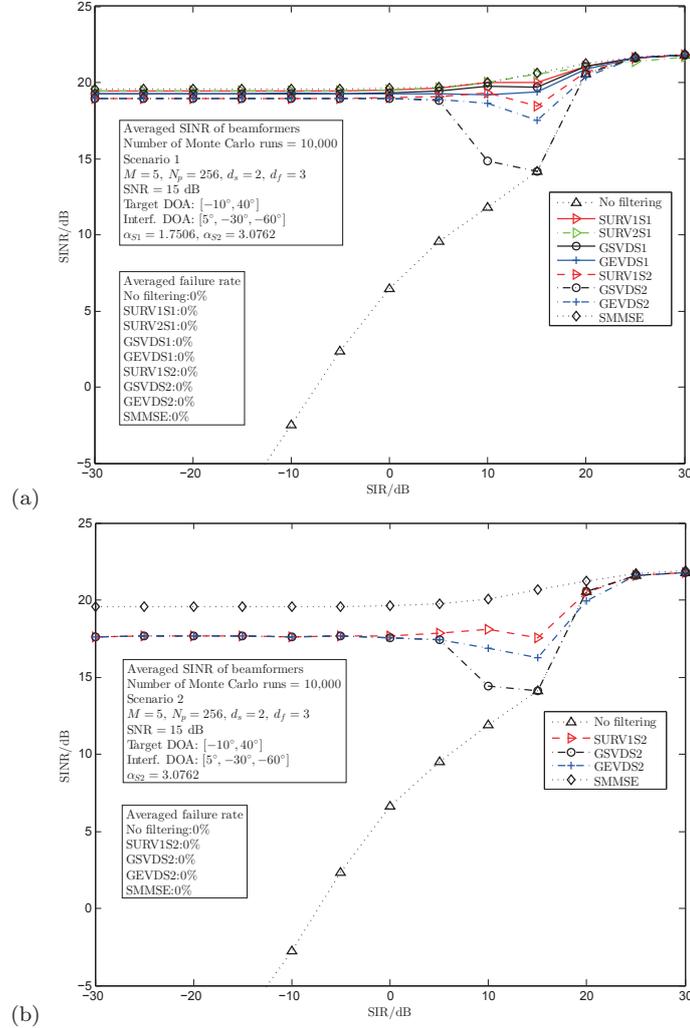


Figure 3.4: Separation performance of the proposed algorithms. (a) Scenario 1: SINR as a function of SIR for SNR = 15 dB; (b) Scenario 2: SINR as a function of SIR for SNR = 15 dB.

For Scenario 1, Fig. 3.4(a) shows the output SINR of the beamformers for SNR = 15 dB. Because Scenario 1 is actually a special case of Scenario 2, the performance of the algorithms for Scenario 2 is also shown. As lower bound on the performance, the curve “No filtering” shows the performance of the direct ACMA separation without suppression of partially overlapping interference packets. It is seen to behave poorly for low SIR. As upper bound on the performance, the curve “SMMSE” shows the performance of the standard minimum mean squared

error receiver obtained when all source data is known. It is seen from the plots that all the proposed algorithms keep a nearly constant performance and have identical performance in the low and high SIR regions. For medium SIR, all algorithms show a small drop in performance due to the poor estimation of the weak interference. GSVD-S2 is notably a little worse than the other algorithms in the medium SIR region.

For Scenario 2, Fig. 3.4(b) shows the output SINR when $\text{SNR} = 15$ dB. The performance is seen to be quite similar to the plots for Scenario 1, although all algorithms have a small performance gap compared to the SMMSE performance in the low SIR region. For medium SIR, the performance of GSVD is slightly worse than the other algorithms.

In summary, it is seen that all the algorithms have a very similar performance, which means that the choice of algorithm ought to be based on other criteria, e.g., the complexity of implementation. Recall that SURV2-S1 is the most attractive when the complexity is considered.

3.10 Experiments

A demonstration system for testing the proposed algorithms in a real application has been built: the automatic identification system for ships. AIS data is used for exchanging navigational information among ships and between ships and base stations in the maritime frequency band at 162 MHz. Most AIS data packets are 256 symbols long, Gaussian minimum shift keying (GMSK) modulated, and transmitted at a rate of 9.6 kbps. An uncalibrated receiver array consisting of $M = 4$ antennas is used to collect the baseband data samples. The antennas are roughly positioned at half-wavelength spacing. The RF front-ends for the 4 antennas are not identical and uncalibrated. The receiver array is placed at an altitude of 68 meters in the window of a building in Delft, facing the very open harbor and coastline of Rotterdam, about 10 nautical miles away. The receiver is behind two layers of window glass, and next to the metal frame of the building, i.e., multipath is expected to be present. Although AIS works in a synchronized network and no packet collision is tolerated in its protocol, the actual AIS packets from different communication cells are frequently seen overlapping by the receiver.

Fig. 3.5 shows in the top panels as example the amplitude of raw samples from the 4 antennas. The shown analysis window contains at least three overlapping data packets. The bottom panels shows one packet after separation from the other two partially overlapping packets. The length of the analysis window is $N_s = 2N_p = 512$ symbols. Data was sampled using an oversampling ratio of 10, so that the number of samples in one analysis window is 5120. The beamformers are computed after downsampling by a factor of 10. The upper panels in Fig. 3.5a and 3.5b show that the antennas have different responses to the same packet, which is mainly caused by multipath fading. In the shown example, one interference packet overlaps the head of the middle packet while another packet overlaps the

tail. The middle packet is the target packet in this analysis window. The input SIR is below 0 dB and the output SNR is around 10 dB. The lower panels in Fig. 3.5a and 3.5b show the amplitude of one packet separated by SURV1-S1 and SURV1-S2, respectively. Without the proposed algorithms, it is even difficult to visually identify the start and end of the separated packet.

The experience with this demonstrator shows that the separation performance is good; the algorithms are reliable and robust, and messages with good SNR are easily decoded after separation. More complete results on this experiment will be reported in Chapter 4.

3.11 Conclusions

In this chapter, several blind beamforming techniques for suppressing asynchronous co-channel interference has been established. These algorithms are based on the subspace estimations using GSVD and GEVD. Subsequently, SURV are introduced as a computationally efficient replacement for the GSVD. The provided simulations have shown that these algorithms have essentially equal performance, close to that of a reference MMSE receiver with completely known target signals.

In the next chapter, the effectiveness and robustness of the proposed algorithms on experimental data from a real application (AIS) will be demonstrated. The research also includes the design of tracking algorithms dedicated to specific applications.

3.A Comparison of SURV and SVD Principal Subspace Estimates

Some quantification of the accuracy of the subspace estimation obtained by using the SURV is given first. Consider a data model $\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{N}$, where \mathbf{H} has $d < M$ columns. For a large number of samples, such as N samples, the corresponding covariance model is $\frac{1}{N}\mathbf{X}\mathbf{X}^H \rightarrow \mathbf{R}_x = \mathbf{H}\mathbf{R}_s\mathbf{H}^H + \sigma^2\mathbf{I}$ when N tends to infinite. Without loss of generality, assume $\mathbf{R}_s = \mathbf{I}$ and insert the “skinny” SVD of \mathbf{H} as $\mathbf{H} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^H$. Let σ_0 be the smallest singular value of \mathbf{H} and take $\epsilon > \sigma\sqrt{N}$ but not larger than $\sqrt{N}(\sigma_0^2 + \sigma^2)^{1/2}$. Now, the SURV of $[\epsilon\mathbf{I}, \mathbf{X}]$ in given (3.31) and (3.32) is performed, and results in

$$\mathbf{X}\mathbf{X}^H - \epsilon^2\mathbf{I} = \mathbf{B}\mathbf{B}^H - \mathbf{A}\mathbf{A}^H. \quad (3.60)$$

For N samples (N is large enough),

$$\frac{1}{N}(\mathbf{X}\mathbf{X}^H - \epsilon^2\mathbf{I}) \rightarrow \mathbf{H}\mathbf{R}_s\mathbf{H}^H - \left(\frac{\epsilon^2}{N} - \sigma^2\right)\mathbf{I}.$$

For the given range of ϵ , it follows that the expected dimension of \mathbf{B} is that of \mathbf{H} . All the subspace estimates with tolerance ϵ are given by $\mathbf{B}' = \mathbf{B} - \mathbf{A}\mathbf{M}$ with

$\|\mathbf{M}\| < 1$, and these are the estimates of $\text{ran}(\mathbf{H})$. If (3.35) is inserted into \mathbf{B}' as a special case, then all subspace estimates of $\text{ran}(\mathbf{H})$ satisfy

$$\mathbf{B}' = \mathbf{U}_0(\boldsymbol{\Sigma}_0^2 + N\sigma^2 - \epsilon^2)^{1/2} - \mathbf{U}_0^\perp(\epsilon^2 - N\sigma^2)^{1/2}\mathbf{M}$$

where \mathbf{U}_0^\perp is a matrix containing the orthogonal complement of \mathbf{U}_0 . Thus, when ϵ is close to $\sigma\sqrt{N}$, $\text{ran}(\mathbf{B}')$ is close to $\text{ran}(\mathbf{U}_0) = \text{ran}(\mathbf{H})$. Moreover, the bound on the subspace angle between $\text{ran}(\mathbf{B}')$ and $\text{ran}(\mathbf{H})$ can also be found to be of order $(\frac{\epsilon^2}{N} - \sigma^2)^{1/2}$. Therefore, for accurate results, it is better to take the threshold ϵ close to $\sigma\sqrt{N}$, but slightly larger.

For N samples (N is large enough), the parameter ϵ cannot be selected to be too close to $\sigma\sqrt{N}$, because there exists the risk that \mathbf{X} has more than d singular values above the threshold. As derived in Appendix 3.C, it is better to set $\epsilon > \beta_1\sigma(\sqrt{N} + \sqrt{M})$, where $\beta_1 > 1$ is a factor that depends on M and N , see (3.62).

3.B GSVD is a Special Case of SSE-2

Given two matrices $\mathbf{N} \in \mathbb{C}^{M \times N_1}$ and $\mathbf{X} \in \mathbb{C}^{M \times N_2}$, consider the GSVD (3.36).

$$\begin{aligned} \mathbf{N} &= \mathbf{F}\mathbf{C}\mathbf{U}^H = \begin{bmatrix} d & M-d \\ \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \mathbf{C}_1 & \\ & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1^H \\ \mathbf{U}_2^H \end{bmatrix} \\ \mathbf{X} &= \mathbf{F}\mathbf{S}\mathbf{V}^H = \begin{bmatrix} d & M-d \\ \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix} \begin{bmatrix} \mathbf{S}_1 & \\ & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \end{aligned}$$

where the sorting and partitioning is such that $\mathbf{C}_1 > \mathbf{S}_1$, $\mathbf{C}_2 < \mathbf{S}_2$. Define $\boldsymbol{\Theta}$ and \mathbf{T} as

$$\begin{aligned} \boldsymbol{\Theta} &= \begin{bmatrix} \mathbf{U}_2\mathbf{C}_2 & \mathbf{U}_1\mathbf{S}_1 & | & -\mathbf{U}_1\mathbf{C}_1 & -\mathbf{U}_2\mathbf{S}_2 \\ -\mathbf{V}_2\mathbf{S}_2 & -\mathbf{V}_1\mathbf{C}_1 & | & \mathbf{V}_1\mathbf{S}_1 & \mathbf{V}_2\mathbf{C}_2 \end{bmatrix} \\ &\quad \times \text{diag}[\mathbf{C}_2^2 - \mathbf{S}_2^2, \mathbf{S}_1^2 - \mathbf{C}_1^2, \mathbf{S}_1^2 - \mathbf{C}_1^2, \mathbf{C}_2^2 - \mathbf{S}_2^2]^{-1/2} \\ \mathbf{T} &= \boldsymbol{\Theta}_{11}^{-1} = \begin{bmatrix} (\mathbf{C}_2^2 - \mathbf{S}_2^2)^{\frac{1}{2}} \mathbf{C}_2^{-1} \mathbf{U}_2^H \\ (\mathbf{S}_1^2 - \mathbf{C}_1^2)^{\frac{1}{2}} \mathbf{S}_1^{-1} \mathbf{U}_1^H \end{bmatrix}. \end{aligned}$$

It is readily verified that $\boldsymbol{\Theta}$ is \mathbf{J} -unitary, and that

$$\begin{aligned} [\mathbf{N} \quad \mathbf{X}]\boldsymbol{\Theta} &= [\mathbf{F}_2(\mathbf{C}_2^2 - \mathbf{S}_2^2)^{\frac{1}{2}} \mathbf{0} \mid \mathbf{F}_1(\mathbf{S}_1^2 - \mathbf{C}_1^2)^{\frac{1}{2}}] \\ \mathbf{T} [\mathbf{I} \quad \mathbf{0}] \boldsymbol{\Theta} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} & | & \mathbf{0} & -\mathbf{C}_2^{-1}\mathbf{S}_2 \\ \mathbf{0} & \mathbf{I} & | & -\mathbf{S}_1^{-1}\mathbf{C}_1 & \mathbf{0} \end{bmatrix}. \end{aligned}$$

Let $\mathbf{A} = \mathbf{F}_2(\mathbf{C}_2^2 - \mathbf{S}_2^2)^{\frac{1}{2}}$ and $\mathbf{B} = \mathbf{F}_1(\mathbf{S}_1^2 - \mathbf{C}_1^2)^{\frac{1}{2}}$. The first equation establishes $\boldsymbol{\Theta}$, \mathbf{A} , \mathbf{B} as an SSE decomposition, and the second equation shows that $\boldsymbol{\Theta}$ satisfies a structural constraint that makes it an SSE-2 [14], with favorable properties.

Now, let $[\mathbf{A}, \mathbf{B}]$ be further decomposed as $[\mathbf{A}, \mathbf{B}] =: \mathbf{Q}\mathbf{R} = [\mathbf{Q}_A, \mathbf{Q}_B][\mathbf{R}_A, \mathbf{R}_B]$, where $\mathbf{Q} = [\mathbf{Q}_A, \mathbf{Q}_B]$ is unitary and $\mathbf{R} = [\mathbf{R}_A, \mathbf{R}_B]$ is lower triangular, then the required two-sided decomposition (SURV) in (3.29) is obtained, moreover, $\text{ran}(\mathbf{Q}_B) = \text{ran}(\mathbf{F}_1)$.

3.C Bounds on Eigenvalues of Random Matrices

The following results from random matrix theory are needed in Appendix 3.D. Consider a random complex white Gaussian noise variable $\mathbf{n}[k]$ with covariance matrix $\mathbf{R}_n = \sigma^2 \mathbf{I}$. For N samples, the corresponding sample covariance matrix is $\hat{\mathbf{R}}_n = \frac{1}{N} \sum_{k=1}^N \mathbf{n}[k] \mathbf{n}[k]^H$. The largest eigenvalue of $\hat{\mathbf{R}}_n$ is denoted by $\hat{\lambda}_1$ and the smallest by $\hat{\lambda}_M$.

When $N \rightarrow \infty$, it follows from Bai and Yin [79] that

$$\begin{aligned} \hat{\lambda}_1 &\rightarrow \sigma^2 \rho_1^2, & \rho_1 &:= 1 + \frac{\sqrt{M}}{\sqrt{N}}, \\ \hat{\lambda}_M &\rightarrow \sigma^2 \rho_M^2, & \rho_M &:= 1 - \frac{\sqrt{M}}{\sqrt{N}}. \end{aligned} \quad (3.61)$$

These are equal to the expected values of $\hat{\lambda}_1$ and $\hat{\lambda}_M$. For more accurate upper bounds, the known distributions of eigenvalues can be employed, as follows. Define centering and scaling constants as

$$\begin{aligned} \mu_1 &= N\rho_1^2, & \nu_1 &= N^{1/2}M^{-1/6}\rho_1^{4/3} \\ \mu_M &= N\rho_M^2, & \nu_M &= N^{1/2}M^{-1/6}\rho_M^{4/3}. \end{aligned}$$

Under some conditions, it has been shown in [80] that $w_1 := \frac{N\hat{\lambda}_1/\sigma^2 - \mu_1}{\nu_1}$ converges to the Tracy-Widom distribution of order 2, $w_1 \sim F_2(s)$ [81]. Similarly, it has been shown in [82] that $w_M := \frac{N\hat{\lambda}_M/\sigma^2 - \mu_M}{\nu_M}$ converges to the reflected Tracy-Widom distribution of order 2, $w_M \sim 1 - F_2(-s)$.

For the largest eigenvalue, there should exist a threshold u_1 for which the probability that $\hat{\lambda}_1 > u_1$ is zero. This fact is deduced as follows. By using the distribution of $\hat{\lambda}_1$, it is known that

$$\begin{aligned} P(\hat{\lambda}_1 > u_1) = 0 &\Leftrightarrow P\left(\frac{N\hat{\lambda}_1/\sigma^2 - \mu_1}{\nu_1} > \frac{N u_1/\sigma^2 - \mu_1}{\nu_1}\right) = 0 \\ &\Leftrightarrow \frac{N u_1/\sigma^2 - \mu_1}{\nu_1} = F_2^{-1}(1) \end{aligned}$$

where $F_2^{-1}(s)$ is the inverse CDF, and $f_1 := F_2^{-1}(1) \approx 2.24$. It follows that

$$\begin{aligned} u_1 &= \frac{\sigma^2}{N}(\mu_1 + \nu_1 f_1) \\ &= \frac{\sigma^2}{N} \left[N\rho_1^2 + N^{1/2}M^{-1/6}\rho_1^{4/3} f_1 \right] \\ &= \sigma^2 \rho_1^2 \beta_1^2, \quad \beta_1^2(N) := 1 + N^{-\frac{1}{2}}M^{-\frac{1}{6}}\rho_1^{-\frac{2}{3}} f_1 \end{aligned} \quad (3.62)$$

which shows a refinement of (3.61) by a factor β_1^2 . For the smallest eigenvalue, there also should exist a threshold u_M for which the probability that $\hat{\lambda}_M < u_M$ is zero. Through the process similar to the above, it can be obtained that

$$u_M = \sigma^2 \rho_M^2 \beta_M^2, \quad \beta_M^2(N) := 1 - N^{-\frac{1}{2}} M^{-\frac{1}{6}} \rho_M^{-\frac{2}{3}} f_1 \quad (3.63)$$

A plot of β_1 and β_M for $M = 4$ is shown in Fig. 3.6. It is seen that β_1 is between 1 and 2 for reasonable values of N . From (3.63), it is possible to compute the smallest value of N for which $\beta_M > 0$ as

$$N_{min} = \left\lceil \left(\frac{b}{3s} + s \right)^4 \right\rceil \quad (3.64)$$

where $s = \left[\frac{1}{2}c + \left(\frac{1}{4}c^2 - \frac{1}{27}b^3 \right)^{1/2} \right]^{1/3}$, $b = M^{1/2}$, and $c = \left(\frac{f_1}{M^{1/6}} \right)^{3/2}$. For example, for $M = 4$, it can be used that $N \geq N_{min} = 11$.

3.D Noise Power Shifting Using γ

In Eq. (3.44) ff. a parameter γ was introduced to make the noise subspace become a small but negative component in a covariance difference equation. This appendix shows a more detailed derivation for a slightly more general case.

The noise power shift is sometimes positive, sometimes negative, so it is necessary to introduce a more general notation that captures this attribute in a single expression. Then, let

$$[\mathbf{Y}_1^+ \ \alpha \mathbf{Y}_2^- \ \gamma \mathbf{I}^j] \Theta = [\mathbf{A} \ \mathbf{0} \ | \ \mathbf{B} \ \mathbf{0}] \quad (3.65)$$

where, $j = \pm 1$, and the superscripts $+$, $-$ and j indicate the ‘signature’ of each component, i.e., the entries of the corresponding signature matrix \mathbf{J} . More details on this notation can be found in [14]. After Eq. 3.65 is squared, it follows that

$$\mathbf{Y}_1 \mathbf{Y}_1^H - \alpha^2 \mathbf{Y}_2 \mathbf{Y}_2^H + j\gamma^2 \mathbf{I} = \mathbf{A} \mathbf{A}^H - \mathbf{B} \mathbf{B}^H. \quad (3.66)$$

which suggests adding $\gamma^2 \mathbf{I}$ when $j = +1$ and subtracting $\gamma^2 \mathbf{I}$ when $j = -1$.

Assume \mathbf{Y}_1 has N_1 samples, and \mathbf{Y}_2 has N_2 samples. Then the data model for (3.66) is

$$\begin{aligned} N_1 \hat{\mathbf{R}}_1 - \alpha^2 N_2 \hat{\mathbf{R}}_2 + j\gamma^2 \mathbf{I} &\approx \mathbf{H}_s (N_1 \hat{\mathbf{R}}_{s1} - \alpha^2 N_2 \hat{\mathbf{R}}_{s2}) \mathbf{H}_s^H \\ &\quad - \mathbf{H}_f (\alpha^2 N_2 \hat{\mathbf{R}}_{f2} - N_1 \hat{\mathbf{R}}_{f1}) \mathbf{H}_f^H + N_1 \hat{\mathbf{R}}_{n1} - \alpha^2 N_2 \hat{\mathbf{R}}_{n2} + j\gamma^2 \mathbf{I} \end{aligned}$$

where ignoring the crossterms in the correlations (they diminish by order $1/N$ whereas the remaining components of order $1/\sqrt{N}$). The objective is to select

proper j and γ so that $N_1 \hat{\mathbf{R}}_{n1} - \alpha^2 N_2 \hat{\mathbf{R}}_{n2} + j\gamma^2 \mathbf{I}$ is (slightly) negative definite, and thus becomes part of $\mathbf{B}\mathbf{B}^H$ that also contains the interference subspace.

Let $\hat{\lambda}_1$ be the largest eigenvalue of $\hat{\mathbf{R}}_{n1}$ and $\hat{\lambda}_M$ be the smallest eigenvalue of $\hat{\mathbf{R}}_{n2}$, and define

$$t = \alpha^2 N_2 \hat{\lambda}_M - N_1 \hat{\lambda}_1.$$

The inequality $j\gamma^2 < t$ is sufficient for achieving the above objective. Note that whether $t > 0$ or $t < 0$ are possible, depending on α and the number of samples.

As shown in Appendix 3.C, a good estimate of t is

$$t = \sigma^2 [\alpha^2 \beta_M^2(N_2) (N_2^{\frac{1}{2}} - M^{\frac{1}{2}})^2 - \beta_1^2(N_1) (N_1^{\frac{1}{2}} + M^{\frac{1}{2}})^2] \quad (3.67)$$

where $\beta_1(N)$ was defined in (3.62) and $\beta_M(N)$ in (3.63). It is convenient to set $j = \text{sign}(t)$ and $\gamma = \sqrt{|t|}$ as in (3.44).

A similar result is needed in (3.56) in Scenario 2, involving the SURV of $[\alpha \mathbf{Y}_1 \ \mathbf{Y}_2 \ \gamma \mathbf{I}]$. For that case, it can be obtained that

$$t = \sigma^2 [\beta_M^2(N_2) (N_2^{\frac{1}{2}} - M^{\frac{1}{2}})^2 - \alpha^2 \beta_1^2(N_1) (N_1^{\frac{1}{2}} + M^{\frac{1}{2}})^2] \quad (3.68)$$

and $j = \text{sign}(t) = -1$, $\gamma = \sqrt{|t|}$. As for the result, in (3.57) involving the SURV of $[\alpha \mathbf{Y}_2 \ \mathbf{Y}_1 \ \gamma \mathbf{I}]$, it can be obtained that

$$t = \sigma^2 [\beta_M^2(N_1) (N_1^{\frac{1}{2}} - M^{\frac{1}{2}})^2 - \alpha^2 \beta_1^2(N_2) (N_2^{\frac{1}{2}} + M^{\frac{1}{2}})^2]$$

and $j = \text{sign}(t) = -1$, $\gamma = \sqrt{|t|}$.

3.E Convergence of Correlations

Consider two signals $s_1[k]$ and $s_2[k]$, $k = 0, \dots, N-1$. Each signal has N_p nonzero samples and $N - N_p$ zero samples. The nonzero parts are random signals with zero mean and unit variance, and overlap with N_x samples ($N_x < N_p$). Stack the two signals into a data matrix \mathbf{S}_x of size $2 \times N$ and compute the sample correlation matrix $\hat{\mathbf{R}}_x = \frac{1}{N} \mathbf{S}_x \mathbf{S}_x^H$. Similarly, let \mathbf{S}_y be a $2 \times N$ data matrix with two equal-variance zero-mean stationary random sources, and compute $\hat{\mathbf{R}}_y = \frac{1}{N} \mathbf{S}_y \mathbf{S}_y^H$.

In order to have equal auto-correlations in $\hat{\mathbf{R}}_x$ and $\hat{\mathbf{R}}_y$, the variance of the sources in \mathbf{S}_y should be set lower by a factor $\ell = N_p/N < 1$. The cross-correlations in $\hat{\mathbf{R}}_x$ go to zero as $O(\frac{N_x}{N^2})$. For $\hat{\mathbf{R}}_y$, taking into account the factor ℓ , the cross-correlations go to zero as $O(\frac{N_p}{N^2})$. Because $N_x < N_p$, the cross-correlations go faster to zero for the nonstationary case. In conclusion, the sample correlation matrices for the nonstationary signals converge faster to their models than for the stationary signals.

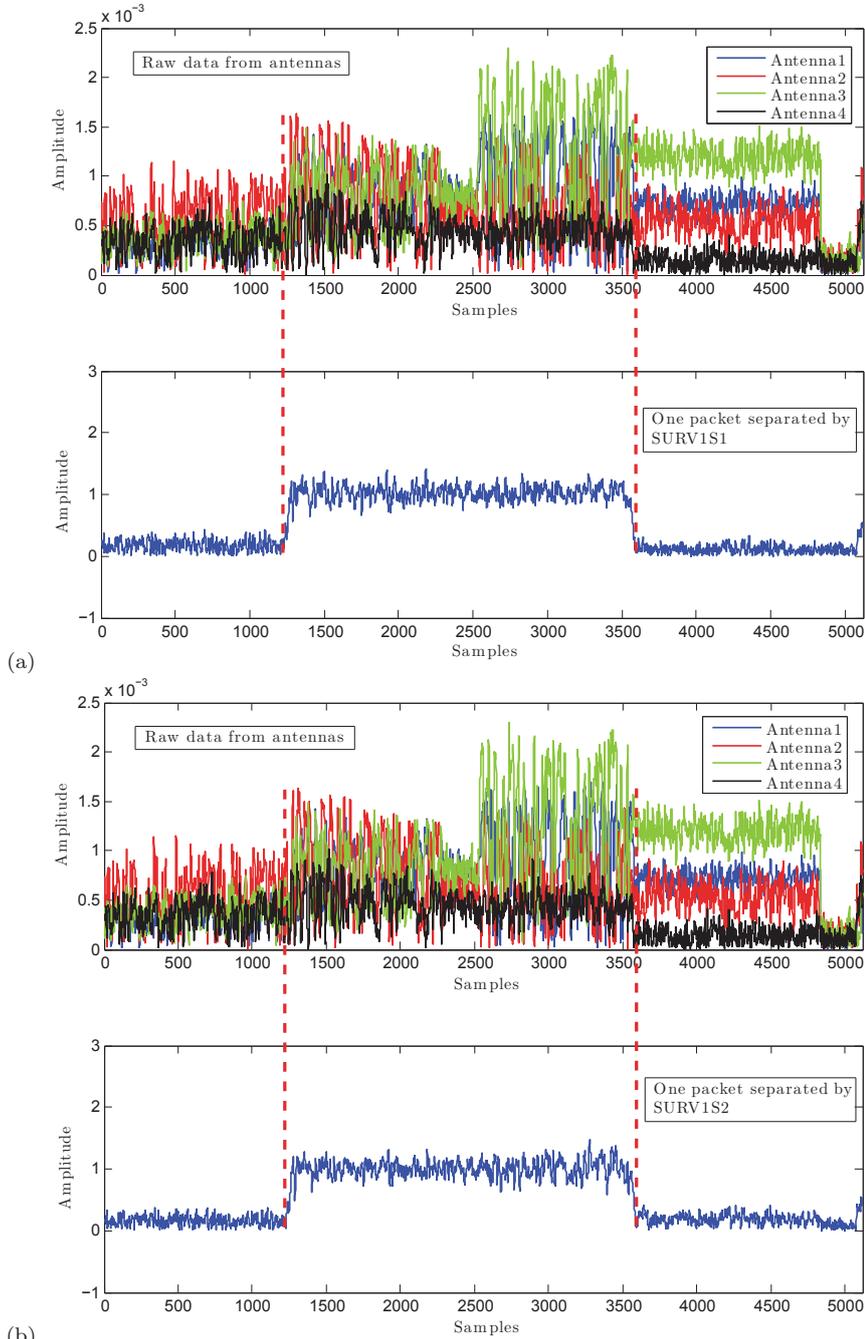


Figure 3.5: Examples of the proposed source separation algorithms on experimental AIS data. The upper panels in (a) and (b) show the amplitude of the raw data from the four antennas in one analysis window. The lower panel in (a) shows the amplitude of one packet separated by SURV1-S1, whereas the lower panel in (b) shows the separation by SURV1-S2.

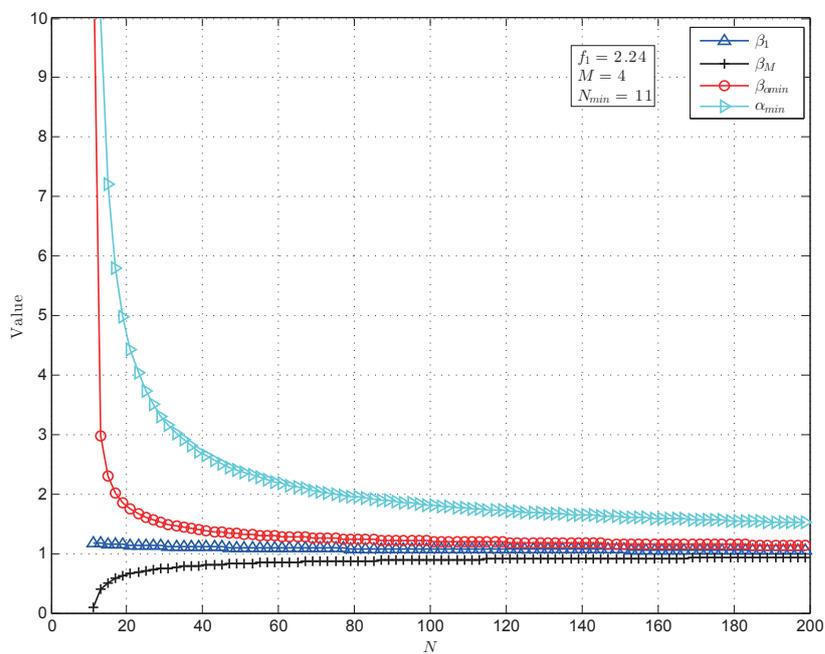


Figure 3.6: β_1 , β_M , $\beta_{\alpha_{min}} = \beta_1/\beta_M$ and α_{min} as a function of N , for $M = 4$ and matrices of equal dimensions.

Chapter 4

Robust Multi-User Receiver for Satellite AIS Using Blind Beamforming Techniques and Tracking

4.1 Introduction

In the previous chapters we have derived the SURV tool for subspace tracking, and applied it in the beamforming technique for separating overlapping data packets. We now look at the system aspects of these tools. The beamforming technique with tracking based on SURV is applied to Satellite AIS in this chapter. Fig. 4.1 shows that the field of view of a satellite covers a large number of ships in certain regions. The lack of synchronization and coordination among communication cells results in asynchronous interference. Although the protocol of AIS is designed to avoid message collision in a service coverage up to a diameter of 200 nautical miles [83], both types of collisions (See the definitions of the collisions in Chapter 1) are witnessed in real-world AIS signals, e.g., at busy harbors.

Several recent papers [9,84,85] discuss the use of the receivers equipped with only one antenna to detect AIS signals in such harsh environment. This type of receiver is targeted at a single communication cell and works fine if the interference from other cells is weak. The papers [86,87] exploit the Doppler diversity and apply interference cancellation to deal with interference. The wave polarizations caused by the Faraday rotation and the zonal filters discussed in the papers [86,87] (but not the focus of this thesis) can be used with the beamforming techniques to be introduced in this chapter to improve the performance of the receivers.

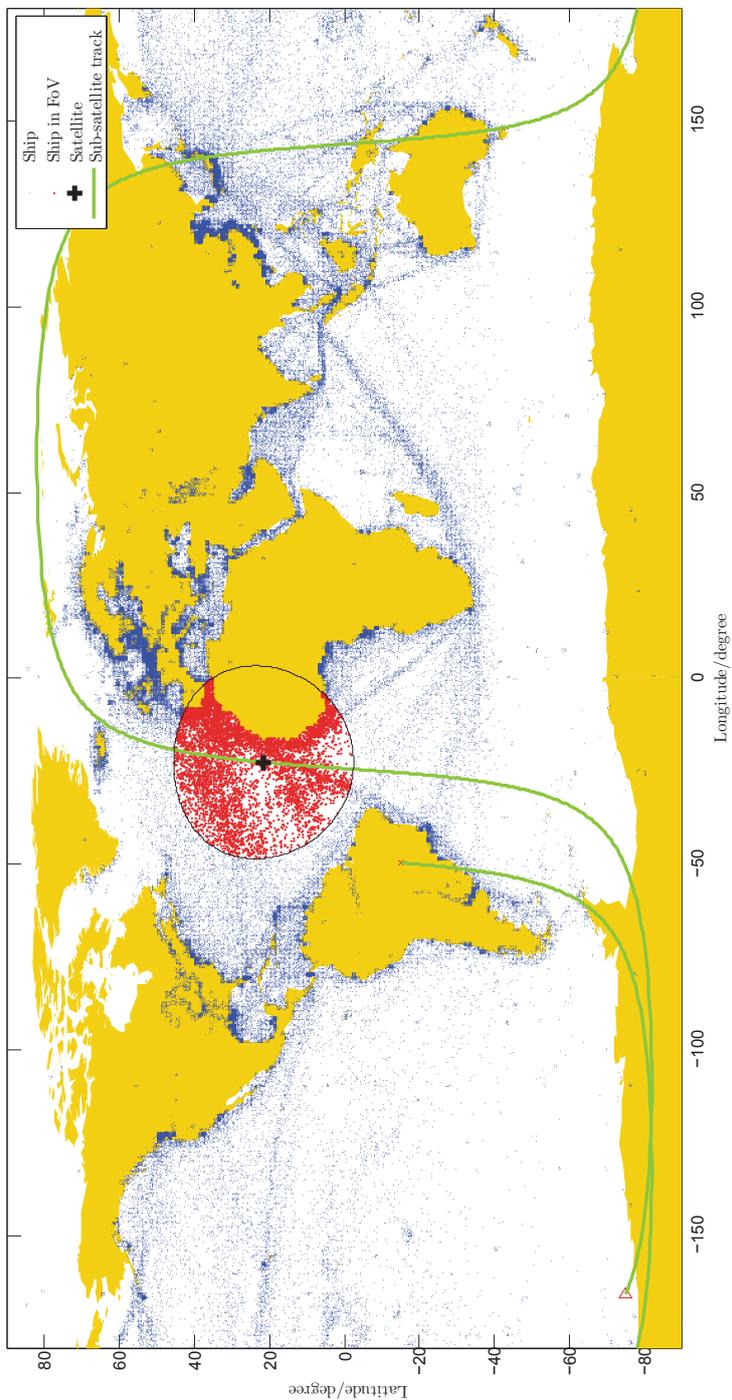


Figure 4.1: Global ship distribution [88, 89] and the field of view (FoV) of a LEO satellite orbiting at an altitude of 600 km, at the position denoted by the black “+” sign. The blue dots denote ships, and the red dots denote ships within the FoV.

Receivers equipped with an antenna array provide the possibility of receiving AIS signals from multiple communication cells simultaneously [5, 23, 90, 91]. The additional antennas enable source separation by beamforming techniques. The main issue is the estimation of the beamforming vectors.

Generic estimation techniques relying on known training sequences [67, 68, 92, 93] are inefficient for the considered problem since the AIS scenario is asynchronous and all messages use the same training sequence. For Satellite AIS, an additional complication is the significant Doppler shift present on each message, which has to be estimated as well.

Instead, blind source separation techniques can employ the known signal structure, such as a constant modulus modulation [12]. These techniques are almost all based on block stationary scenarios, i.e., synchronous interference. For example, they would be able to handle Type 1 collisions, but are not robust in the case of Type 2 collisions (partially overlapping messages).

Alternatively, the structure of the array response (direction) vectors is used. In this category, it was proposed in [5, 91] that both the classical direction of arrival (DOA) estimation and the beamforming techniques on structured linear arrays are used to separate AIS messages. Highly accurate DOA estimation is essential, which requires careful calibration and a known beam pattern of the array. For a LEO satellite, the additional calibration complicates the satellite payload. Moreover, these techniques cannot distinguish between complete and incomplete messages and thus waste processing complexity on incomplete messages.

Thus, the Type 2 collision is a key problem for the considered receivers, especially for those on LEO satellites, where Type 2 collisions frequently occur together with other issues, such as a large variance in Doppler frequency shifts and a large variance in the power of the received signals [4–7].

This chapter mainly addresses this problem and focuses on a multi-antenna receiver in Satellite AIS. A multi-user receiver consisting of three processing stages is proposed. In the first stage, asynchronous messages are detected as interference and suppressed by a new blind beamforming technique (BBT) which was recently proposed (See Chapter 3 and [15]). In this technique, the data from an observation interval is split into two sub-blocks and analyzed using a generalized singular decomposition (GSVD) [11]. This allows to detect subspaces of array response vectors of signals that are present in one block but not in the other, i.e., the subspaces of the synchronized and unsynchronized signals, respectively. The asynchronous interference is then suppressed via beamforming (projecting out the corresponding subspace). In the second stage, there exists only synchronous interference, and beamformers for each target message are found by the algebraic constant modulus algorithm (ACMA) [12]. In the final stage, the interference-free user data are demodulated by using a standard single user receiver.

This chapter will build the above receiver structure, and will focus mostly on the (innovative) first stage, especially on how the GSVD can be implemented by the computationally efficient signed URV (SURV) algorithm [14] proposed in Chapter 2. The SURV is very easily updated as new data arrives and old data is

discarded: it is a subspace tracking algorithm. This allows the analysis window to be a sliding window, as remarked in Chapter 3 and [15]. The proposed receiver structure is simulated for different scenarios of Satellite AIS including those cases with practical global ship densities (distributions).

Outline

In Section 4.2, the data model is introduced. In Section 4.3, the proposed multi-user receiver is presented in detail. In Section 4.4, the blind beamforming technique is elaborated. In Section 4.5, some simulation results are provided, and the conclusion is placed in Section 4.6.

Notation

Throughout the chapter, the following notational conventions are adopted: $(\cdot)^*$ denotes the complex conjugate operator. $(\cdot)^T$ denotes matrix transpose and $(\cdot)^H$ denotes matrix conjugate transpose. $E\{\cdot\}$ is the expectation operator. $\|\cdot\|$ denotes the matrix 2-norm, which is equal to the largest singular value of the matrix. The column span (range) of a matrix \mathbf{A} is $\text{ran}(\mathbf{A})$.

4.2 Data Model

In AIS, each communication cell divides time into time frames of 1 minute. A frame consists of 2250 time slots. One AIS message can occupy up to five consecutive time slots but most messages are default messages occupying one time slot. A default message is $N_p = 256$ bits long (See Fig. 4.2), and this is also the length of a time slot. For the application point of view, the default messages are of most interest as they contain information related to the ships' identity, movement and position.

The AIS signal modulation is described in [83]. Essentially, a message consists of a binary sequence that is a Gaussian minimum shift keying (GMSK) modulated and transmitted at a rate of 9.6 kbps on a carrier at 162 MHz. The complex baseband representation of the transmitted AIS signal thus has a constant envelope form (it is a constant-modulus signal).

At the receiver side, an antenna array of M elements is used. Consider a short time interval ("analysis window"), in which d AIS signals are present. The complex baseband representation of the antenna signals sampled at the symbol rate $1/T_s$ is denoted by $x_i[k]$, where i is the antenna index and k the sample index, and these signals are stacked into a vector $\mathbf{x}[k] = [x_1[k], \dots, x_M[k]]^T$. Likewise, the corresponding AIS signals are stacked in a vector $\mathbf{s}[k] = [s_1[k], \dots, s_d[k]]^T$, where $s_i[k]$ denotes the i th AIS signal, including distortion by a Doppler shift. A narrow-band model applies to this situation, so that $\mathbf{x}[k]$ can be modeled as

$$\mathbf{x}[k] = \mathbf{h}_1 s_1[k] + \dots + \mathbf{h}_d s_d[k] + \mathbf{n}[k] = \mathbf{H}\mathbf{s}[k] + \mathbf{n}[k], \quad (4.1)$$

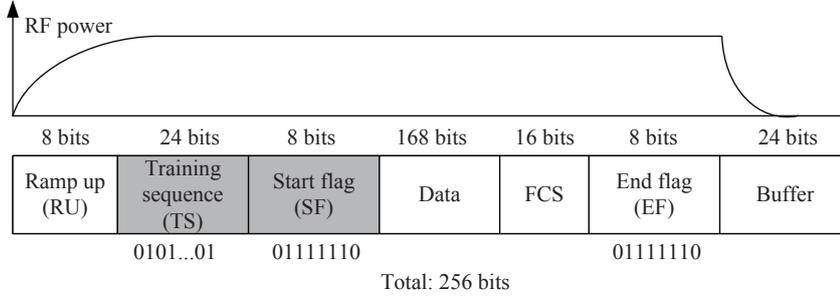


Figure 4.2: A default AIS message. It has power-up and power-off transient periods at its head and tail. The last 10 bits are normally considered empty with zero RF power.

where the vectors \mathbf{h}_i are the channel vectors (array response vectors or direction vectors) corresponding to each signal, $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_d] \in \mathbb{C}^{M \times d}$ is the channel matrix, and $\mathbf{n} \in \mathbb{C}^M$ is the noise vector. In this model, the noise is assumed to be i.i.d. zero mean complex Gaussian, with covariance matrix $\mathbf{R}_n = \mathbf{E}(\mathbf{n}\mathbf{n}^H) = \sigma^2\mathbf{I}$.

Let N_s denote the number of sample vectors $\mathbf{x}[k]$ in an analysis window, and stack these samples into a matrix $\mathbf{X} = [\mathbf{x}[1], \dots, \mathbf{x}[N_s]]$, and similarly for the signals and the noise. The data model corresponding to (4.1) is

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{N}. \quad (4.2)$$

It is better to select the size of the analysis window to be larger than one time slot (see Fig. 4.3), typically $N_s = 2N_p$. The Messages located around the center of the window are to be recovered and are considered “target messages”. These messages are completely received. The other messages are “interference messages”. These interference messages come from unsynchronized interference due to the lack of synchronization among adjacent cells and the large difference in propagation delays for signals to arrive at the satellite. These messages may be incomplete. The length of the analysis window has to be larger than a time slot so that enough samples of the interfering messages can be collected so as to do the estimation of their direction vectors in case they partially overlap with the target messages.

After collecting more samples, the analysis window will shift so that the interference messages in the current analysis window may become target messages in the next analysis window. If the coordinated universal time (UTC) is available (such as a GPS 1 pps signal), then the positions of the analysis windows can be aligned to the time slots of a reference cell. Otherwise, a typical window shift could be set less than $0.5N_p$ (being dependent on detection thresholds).

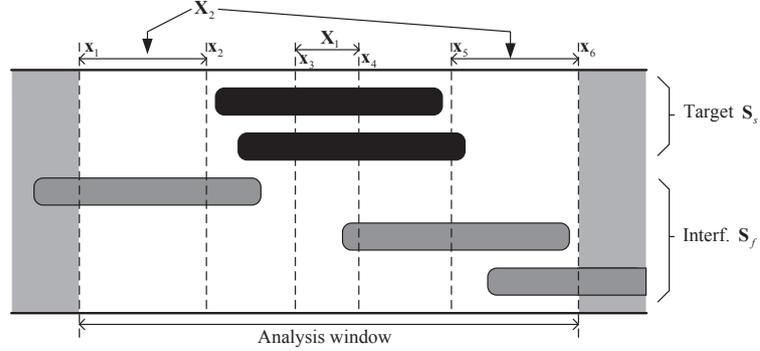


Figure 4.3: The analysis window and two sub-blocks, X_1 and X_2 . x_2 and x_6 are two incoming vectors for X_2 . x_1 and x_5 are two leaving vectors for X_2 . x_4 is an incoming vector for X_1 . x_3 is a leaving vector for X_1 .

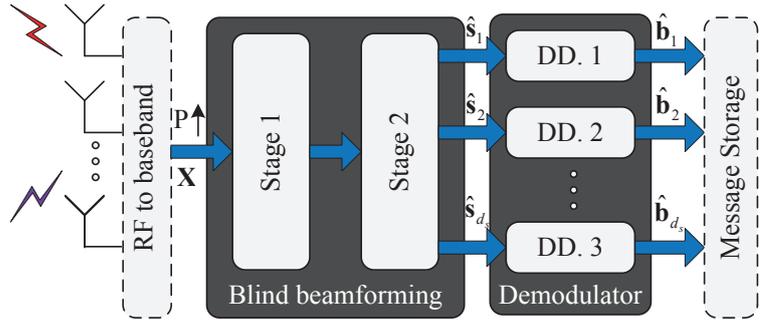


Figure 4.4: Block diagram of the proposed multi-user receiver.

4.3 Multi-User Receiver Structure

In this section, a multi-user receiver for Satellite AIS will be proposed. The block diagram of the receiver consists of four function blocks: an RF frontend block, a blind beamforming block, a demodulation block and a message storage block (see Fig. 4.4). Several of the blocks use mature techniques from the literature (such as the demodulator) and will not be detailed in this chapter. The functions of the blocks are described as follows.

1. **RF Frontend** The first block is a standard RF frontend for downconverting the received RF signals of each antenna to baseband. Here, the receivers are selected as RF quadrature receivers.
2. **Blind Beamforming** The second block is a blind beamforming block, where the incoming sample vectors are buffered and collected into a ma-

trix \mathbf{X} . The objective of this block is to compute a beamforming matrix \mathbf{W} that (i) can suppress any asynchronous interference, and (ii) can separate individual target messages. The result after these two stages is $\hat{\mathbf{S}}_s = \mathbf{W}^H \mathbf{X}$ where \mathbf{S}_s has size $d_s \times N_s$ and each row of \mathbf{S}_s contains a target message; d_s is the number of target messages. \mathbf{S}_s is handed over to the next block for demodulation, after which this block waits until the next sample matrix \mathbf{X} becomes available. As mentioned in Section 4.2, these matrices are overlapping, so partial messages first considered interference will later become target messages when they arrive at the center of \mathbf{X} .

3. **Demodulation** After beamforming, the i th target signal is recovered as

$$\hat{\mathbf{s}}_{si} = \mathbf{w}_i^H \mathbf{X}. \quad (4.3)$$

This signal is collision-free and centered in the vector (thus it is complete). At this point, the message can be decoded using a standard AIS detection algorithm, e.g., the differential detector described in [85]. The use of Viterbi decoder could improve the performance further [87]. Possible duplicated messages could be removed by processing at a upper level, e.g., by checking the MMSI and CRC in messages.

For the purpose of this chapter, the blind beamforming block is the most worthy of discussion in more details. In the considered scenario, the presence of incomplete messages in the analysis window makes the block of data “non-stationary”, that is, it contains signals that are partially zero. This situation is not suitable for the classic blind source separation algorithms which are normally designed for block stationary cases. Therefore, the first stage of the receiver will suppress the unsynchronized messages before a classic blind source separation algorithm is applied.

Starting with \mathbf{X} , the asynchronous interference messages are detected and suppressed by a first stage beamformer $\mathbf{W}^{(1)}$: $M \times d_s$ such that

$$\mathbf{X}^{(1)} = \mathbf{W}^{(1)H} \mathbf{X} \quad (4.4)$$

contains only target messages, synchronized to the center of $\mathbf{X}^{(1)}$. Next, this part of $\mathbf{X}^{(1)}$ is sent to the second stage. A constant modulus algorithm such as ACMA [12] can be used to compute separating beamformers $\mathbf{W}^{(2)}$: $d_s \times d_s$ so that the individual target messages are retrieved. ACMA is a block-based algorithm that can very reliably compute *all* beamformers $\mathbf{w}^{(2)}$ such that $\mathbf{w}^{(2)H} \mathbf{X}^{(1)}$ is a signal with a constant amplitude, such as an AIS message including a possible Doppler shift. The ACMA beamformers are found by solving a joint diagonalization problem.

The overall beamformers are formed as

$$\mathbf{W}^H = \mathbf{W}^{(2)H} \mathbf{W}^{(1)H} \quad (4.5)$$

and the target messages are recovered as

$$\hat{\mathbf{S}}_s = \mathbf{W}^H \mathbf{X} \quad (4.6)$$

where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{d_s}]$.

The computational complexities of the first and the second stage are $O(M^2 N_s)$ and $O(d_s^6)$, respectively. The first stage is the key step for improving the performance of ACMA as well as the whole receiver in the presence of asynchronous interference. This new technique with several design parameters will be further detailed in the next section.

4.4 Blind Beamforming Technique

In the considered scenario, the presence of incomplete messages in the analysis window makes the block of data nonstationary. This situation is not suitable for the classic source separation algorithms which are normally designed to suppress the asynchronous interference in block stationary cases. To combat this problem, it is necessary to explore a blind beamforming technique to suppress the interference messages before applying the classic source separation algorithms. The desired algorithm was proposed in Chapter 3 and [15]).

The key thing is to identify target messages and interference messages in one analysis window. The design principle of the algorithm is to treat complete messages as targets, and incomplete messages as interferences. This is done by dividing the analysis window into two data blocks and by comparing the power of the messages concentrated in the two data blocks. The distinction between the target messages and the interference messages is given as follows. From the analysis window, two matrices can be formed: $\mathbf{X}_1: M \times N_1$ and $\mathbf{X}_2: M \times N_2$ as shown in Fig. 4.3 (data in the two matrices are downsampled by P from \mathbf{X})

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{H}_s \mathbf{S}_{s1} + \mathbf{H}_f \mathbf{S}_{f1} + \mathbf{N}_1 = \mathbf{H} \mathbf{S}_1 + \mathbf{N}_1 \\ \mathbf{X}_2 &= \mathbf{H}_s \mathbf{S}_{s2} + \mathbf{H}_f \mathbf{S}_{f2} + \mathbf{N}_2 = \mathbf{H} \mathbf{S}_2 + \mathbf{N}_2. \end{aligned} \quad (4.7)$$

where $\mathbf{H} = [\mathbf{H}_s \ \mathbf{H}_f]$ and $\mathbf{S}_i = [\mathbf{S}_{si}^T \ \mathbf{S}_{fi}^T]^T$, $i = 1, 2$. \mathbf{H}_s and \mathbf{H}_f are the channel matrices of the target messages and the interference messages, respectively. \mathbf{S}_{si} : $d_s \times N_i$ and \mathbf{S}_{fi} : $d_f \times N_i$ are the source data matrices containing the target messages and the interference messages, respectively. Two corresponding sample covariance matrices are formed as (the cross-correlation items are ignored)

$$\begin{aligned} \hat{\mathbf{R}}_1 &= \frac{1}{N_1} \mathbf{X}_1 \mathbf{X}_1^H \\ &\approx \frac{1}{N_1} (\mathbf{H}_s \mathbf{S}_{s1} \mathbf{S}_{s1}^H \mathbf{H}_s^H + \mathbf{H}_f \mathbf{S}_{f1} \mathbf{S}_{f1}^H \mathbf{H}_f^H + \mathbf{N}_1 \mathbf{N}_1^H) \\ \hat{\mathbf{R}}_2 &= \frac{1}{N_2} \mathbf{X}_2 \mathbf{X}_2^H \\ &\approx \frac{1}{N_2} (\mathbf{H}_s \mathbf{S}_{s2} \mathbf{S}_{s2}^H \mathbf{H}_s^H + \mathbf{H}_f \mathbf{S}_{f2} \mathbf{S}_{f2}^H \mathbf{H}_f^H + \mathbf{N}_2 \mathbf{N}_2^H) \end{aligned} \quad (4.8)$$

The distinction is defined as

$$\begin{cases} \alpha^2 \mathbf{S}_{s1} \mathbf{S}_{s1}^H > \mathbf{S}_{s2} \mathbf{S}_{s2}^H \\ \alpha^2 \mathbf{S}_{f1} \mathbf{S}_{f1}^H < \mathbf{S}_{f2} \mathbf{S}_{f2}^H \end{cases} \quad (4.9)$$

where α is a scaling factor to make the scaled power of the matrices satisfy the inequalities for given \mathbf{S}_{si} and \mathbf{S}_{fi} . The power of \mathbf{S}_{si} and \mathbf{S}_{fi} depends on the split parts of the messages. The value of α is determined from a marginal case of this splitting. In a marginal case, a target message should be detected when one part of this message fills up \mathbf{X}_1 and the other part fills up \mathbf{X}_2 (messages could occupy two time slots). Note that, at this moment, the noise satisfies the same criteria. The ambiguity between target messages and the noise is naturally removed by a threshold γ (introduced later). In this case, the length of \mathbf{S}_{s1} is N_1 and the length of \mathbf{S}_{s2} is N_2 . The first inequality in (4.9) is used to compute α . From the results of random matrix theory in Chapter 3, the lower bound of α is

$$\alpha > \frac{\sqrt{N_2} + \sqrt{M}}{\sqrt{N_1} - \sqrt{M}} \quad (4.10)$$

where the values of N_1 and N_2 are given in Section 4.5. Here, α can be set larger than the lower bound in some cases but is generally preferred to be as close to the lower bound as possible. The beamformer from the decomposition

$\text{SURV}^{j_\gamma \pm -}(\gamma \mathbf{I}, \alpha \mathbf{X}_1, \mathbf{X}_2)$ can be computed as

$$\mathbf{Q}^H [\gamma \mathbf{I} \alpha \mathbf{X}_1 \mathbf{X}_2] \Theta = \mathbf{Q}^H [\mathbf{A} \mathbf{0} | \mathbf{B} \mathbf{0}] = [\mathbf{R}_A \mathbf{0} | \mathbf{R}_B \mathbf{0}] \quad (4.11)$$

where the sign “ \pm ” and j_γ above the matrices denote the signatures of the columns of the matrices (explained in detail in [14]), Θ is a \mathbf{J} -unitary matrix, \mathbf{Q} is a unitary matrix, \mathbf{A} : $M \times d_s$ and \mathbf{B} : $M \times (M - d_s)$, \mathbf{R}_A : $M \times d_s$ and \mathbf{R}_B : $M \times (M - d_s)$ together are lower triangular, and γ is a threshold to balance the scaled noise power in $\alpha \mathbf{X}_1$ and \mathbf{X}_2 , and to keep the noise subspace in the negative part of the SURV. γ is given as

$$\gamma = \sqrt{|t|}, \quad j_\gamma = \text{sign}(t); \quad (4.12)$$

$$t = \sigma^2 \left[\left(\sqrt{N_2} - \sqrt{M} \right)^2 - \alpha^2 \left(\sqrt{N_1} + \sqrt{M} \right)^2 \right], \quad (4.13)$$

The separating beamformers is obtained as

$$\mathbf{W}^{(1)H} = \mathbf{Q}_A^H. \quad (4.14)$$

The tracking form implementation of (4.9) is

$$\text{SURV}([\mathbf{A}, \mathbf{x}_1, \alpha \mathbf{x}_4, \mathbf{x}_5], [\mathbf{B}, \mathbf{x}_2, \alpha \mathbf{x}_3, \mathbf{x}_6]). \quad (4.15)$$

This tracking algorithm is called GSVD-T. The updating of GSVD-T is the same as an ordinary updating of SURV [19,24]. When the analysis window is advancing 1 column, 6 column vectors need to be updated, 1 incoming and 1 leaving vectors for \mathbf{X}_1 and 2 incoming and 2 leaving vectors for \mathbf{X}_2 (See Fig. 4.3).

Table 4.1: Configuration of the Satellite AIS model.

| | First Scenario | Second Scenario |
|--------------------------|-------------------------------|-----------------|
| Carrier frequency | 162.025 MHz | |
| Num. of channels | 2 | |
| Channel bandwidth | 25 kHz | |
| Modulation | 9.6 kbps GMSK | |
| BT | 0.3 | |
| Satellite altitude | 600 km | |
| Satellite speed | 7561.65 m/s | |
| Orbit period | 5792.52 s | |
| Radius of FoV | 1396.25 nmi | |
| Ship visible time | 704 s per sat. pass | |
| Message length | 256 bits | |
| Empty message buffer | 10 bits | |
| Ship emission power | 12.5 W(Class A)/2 W(Class B) | |
| Ship transmit antenna | Half-wave dipole | |
| Sat. receive antenna | Array of directional elements | |
| Sat. antenna spacing | Half a wavelength | |
| Array spinning speed | 1 round/30 s | |
| Max. SNR at the receiver | 25 dB [94] | |
| Channel filter bandwidth | None | 15 kHz |
| Orbit inclination | None | 98° |
| Cell shape | Square | Geodesic grid |
| Cell size | 40 × 40 nmi ² | Radius 32 nmi |
| Num. of Cells in FoV | 5184 | Variable |
| Ship report interval | 6 s | Mixed |

Table 4.2: Composition of types of ships in different areas in the second scenario.

| Harbor | Percentage | Ship report interval |
|----------------|------------|----------------------|
| Moored | 24% | 3 min |
| Moving Class A | 1% | 2 s |
| | 4% | 6 s |
| | 63% | 10 s |
| Moving Class B | 8% | 30 s |
| Coastline | Percentage | Ship report interval |
| Moving Class A | 60% | 6 s |
| | 40% | 10 s |
| Ocean | Percentage | Ship report interval |
| Moving Class A | 100% | 3 min |

4.5 Simulation Results

In this section, the performance of the receivers are compared in two simulation scenarios. Four receivers are compared and they are

1. *GSVD-T+ACMA*: The receiver proposed in Section 4.3.
2. *GSVD-SI+ACMA*: The receiver proposed in Chapter 3 and [23].
3. *ACMA*: The proposed receiver but without the first stage processing.
4. *ESPRIT+Capon*: The receiver using ESPRIT and Capon beamforming in [91].

The two scenarios share some common setup parameters. An AIS receiver on a LEO satellite orbiting at altitude 600 km high is assumed. On the ground, the SOTDMA protocol is implemented in each communication cell. The scenario has two frequency channels but only one frequency channel is simulated here. The ship report interval is defined under the condition of enabling the two frequency channels for Terrestrial AIS (No newly assigned AIS channels for satellite AIS are taken into account). The report interval for one frequency channel is the number of frequency channels times the ship report interval. The transmission time of messages follows a poisson process in a given report interval, and the intensity of this poisson process changes with the number of ships within the satellite field of view (FoV). The typical channel effects, such as path loss, path delays and Doppler frequency shifts, are modeled. A rotating linear antenna array of M directional elements are used as the receive antennas. The antennas stand perpendicularly to FoV and rotate around the center of the satellite. The radiation pattern of the antennas draw a curve of $\cos^2\theta$ in half of the FoV, where θ is DOA and $-0.5\pi \leq \theta \leq 0.5\pi$. The array only receives signals from half of FoV at any time as a result of the directivity of the element antennas. Thus, the rotation of the satellite is necessary to avoid the signals from Terrestrial AIS saturating the receiver. The main difference of the two scenarios is the type of ship distribution within FoV. The detailed parameters are listed in Table 4.1. One performance measure is the message detection probability (MDP), the ratio of the number of successfully decoded messages to the number of messages sent. Another performance measure is the ship detection probability (SDP), the ratio of the number of detected ships to the number of ships covered by the footprints of the satellite. For MDP, the performance of all the receivers do not exceed 0.5 because of the coverage of half of FoV at any time. For SDP, the performance of all receivers do not reach 1 because ships have different visible time period to the satellite, ranging from zero to time period of one satellite passing. Note that, MDP and SDP are closely related because a higher MDP often means that SDP grows faster in a given visible time period. For *GSVD-T+ACMA*, the choices of N_1 and N_2 are $N_1 = 0.25N_p$, $N_2 = N_p$, and α is set to 2.5 times of its lower

bound. For fair comparison, the minimum spacing between analysis windows is set to $0.25N_p$ for all the receivers.

In the first simulation scenario, the positions of ships have a uniform distribution within FoV. All the ships are randomly spread on the surface of FoV and are moving in the opposite direction against the satellite velocity vector. The number of ships within FoV is kept constant.

Fig. 4.5(a) and Fig. 4.5(b) show the MDP and the SDP of the compared receivers, respectively, as a function of the number of antennas when the number of ships within FoV is 5,000. It is a scenario with medium ship density within FoV. It is seen from both figures that the receivers using the blind source separation (BSS) algorithms, including GSVD-T+ACMA, GSVD-SI+ACMA and ACMA, exhibit better performance than the classic algorithm, ESPRIT+Capon. Among the BSS algorithms, GSVD-T+ACMA has the lowest computational complexity and the best performance. ACMA is not approaching 1 while the number of antennas grow because no suppression of asynchronous interference messages is done. Moreover, without removing the asynchronous interference messages, ACMA usually has a larger rank estimate d_s than GSVD-T+ACMA and GSVD-SI+ACMA, which dramatically increases the computational complexity.

Fig. 4.6 shows results similar to Fig. 4.5. The number of ships, 10,000, is a worst case in Satellite AIS, and the MDP and SDP are reasonably lower than the case of 5,000 ships. The results show that more antennas are needed to increase the MDP and SDP to a satisfactory level. Again, GSVD-T+ACMA gives the best performance among all the receivers.

In the second simulation scenario, the practical global ship distribution and the satellite footprints are taken into account. The ship distribution is shown in Fig. 4.1 and the composition of types of ships is listed in Table 4.2. Note that the physical moment is when the satellite is passing over the north coast of Europe. This area is one of the busiest areas around the world. In one satellite footprint (alternatively, one FoV) in this scenario, it is seen that the ship density along the coastlines is very high (over 10,000 ships), and the density in the open sea is relatively low (less than 5,000 ships). Different types and status of ships as well as the corresponding report intervals are considered in this simulation. The parameters of this simulation are listed in Table 4.1.

Fig. 4.7(a) and Fig. 4.7(b) show the number of detected messages and the number of detected ships in this simulation, respectively. The MDP and SDP are too small and have little meanings here. It is seen that GSVD-T+ACMA still detects the most messages and ships, which is about twice of that detected by ESPRIT+Capon. It is still attractive to use GSVD-T+ACMA when fewer antennas are used, i.e., 2 or 4 antennas. It can be seen that, ACMA noticeably drops when the number of antennas exceeds 8 and even gives worse performance than ESPRIT+Capon, which is a strong support for using the BBT proposed in this paper in production.

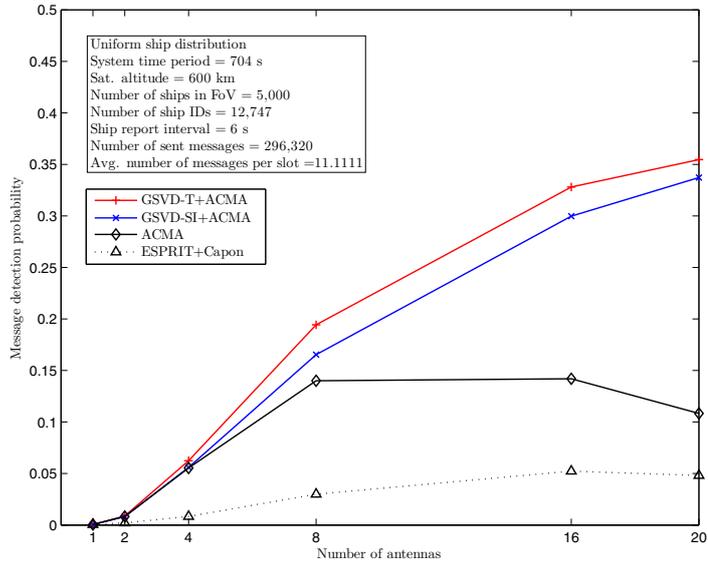
Now, it is the time to provide some visual comparison between the performance of GSVD-T+ACMA and ESPRIT+Capon. Fig. 4.8(a) and Fig. 4.8(b)

illustrate the positions of the ships detected by the two receivers within the satellite's footprints on the surface of the earth, respectively. The antenna array has 4 elements. From the results, it is seen that GSVD-T+ACMA detects more ships than ESPRIT+Capon, especially those at the edge of FoV, where the incoming signals are weak. In other words, ESPRIT+Capon has problems in detecting weak signals as the errors in DOA estimation increase [95]. For the inland area, the ships are almost impossible to be detected: the ship density is too high [94]. This problem can be slightly eased by using more antennas, but this measure is very limited because the computational complexity is also increased. Fortunately, the ships in the inland area can be detected by Terrestrial AIS, which can compensate Satellite AIS.

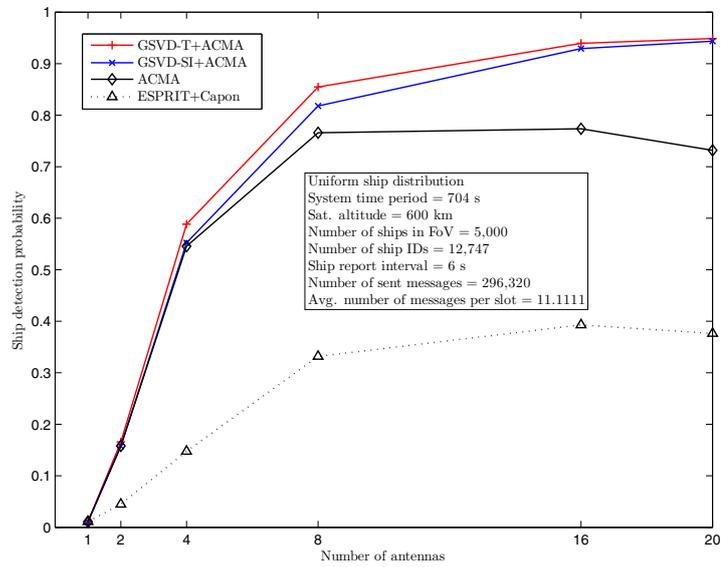
4.6 Conclusions

In this chapter, a multi-user receiver for Satellite AIS has been built. The proposed receiver uses the blind beamforming technique of Chapter 3 to effectively improve the detection of AIS messages under asynchronous interferences. The beamforming algorithm can be easily implemented in a tracking form through SURV and the complexity is reduced. The whole receiver uses "blind" algorithms and requires less knowledge of the beam pattern, which facilitates the design of antenna arrays and satellites.

Simulation results in two scenarios of Satellite AIS confirm the effectiveness of the proposed receivers and show it being superior to previous receivers.

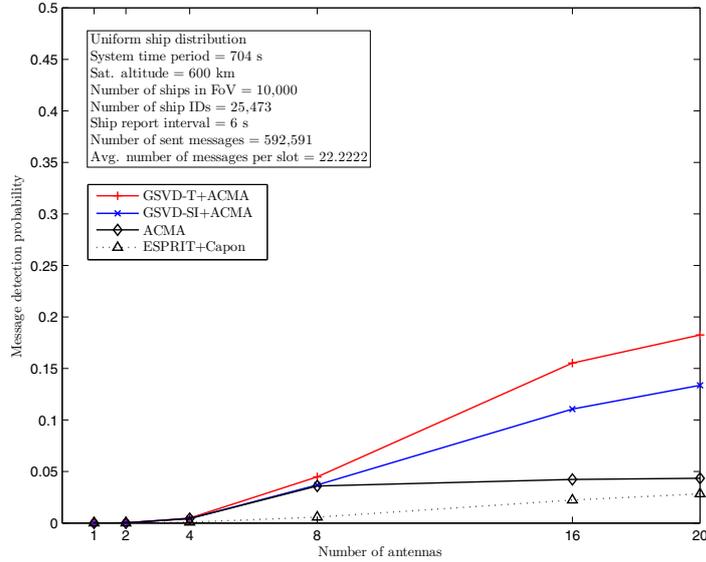


(a)

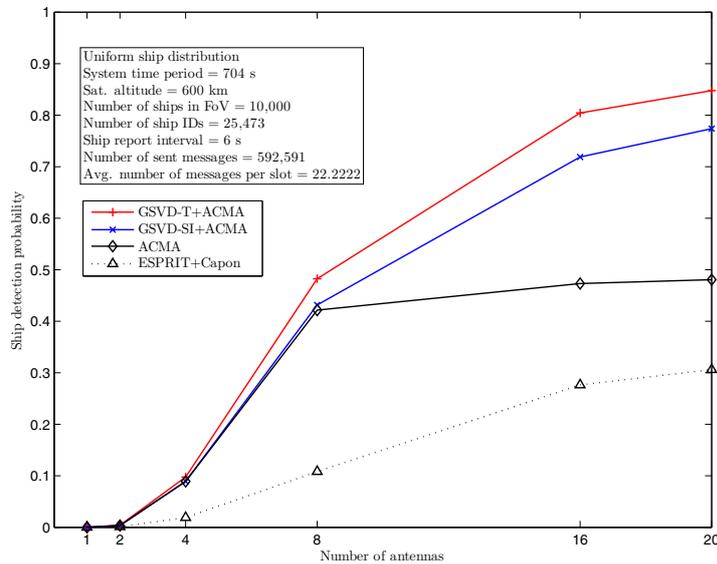


(b)

Figure 4.5: Performance of receivers when 5,000 ships are uniformly distributed within FoV. (a) Message detection probability of receivers as a function of the number of antennas; (b) Ship detection probability of receivers as a function of the number of antennas.

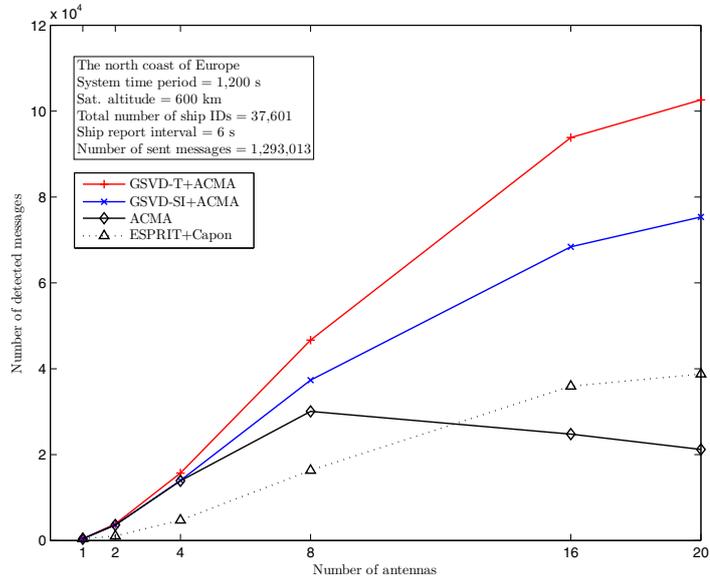


(a)

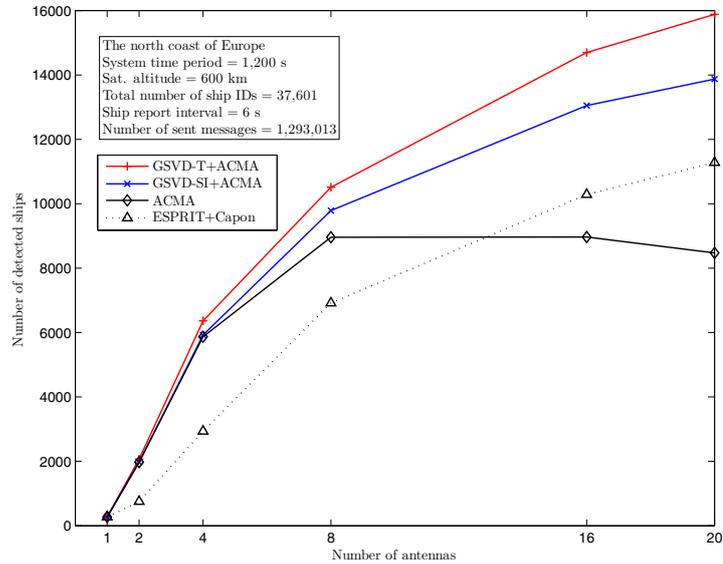


(b)

Figure 4.6: Performance of receivers when 10,000 ships are uniformly distributed within FoV. (a) Message detection probability of receivers as a function of the number of antennas; (b) Ship detection probability of receivers as a function of the number of antennas.

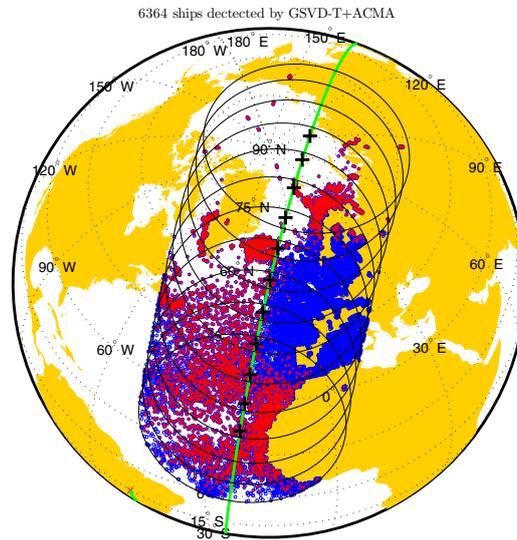


(a)

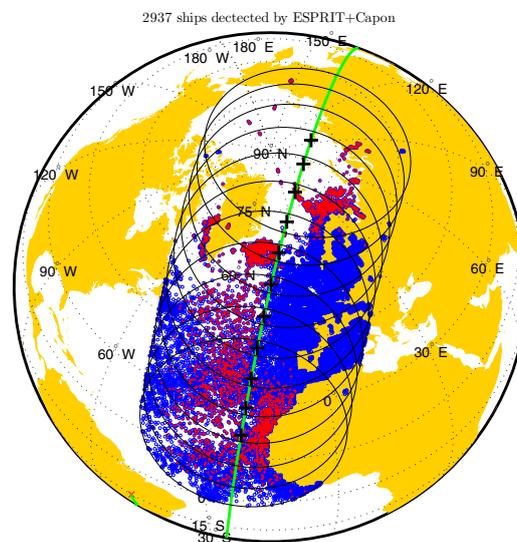


(b)

Figure 4.7: Performance of receivers when the satellite flies over the north coast of Europe. (a) Number of detected messages of receivers as a function of the number of antennas; (b) Number of detected ships of receivers as a function of the number of antennas.



(a)



(b)

Figure 4.8: Display of the positions of the ships detected by GSVD-T+ACMA and ESPRIT+Capon when the satellite flies over the north coast of Europe. Four antennas are used. (Not every footprint is shown. Only ships within the footprints are shown.) The blue circles denote the ships. The red signs “+” denote the detected ships. (a) GSVD-T+ACMA; (b) ESPRIT+Capon.

Chapter 5

SystemC-AMS Model of a Dynamic Large-Scale Network of Satellite AIS

5.1 Introduction

This chapter introduces the software testbed for satellite AIS. This software testbed is a tool for verifying and validating the algorithms proposed in former chapters.

AIS [96] is a wireless communication system in the VHF maritime mobile band to facilitate the efficient exchange of navigational data between ships and between ships and shore base stations. Recently, the concept of Satellite AIS is introduced to improve the capability of detecting and tracking ships at distances far away from coastlines, which cannot be accomplished by Terrestrial AIS stations. LEO satellites [4–6, 97] are capable of carrying AIS receivers to expand the scope of services. The old-fashioned AIS receivers are only reliable in sparsely trafficked areas, and hence, the new-style AIS receivers to be applied in very busy areas urgently need the support of advanced technologies.

In Satellite AIS, LEO satellites face technical challenges such as high Doppler frequency shifts, high variance of path loss, large relative time delays in time slots, and severe collisions of AIS messages. These challenges either do not happen or are not considered in Terrestrial AIS. As the old-fashioned AIS receivers do not function well in Satellite AIS, some advanced multi-user AIS receivers (MU) [13, 16, 23] were designed to improve the service quality of Satellite AIS. The receivers can be equipped with antenna arrays to combat the challenges. To know the performance of the receivers in Satellite AIS during the design period, the receivers need to be systematically simulated in a virtual environment approximating the

⁰Part of this chapter was published in the 21st annual workshop on circuits, systems and signal processing-ProRISC, Nov. 2010. [21]. The presentation of this chapter is kept self-contained as a reader can start reading without going over the former chapters.

real scenario of the satellite in space [98]. This work is very important for saving the cost and reducing the risk in further implementation of the receivers.

A functional model producing the full effects of the transmission through the physical channel between the ships and the LEO satellite is built in this chapter. This model is a wireless communication network. In this network, the number of functional modules associated with the ships can be over ten thousands. The whole network needs to be precisely timed at the signal level (physical layer) and the slot level (link layer) according to the SOTDMA protocol. Both the dynamical movement and the FoV coverage of the LEO satellite on the practical global distribution of ships need to be constructed at a certain level of details in the physical channel. The model should possess the ability of flexible reconfiguration and extension for future development.

SystemC-AMS [99] is a modeling tool to meet the requirements. The timed data flow (TDF) of SystemC-AMS provides a precise TDF in an extendible structure and an open function for customized modules [17, 100–107]. With the help of SystemC-AMS, one can naturally model the real-time transmission for the complex system. The transmitters, the wireless channel and the satellite receiver are seen as the corresponding TDF modules of SystemC-AMS in the signal flow of a single TDF cluster. In the signal flow, the AIS messages are encoded, modulated and transmitted according to the AIS standard, distorted by the channel effects, and received by the satellite receiver. The model allows an option to natively implement and test the algorithms although this part of work is done in MATLAB in early development stages. Behind the signal flow are parameters from the mechanism of the satellite orbiting, the practical global ship distribution and the protocol of SOTDMA, which are updated regularly and managed by a system controller.

This chapter gives an overview of the proposed simulation system while details are omitted because it would lead to a lengthy system manual.

Outline

This chapter is organized as follows. In Section 2, the mechanism behind the model is introduced. In Section 3, the SystemC-AMS model is discussed in detail. In Section 4, the simulation setup and results are provided. Section 5 concludes this chapter.

5.2 Mechanism behind the Model

In this section, the mechanism behind the model is presented. Five aspects of the mechanism are covered: the physical layer of AIS, the link layer of AIS (i.e., the SOTDMA protocol), physical channel effects, the satellite orbit design and the global ship distribution. The model is built on the top of proper abstraction of the mechanism.

5.2.1 AIS Physical Layer

In the model, the international standard of AIS [83] is followed as Satellite AIS do not change the protocol of Terrestrial AIS. The algorithms for the receivers in Satellite AIS are designed at the physical layer. The accurate function of the physical layer of AIS needs to be modeled. At the transmitter side, this includes both the encoding and the modulation of AIS messages. An AIS message is typically $N_p = 256$ bits long (See Fig. 4.2). The whole encoding process consists of embedding navigational information (geographical coordinates) into the data part of messages, computing the frame check sequence (FCS), which is a 16-bit cyclic redundancy code (CRC), bit-stuffing (a zero is inserted after every five continuous 1s), and assembling the training sequence (TS), the start flag (SF) and the end flag (EF). The whole modulation consists of follows mapping the binary sequence $\{b_n\}_{1 \leq n \leq N_p}$ of the AIS message into $\{+1, -1\}$, differentially encoding the sequence (NRZI encoding) as $a_n = b_n a_{n-1}$, $n \geq 1$, $a_0 = +1$, modulating the sequence by Gaussian minimum shift keying (GMSK) [1], and transmitting the modulated AIS signal at 9.6 kbps on carriers at frequency 162 MHz. The baseband representation of the transmitted AIS signals has a constant envelope form as

$$\tilde{s}(t) = e^{j\theta(t)}, \quad (5.1)$$

where $\theta(t)$ is the phase given by

$$\theta(t) = \pi h \int_{-\infty}^t \sum_{n=-\infty}^{+\infty} a_n g(\tau - nT) d\tau, \quad (5.2)$$

where h is the modulation index equal to 0.5, and $g(t)$, a pulse of unit area, is the response of a Gaussian filter to a unit amplitude rectangular pulse of duration T , where T is the symbol period. The GMSK modulation is described by the bandwidth-time product BT , where B is the bandwidth of the Gaussian filter. The BT values used by AIS are limited between 0.3 and 0.5. The default AIS message has power-up and power-off transient periods at the head and the tail of the message. The end of the message filled by the last 10 bits is normally considered empty with zero RF power. At the receiver side, the process is a simply inverted flow of the transmitter side if a single user receiver (SUR) is used. The processing can become more complex when an advanced multi-user receiver is used. In MU, SUR sits at the end of the signal flow. The channel bandwidth is an important factor of the physical channel. The baseband channel bandwidth could be 12.5 kHz or 25 kHz depending on local regulation. The signal bandwidth is assumed to be 25 kHz in this chapter. Some commercial receivers [108] use a narrow baseband filters with 15 kHz bandwidth to get better filtering. If such receivers were used in Satellite AIS, then the filtering would suppress those signals associated with large Doppler frequency shifts.

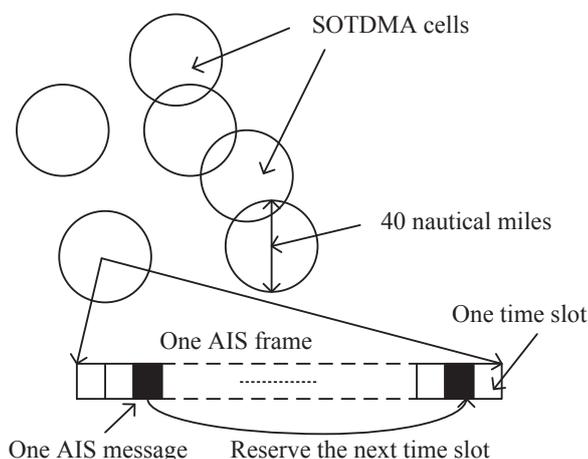


Figure 5.1: SOTDMA cells and the SOTDMA protocol.

5.2.2 AIS Link Layer

The AIS link layer arranges the transmission of AIS messages within an SOTDMA cell (See Fig. 5.1). An AIS frame equals one minute and 2250 time slots. In one SOTDMA cell, the wireless channel is open to all users so that every user can hear the AIS messages transmitted by the other users. A user listens to the traffic in an AIS frame prior to any operation, and then selects one empty time slot to transmit and reserves the next time slot. The time interval is fixed, e.g., 6 seconds or 1 minute. All the stations in all the cells are synchronized to universal time control (UTC), e.g., GPS. The AIS frame and time slots are aligned to UTC. Currently, Satellite AIS does not intervene the activity of Terrestrial AIS. The satellite receiver plays a passive role and silently listens to the activity in AIS.

5.2.3 Physical Channel Effects

In the model, not only the standard AIS protocol but also the physical channel effects brought by LEO satellites are taken into account.

In Satellite AIS, the mutual transmission link expands from that between ships and base stations in vicinity to that between a large number of ships spread out in many cells and to a fast-moving LEO satellite across the sky.

The AIS protocol for managing the transmission loses its effectiveness in this scenario. The AIS messages arriving at the satellite receiver cannot be synchronized due to the different propagation time of the messages.

The received power of AIS signals varies in a large range: it is a natural effect (this allows frequency reuse in cells) for Terrestrial AIS but a negative effect (weak and strong signals are both wanted) for Satellite AIS.

The high-speed movement of the LEO satellite creates distortion in the frequency domain: Doppler frequency shifts occur. Large deviation from the central frequency causes troubles in tracking the frequency of the received signals. The baseband filter attenuates the signals close to the margin of its passband. Such weak signals can be strengthened by antenna arrays.

The collision of messages also requires the use of antenna arrays to allow source separation. The advanced multi-user receivers are indeed equipped with such antenna arrays. In this case, an additional channel effect is introduced for regular arrays, the direction of arrival (DOA) of AIS signals imping on the antennas array.

A proper parametrization of the model can make the entire structure of this model extendable and reconfigurable. After careful examining of this problem, it is found that all the channel effects are all functions of the relative positions between the considered ship and the satellite. Before the detail of this relation is revealed, the movement of the satellite is analyzed.

Consider a single satellite in the low earth orbit at altitude h_{sat} (See Fig. 5.2). Fig. 5.2 illustrates the orbiting scenario, where the view is taken on the observed plane constrained by three objects: the observed ship, the satellite and the nadir. In the figure, the red dotted line denotes the satellite orbit which is in another plane intersecting the observed plane. In the satellite field of view (FoV), ships frequently send out AIS messages to the satellite to report their navigational information. The coordinate system for the satellite is right-handed:

1. The x-axis is parallel to the velocity vector of the satellite.
2. The y-axis is parallel to the FoV and orthogonal to the velocity vector of the satellite .
3. The z-axis is pointed to the nadir.

In this coordinate system, the i th ship has an azimuth seen from the satellite φ_i and a ground range away from the nadir r_i (See Fig. 5.3). (φ_i, r_i) is the parameter pair that uniquely determines the channel effects of the received AIS signals from the i th ship. For the satellite, some parameters associated with the satellite orbit are constants independent of the positions of ships. They are calculated as follows.

The distance between the satellite and the earth center is

$$r_s = h_{sat} + r_e, \quad (5.3)$$

where $r_e = 6371.15\text{km}$ is the radius of the earth. The ships are distributed in a flat FoV (on a disk), which is a projection of the FoV (a sphere cap) towards the center of the earth. Then the positions of ships are projected back to the FoV. The radius of the flat FoV is

$$r_{FoV} = r_e \arccos\left(\frac{r_e}{r_s}\right). \quad (5.4)$$

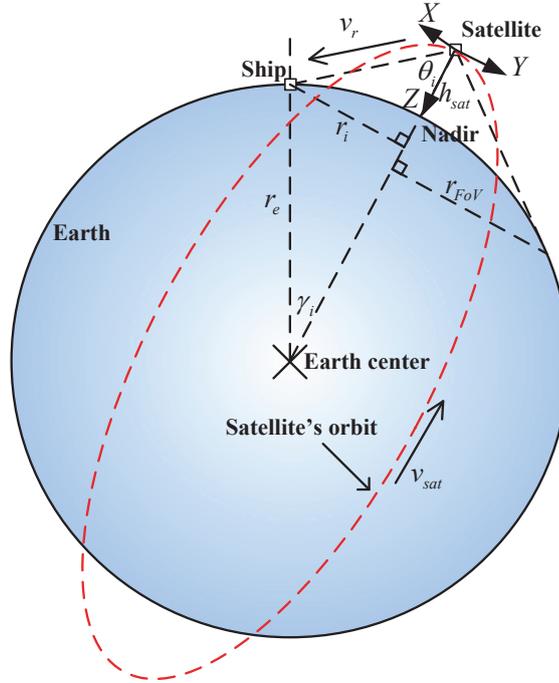


Figure 5.2: The satellite's orbit and the geometric relation between the satellite and the observed ship.

The velocity of the satellite in orbit is

$$v_{sat} = \frac{2\pi r_s}{T_{sat}}, \quad (5.5)$$

where T_{sat} is the period of the satellite orbit known as

$$T_{sat} = \sqrt{\frac{4\pi^2 r_s^3}{\mu}}, \quad (5.6)$$

where the constant $\mu = 3.986 \times 10^5$ is the product of the gravitational constant G and the mass of the earth M_e .

Doppler frequency shift

The Doppler frequency shift $\Delta f_i(\varphi_i, r_i)$ associated with the signal from a ship is given by

$$\Delta f_i(\varphi_i, r_i) = \frac{v_{ir}}{\lambda}, \quad (5.7)$$

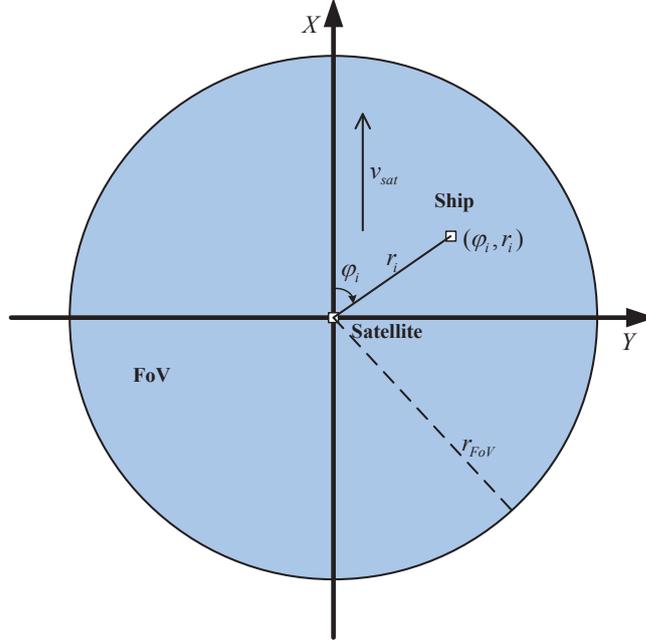


Figure 5.3: The top view of the satellite field of view.

where v_{ir} is the relative velocity between the i th ship and the satellite, and λ is the carrier wavelength given by

$$\lambda = \frac{c}{f_{carrier}}, \quad (5.8)$$

where c is the speed of light, $f_{carrier}$ is the carrier frequency. The formula for v_{ir} is

$$v_{ir} = v_{sat} \cos \varphi_i \sin \theta_i, \quad (5.9)$$

$$\theta_i = \arcsin \left(\frac{r_i}{d_i} \right), \quad d_i = \sqrt{r_s^2 + r_e^2 - 2r_s r_e \cos \gamma_i}, \quad \gamma_i = \arcsin \left(\frac{r_i}{r_e} \right), \quad (5.10)$$

where θ_i is the satellite elevation angle for the i th ship, and d_i is the slant range between the i th ship and the satellite. The distorted carrier frequency of signals seen by the satellite is given by

$$f_i(\varphi_i, r_i) = f_{carrier} + \Delta f_i(\varphi_i, r_i). \quad (5.11)$$

In the baseband, Doppler frequency shifts produce Doppler phase shifts of the signals. The signals are rotated as a function of time as

$$s(t) = \tilde{s}(t) e^{j2\pi \Delta f_i(\varphi_i, r_i) t}. \quad (5.12)$$

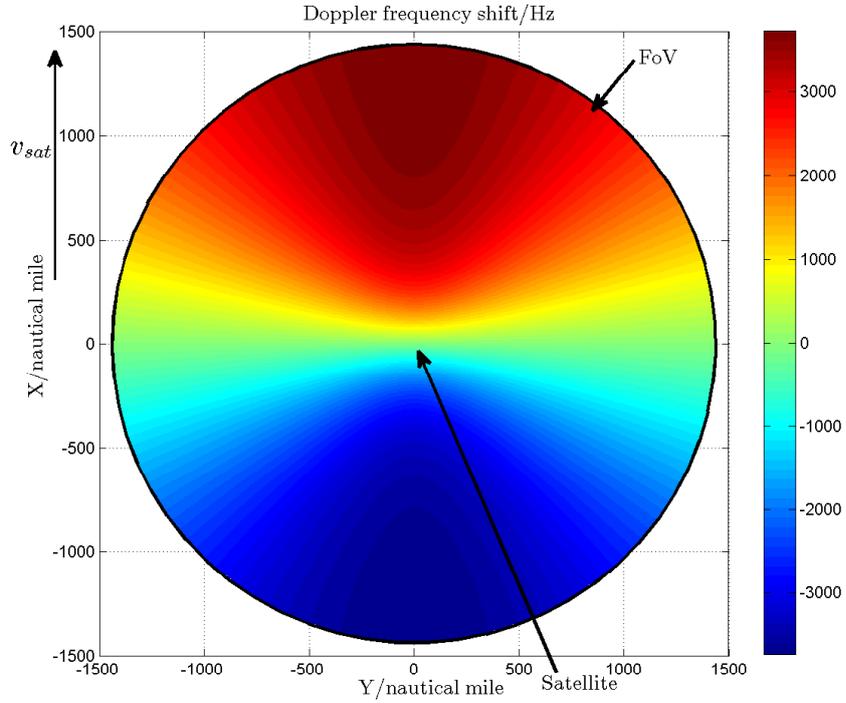


Figure 5.4: The Doppler frequency shifts of the ships located in FoV seen by the LEO satellite.

For a satellite at $h_{sat} = 600\text{km}$, the maximum Doppler frequency shift seen by the satellite is $\pm 3.7\text{kHz}$ for a carrier at frequency $f_{carrier} = 162.025\text{MHz}$ (See Fig. 5.4).

Propagation time delay

The propagation time delay of the message transmitted by the i th ship is

$$\tau_i = \frac{d_i}{c}. \tag{5.13}$$

The relative time delay of the received signals against the signals from the nadir seen by the LEO satellite (See Fig. 5.5) are more useful. The relative time delay is calculated in terms of a ratio of the time length of one time slot T_{slot} given by

$$\tau_{im} = \frac{\text{rem}(\tau_{ir}, T_{slot})}{T_{slot}}, \tag{5.14}$$

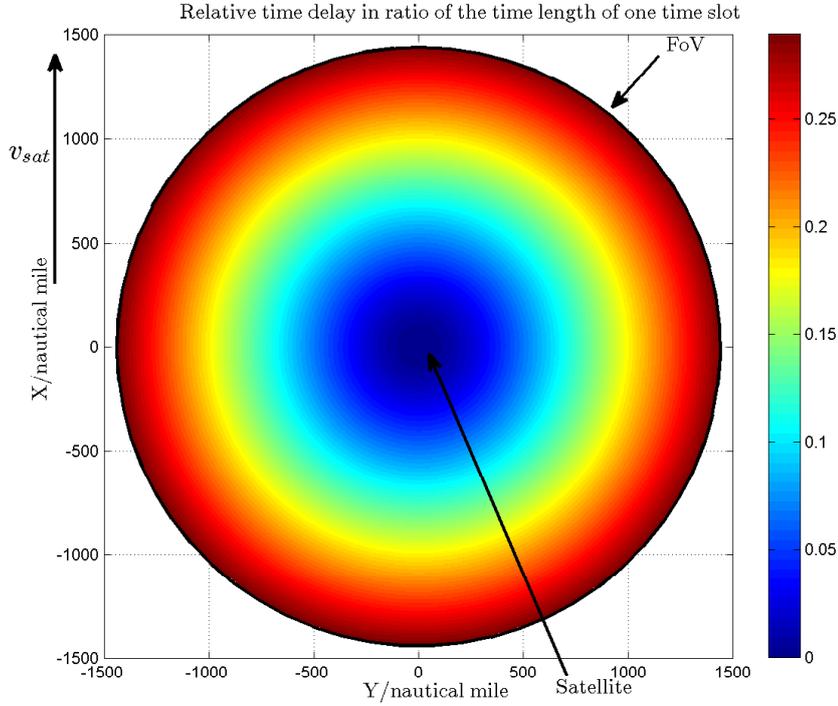


Figure 5.5: The relative propagation time delays of the received signals seen by the LEO satellite compared with the signals from the nadir in terms of a ratio of the time length of one time slot.

where $\text{rem}(a, b)$ is the remainder of a modulo b , $\tau_{ir} = \frac{d_i - h_{sat}}{c}$ and $T_{slot} = 0.026667$ ms.

Path loss and received signal power

The path loss is the loss of the signal power along the propagation path. The calculation of the path loss depends on the type of transmit and receive antennas. Assume that each ship is equipped with one perpendicular half-wave dipole transmit antenna, and the satellite is equipped with omnidirectional circularly polarized half-wave dipole receive antennas. The path loss $L(r_i)$ is a function of r_i because of the geometric symmetry of the directivity of the used antennas. The received signal power is split as the addition of three values

$$P_i(r_i) = P_t(r_i) + L(r_i) + G_a, \quad (5.15)$$

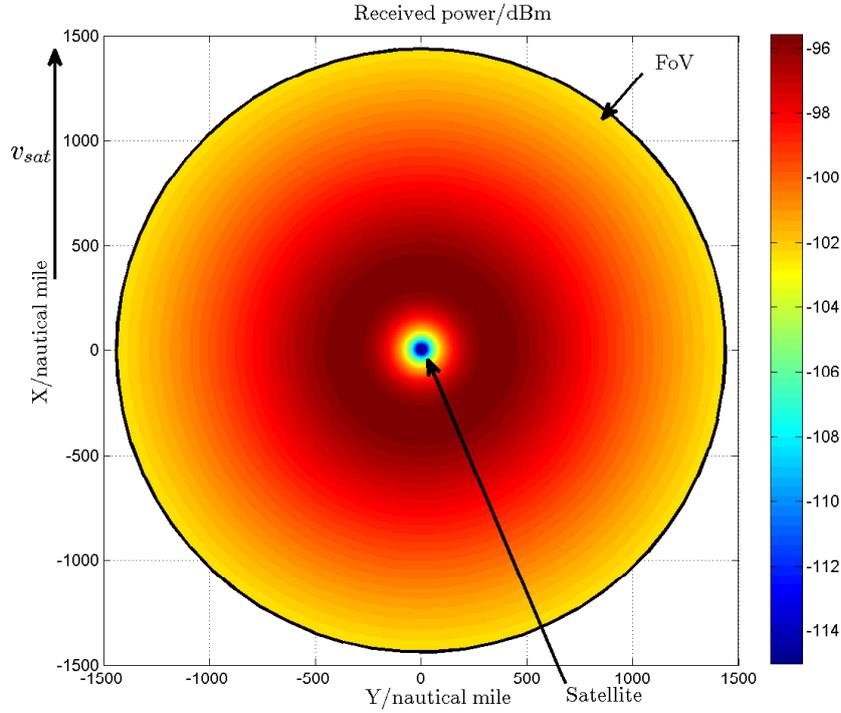


Figure 5.6: The received signal power of the ships seen by the LEO satellite (The minimum value limited to -115 dBm).

where $P_t(r_i)$ is the transmit power from a ship in dBm, G_a is the satellite antenna gain (unit gain but a constant 3 dB polarization loss is added [5]). In this simulation, the received signal power is virtually gained by a value G_{AGC} to bring the value close to numerical “1.0”. Since the load of online computation of the received signal power is heavy and not acceptable, a look-up table precomputed from MATLAB routines is adopted [5] instead. The index of the look-up table is r_i and the resolution of the index is 1 nmi (See Fig. 5.6).

DOA

A regular antenna array is chosen as the receive antennas. This allows the calculation of the direction of arrival (DOA) of signals. The purpose of this setup is for comparing the classic beamforming algorithms with the blind beamforming algorithms.

The array is positioned parallel to the plane of FoV (The array is simpler than

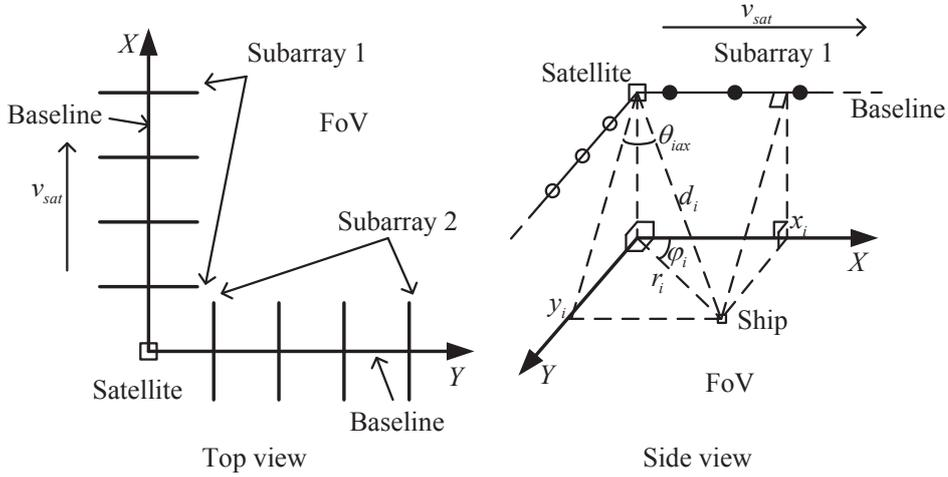


Figure 5.7: The setup of the satellite antenna array and the DOAs (θ_{iax} is shown for example. θ_{iax} is similar to θ_{iax}).

and different from the one in Chapter 4. The setup in this chapter is for validation of the model.) This array consists of two uniform linear subarrays, each of which has $m/2$ element antennas with separation of 0.5 wavelength, where m is the total number of element antennas. The baseline of one of the two subarrays is parallel to the x-axis (the satellite velocity vector) and its antennas are parallel to the y-axis. The baseline of the other subarray is parallel to the y-axis and its antennas are parallel to the x-axis. This setup takes the advantages of DOA from two orthogonal directions along the x-axis and the y-axis, respectively. Since the ships sit at the far filed of the arrays, the satellite antennas are treated as points. The polarization loss of signals coming from different satellite elevation angles is assumed to be a constant equal to 3 dB if the receive antennas are circularly polarized [5]. The DOA of the incident wave of the signals from the ships on the satellite antenna array is calculated by (See Fig. 5.7)

$$\theta_{iax}(\varphi_i, r_i) = \arcsin\left(\frac{x_i}{d_i}\right) = \arcsin\left(\frac{r_i \cos \varphi_i}{d_i}\right), \quad (5.16)$$

$$\theta_{iax}(\varphi_i, r_i) = \arcsin\left(\frac{y_i}{d_i}\right) = \arcsin\left(\frac{r_i \sin \varphi_i}{d_i}\right), \quad (5.17)$$

where $x_i = r_i \cos \varphi_i$ and $y_i = r_i \sin \varphi_i$ are the xy coordinates of the ships in FoV. Each DOA is located in the plane determined by the baseline of one subarray and the slant range of a ship. The DOAs θ_{iax} as a function of the positions of ships in FoV are shown in Fig. 5.8. For θ_{iax} , the figure is a 90-degree clockwise rotation of Fig. 5.8. Other setup of the array can be modeled in a similar way.

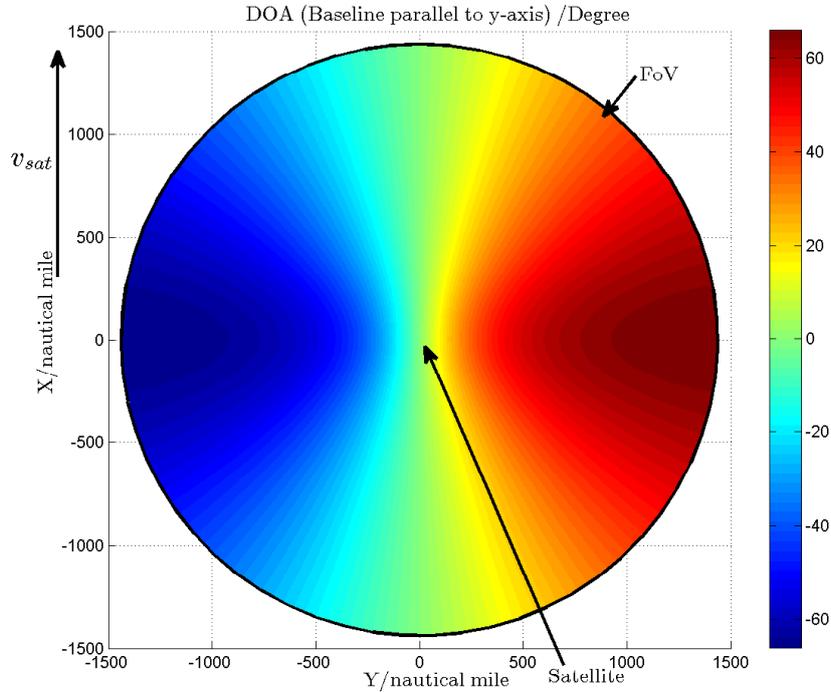


Figure 5.8: The DOA θ_{iay} as a function of the positions of the ships in FoV.

Collision of messages

The satellite sees more than one SOTDMA cells and the transmission is not organized across cells. The messages from different cells are transmitted into the same time slot, leading to colliding at the side of the satellite (See Fig. 5.9). When the collided messages have different DOAs, beamforming techniques (source separation) can be used for separating them.

Baseband channel

The transmission link is modeled at the baseband. The channel effects are precisely described without modulation onto and demodulation from the samples of the carrier at 162 MHz. This abstraction saves time cost in simulation. The channel is a multiple-input and multiple-output (MIMO) channel. The channel has input signal streams from the ship transmit antennas and output signal streams to the satellite receive antennas. Mathematically, a classic data model of the baseband channel is now introduced. Assume that at time t , there are d nonzero

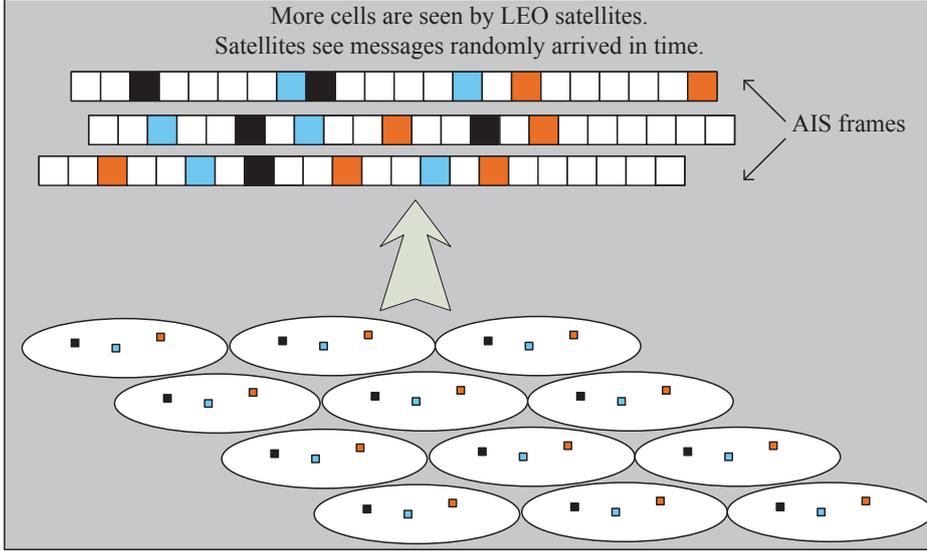


Figure 5.9: *The message collision seen by the satellite.*

signals impinging on the satellite antenna array of m elements. The active signal transmitted by the i th ship is shown as the k th signal in Eq.5.18. The index i is found back by a function $i = I(k)$. The received data samples from each antenna at time t are stacked into a column vector $\mathbf{x}(t) \in \mathbb{C}^M$ as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{n}(t), \quad (5.18)$$

where the columns of $\mathbf{H} \in \mathbb{C}^{M \times d}$ are the channel vectors (associated with DOAs) of each signal, the rows of $\mathbf{s}(t) \in \mathbb{C}^d$ are the data samples of each active signal of (5.12), and $\mathbf{n}(t) \in \mathbb{C}^M$ is the noise vector. The noise is assumed to be white Gaussian with covariance matrix $\mathbf{R}_n = \mathbb{E}(\mathbf{n}(t)\mathbf{n}(t)^H) = \sigma_n^2 \mathbf{I}$, where $\mathbb{E}(\cdot)$ is the expectation operator and σ_n^2 is the noise power measured at the baseband. The channel matrix is decomposed into $\mathbf{H} = \mathbf{A}\mathbf{B}$. Here, the matrix \mathbf{A} is the steering matrix defined as

$$\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_d], \quad \mathbf{a}_k = \begin{bmatrix} \mathbf{a}_{I(k)ay} \\ \mathbf{a}_{I(k)ax} \end{bmatrix}, \quad 1 \leq k \leq d, \quad (5.19)$$

$$\mathbf{a}_{I(k)ay} = \begin{bmatrix} \varphi_a^0(\theta_{I(k)ay}) \\ \vdots \\ \varphi_a^{m/2-1}(\theta_{I(k)ay}) \end{bmatrix}, \quad \mathbf{a}_{I(k)ax} = \begin{bmatrix} \varphi_a^0(\theta_{I(k)ax}) \\ \vdots \\ \varphi_a^{m/2-1}(\theta_{I(k)ax}) \end{bmatrix} \omega, \quad (5.20)$$

$$\omega = \frac{\varphi_a^L(\theta_{I(k)ax})}{\varphi_a^L(\theta_{I(k)ay})}, \quad (5.21)$$

where $\varphi_a^k(\theta) = e^{j2\pi k\Delta \sin \theta}$ is the phase shift between adjacent antennas in one subarray and ω is the phase shift between the two subarrays with $L = 10$. Also, the matrix \mathbf{B} is the matrix of signal power given by

$$\mathbf{B} = \text{diag}\{b_k\}_{1 \leq k \leq d}, \quad (5.22)$$

where $b_k = \frac{\sqrt{2}}{2} \sqrt{10^{(P_k + G_{AGC})/10}}$ is the power of the i th active signal and $P_k = P_{I(k)}(r_{I(k)})$ is the corresponding power in dBm. After the signals are uniformly sampled, the time variable t is replaced by $t = nT_s$ in (5.12) and (5.18), where $T_s = 1/f_s$ and f_s is the sampling frequency. The oversampling ratio $P = \frac{T}{T_s}$ is added to provide a higher sampling resolution in the time domain, where T is the symbol period of signals.

5.2.4 Ship Distribution and Satellite Orbit Design

The FoV is one of the satellite footprints. The FoV has incoming ships and leaving ships while the satellite is moving. Two scenarios are needed. One is a uniform ship distribution with constant number of ships in FoV. The SOTDMA cells are square. The other is a practical ship distribution with variable number of ships in FoV. The SOTDMA cells are constructed from geodesic grids, most of which are hexagons. In the second scenario, a polar earth orbit similar to launched satellites are used. The first scenario will be introduced in this chapter. The second scenario is already presented in Chapter 4.

5.3 C++ Model

SystemC-AMS provides a platform for functional modeling. The TDF modeling style, which is one of the modeling styles in SystemC-AMS, is a useful tool for the modeling of Satellite AIS. TDF allows precisely timed clusters consisting of customized TDF modules. In TDF, ports connect modules. The modules and ports can have different sample rates, which is convenient for high level modeling. TDF modules allow open accesses from/to general C++ classes. As an extension for SystemC, SystemC-AMS modules can be integrated into SystemC modules. In this case, the mixed-signal simulation of analog and digital modules is possible.

The system of Satellite AIS is properly divided. Most parts of the system are modeled by using TDF. A single TDF cluster is established for modeling the activity in the physical layer and the link layer of AIS. One system controller handles the cross-layer control. Several general C++ classes are responsible for the management of the system configuration, the system status, the processing inside the modules, and the computation of dynamic parameters of the model (for protocols and channel effects).

The structure of the model is shown Fig. 5.10, where the black solid lines denote the signal flow, the yellow dash lines denote the control flow, and the

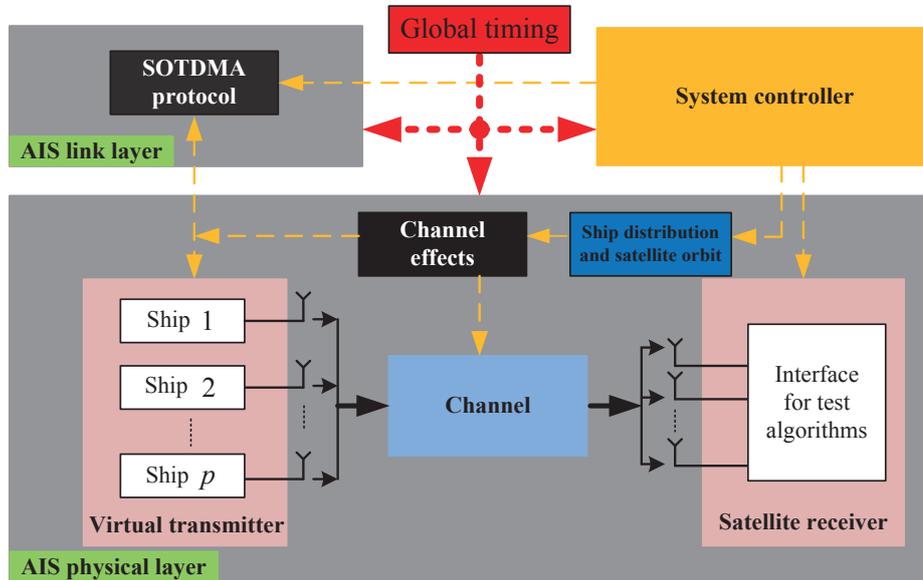


Figure 5.10: The structure of the C++ model.

red dot lines denote the global timing control. Each module in this model is introduced in the following subsections.

5.3.1 Assistant C++ Classes

One of the prominent benefits of using assistant C++ classes is able to separate the code for complex processing from the core SystemC-AMS modules, leaving the coding clean, readable and reusable. A general C++ class *syspara* is defined to help managing the entire system (See Listing 5.1 in Appendix 5.A). It stores the parameters and the status of the system and provides functions to update them (called assistant module “channel effects”). It contains an instance of a C++ class *GroundCell* (See Listing 5.2 in Appendix 5.A) which implements the SOTDMA protocol (called assistant module “SOTDMA protocol”). The instance of this class is declared as “global” and initialized in the “main” function. The look-up table for path loss is put into a file and read out when *syspara* is initialized. Some other assistant classes for message encoding and CRC checking are not listed here.

5.3.2 Virtual Transmitter

The signals in the AIS physical layer flow in the direction from the ship transmitters to the satellite receiver through the channel. This signal flow is modeled in a single TDF cluster of SystemC-AMS with a large number of connections between

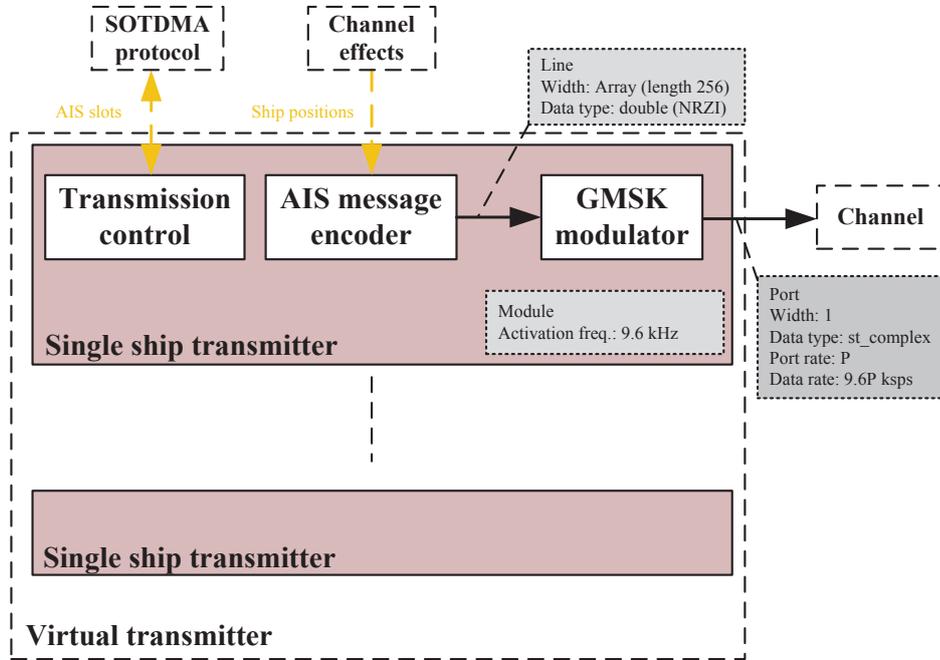


Figure 5.11: The schematic diagram of a single ship transmitter module. The virtual transmitter is a simple wrapper of p single ship transmitters. The output port width of this module is p .

modules. The start-end of this cluster is a virtual transmitter (See Listing 5.3) which wraps p single ship transmitters (See Listing 5.4). Fig. 5.11 shows the structure of a single ship transmitter module. Inside the ship transmitter, a message encoder encodes the ID and the position of the ship (coming from the assistant module “channel effects”) into a data array (data is taken from ± 1.0). The data array is fed into a GMSK modulator. The output of the GMSK modulator is modulated in a self-defined complex-valued data type, *st_complex* (See Listing 5.5 in Appendix 5.A). The rate of the output port is set to P so that the processing thread of this module outputs P samples at each module activation. The activation frequency of the module is set to 9.6 kHz. Then the output data rate is $9.6P$ kilo symbols per second (kps). The transmission control module controls the start and the end of the ship’s transmission by exchanging information with the assistant module “SOTDMA protocol” in the AIS link layer.

5.3.3 Channel

The physical channel is a SystemC-AMS module. It has p input connections from the virtual ship transmitter and m output connections to the satellite receiver. p

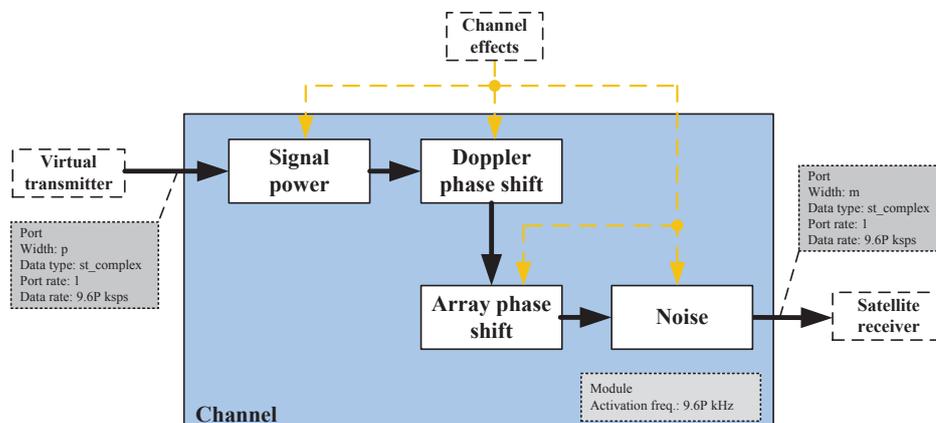


Figure 5.12: The schematic diagram of the channel module.

ought to be set to a number comparable to the maximum number of active signals. It is set to 100 here. The processing inside the module is the implementation of (5.18) (See Listing 5.6 in Appendix 5.A). Fig. 5.12 shows the processing flow inside the channel module, where the p -channel complex-valued input data from the virtual transmitter is multiplied by the signal power, rotated by Doppler phase shifts and array phase shifts, and finally added together to form m output streams separately to the m antennas in the satellite receiver. The parameters needed for the channel effects come from the assistant module “channel effects” in the AIS physical layer. The input and output port rates are 1 because the oversampling is already done in the virtual transmitter. The processing thread of this module inputs and outputs 1 vectorized sample at each module activation. The activation frequency of the module is set to $9.6P$ kHz, and hence, the output data rate is $9.6P$ kbps, being consistent with the data rate of the TDF cluster.

5.3.4 Satellite Receiver

The satellite receiver module is a SystemC module containing a SystemC-AMS part and a SystemC part (See Listing 5.7). Fig. 5.13 shows the structure of the satellite receiver module. The m -channel complex-valued data streams from the channel module are filtered by a baseband filter with a given bandwidth (See Listing 5.8 in Appendix 5.A). The filter is a SystemC-AMS module possessing a configurable Laplace transfer function provided by SystemC-AMS. The filtered data streams interface for testing algorithms. For example, a simple algorithm to detect the number of active signals can be implemented here and the number is available at a single output port of the satellite receiver module. The interface provides a function to output the data streams into a disk file, which is later read out for MATLAB routines. The MU receivers are implemented in MATLAB.

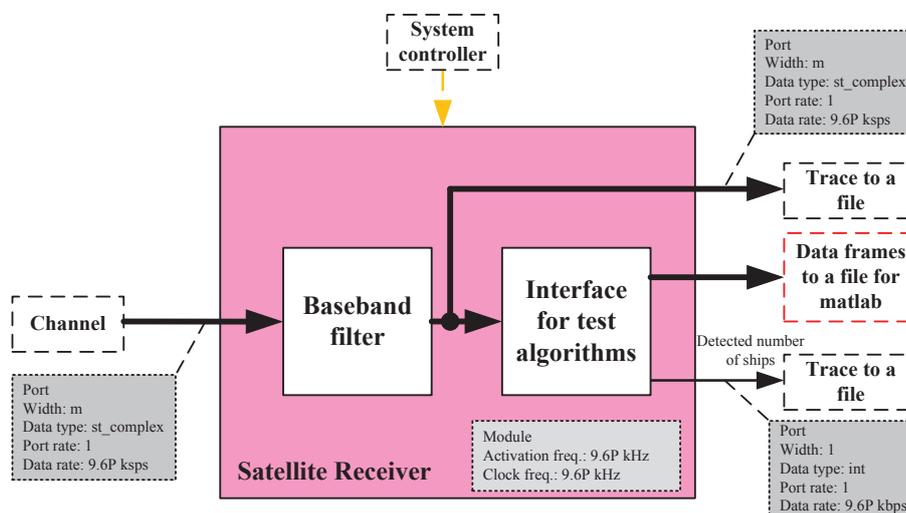


Figure 5.13: The schematic diagram of the satellite receiver module.

At the same time, the same data streams can be output into a tracing file for analysis in other software (VCD file converted to WLF file). The *st_complex* data type provides a standard output format for customized data tracing, allowing analysis of I/Q (complex-valued) signals in SystemC-AMS. The input port of the baseband filter is in the single TDF cluster and has port rate 1. The filter module has activation frequency 9.6P kHz. A conversion port from SystemC-AMS to SystemC is inserted between the baseband filter and the interface for testing algorithms. The interface is in SystemC. The clock frequency of the SystemC modules is 9.6P kHz, which is the same as that of the SystemC-AMS module but in the discrete time domain.

5.3.5 Channel Effects and SOTDMA Protocol

The assistant modules provide functions to update the parameters related to the channel effects and the SOTDMA protocol (See Listing 5.2 in Appendix 5.A). The module “channel effects” calculates the parameters of the channel effects as functions of the positions of ships. The parameters are stored in the arrays of the assistant class *syspara*.

The module “SOTDMA protocol” is more complex. The implementation is related to the ship distribution and the shape of the SOTDMA cells. The margin of SOTDMA cells are approximately circles limited by the horizon. For simplicity, the cells can be assumed to be squares in a regular grid of the flat FoV (See Subsection 5.2.3). The lengths of sides are equal to 40 nautical miles. The positions of the grid are stored in circular *deque* arrays. The grid and cells are moving

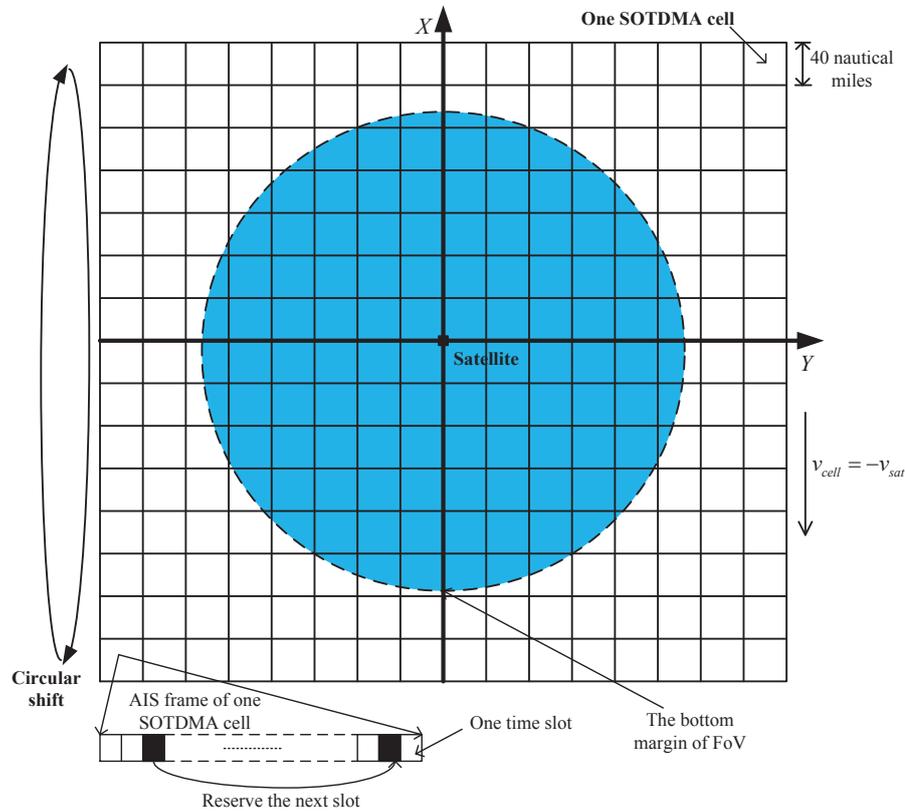


Figure 5.14: The implementation of the SOTDMA protocol.

downward while the satellite flies upward in Fig. 5.14. When one line of cells moves away from the bottom margin of the FoV for a given distance, the storage of the line of cells in the arrays is removed and a new line of cells is pushed into the top line of cells. The status of the AIS frame and the slot assignment in each cell is stored in *deque* arrays. The scheduling of messages in the cell is updated regularly and the start time of the transmission of messages for each ship in the cell is calculated. The actual arrival time of messages is the sum of the start time and the channel propagation time delay. Equivalently, the start time of the transmission is set to the actual arrival time, directly assuming no time delay in the channel module.

5.3.6 Ship Distribution and Satellite Orbit

In the former subsection, the uniform ship distribution and square SOTDMA cells are assumed. That setup is a simulation case for a constant number of ships

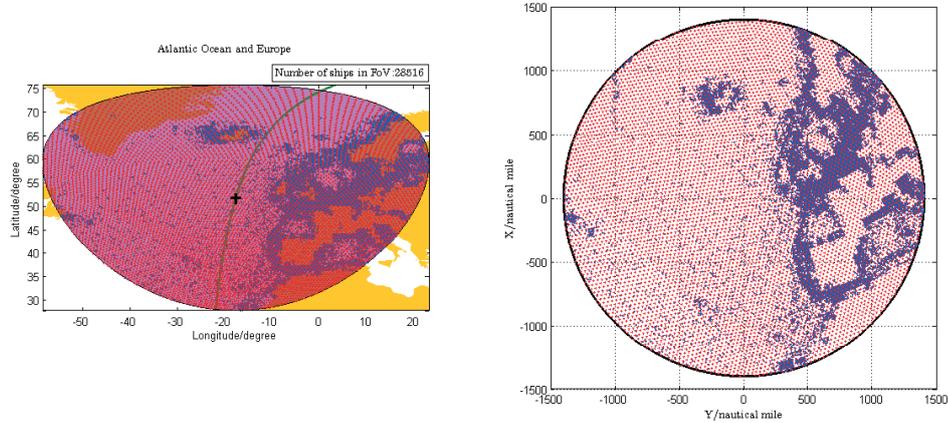


Figure 5.15: The area coverage of FoV and ships during the LEO satellite flying over the Atlantic Ocean and the land of Europe. The shape of SOTDMA cells is geodesic grids. The blue dots denote the ships within FoV. The red plus sign denote the center points of SOTDMA cells. The black sign “+” denotes the sub-satellite point. The green line denotes the sub-satellite track.

in FoV. Another simulation case is more attractive due to using the practical global ship distribution from open databases combined with satellite orbit design from launched satellites [94]. The SOTDMA cells are designed from geodesic grids [109–111]. Fig. 5.15 and Fig. 5.16 show two examples of this simulation case. In these figures, the conversion between the FoV and the flat FoV are shown and this conversion help integrating the practical data into the system.

5.3.7 System Controller

The system controller is the key component to make the system dynamic. The system controller is modeled as a SystemC module with different clock drivers (See Fig. 5.17). The system controller initializes the system by reading the system configuration from the assistant class *syspara*. The system controller has two internal processes regularly refreshing the system status (See Listing 5.9 in Appendix 5.A). One process updates the positions of ships (φ_i, r_i) and the related parameters of the channel effects during the satellite is flying over the ships. The other process checks the start time T_u of the ship transmission and triggers the real-time transmission of messages when it reaches the time point. The first process invokes the functions in the assistant module “SOTDMA protocol” and the second process invokes the functions in the assistant module “channel effects”. The two processes both change the system status stored in the assistant class *syspara*.

The positions of ships (φ_i, r_i) are updated as follows. In the initialization

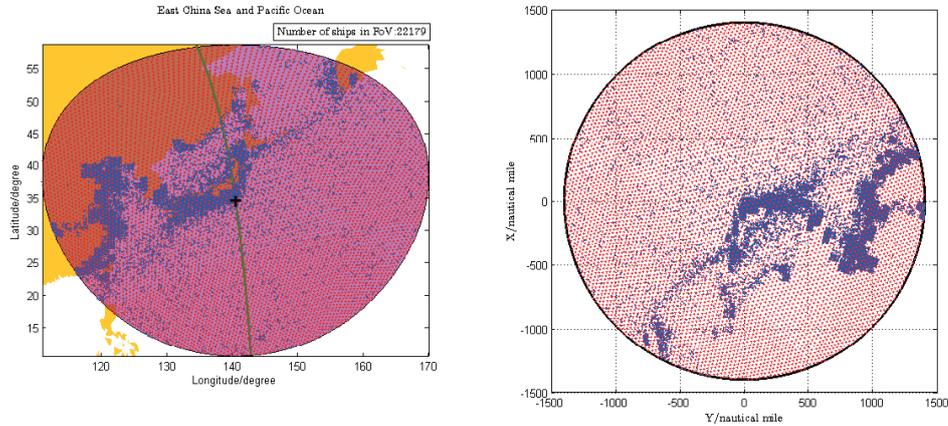


Figure 5.16: The area coverage of FoV and ships during the LEO satellite flying over the East China Sea and the Pacific Ocean. The shape of SOTDMA cells is geodesic grids. The blue dots denote the ships within FoV. The red plus sign denote the center points of SOTDMA cells. The black sign “+” denotes the sub-satellite point. The green line denotes the sub-satellite track.

of the system, ships are randomly distributed in the FoV either uniformly or according to a practical distribution. For simplicity, the uniform ship distribution is used here. All the ships in the FoV moves backward opposite to the velocity vector of the satellite. Every ship has a unique ID. The number of the ships in the FoV is kept constant. The number of ship IDs grows when new ships being added. When the ground range r_i between one ship and the satellite exceeds the radius of FoV, this ship is removed and another ship is randomly generated at the top margin of FoV.

The start time of message transmission follows a poisson process in a given report interval, and the intensity of this poisson process changes with the number of ships within FoV. The start time are pseudo random time points from a look-up array generated offline because the report intervals are known. The calculation of the start time is done by the assistant module “SOTDMA protocol” but the trigger of this transmission is managed by the system controller, which is a cross-layer control through the communication between the ship transmission control module and the system controller across the link layer and physical layer (See Fig. 5.18). The system controller regularly checks whether the system time T_g exceeds the start time of a ship’s transmission T_u , i.e., $T_g > T_u$. When this condition fulfills, the system controller reserves another time slot for this ship associated with a new start time T'_u of this ship by accessing the module “SOTDMA protocol”, and sets the status of this ship to “transmission”. After that, the transmission control module inside this ship gets this status “transmission” and immediately triggers the message transmission. The transmission control module stops the

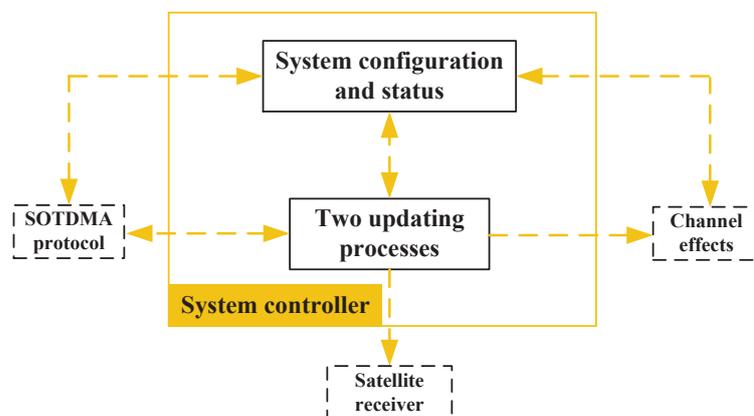


Figure 5.17: The schematic diagram of the system controller module.

transmission when $T_g > T_u + N_p T$, assigns $T_u = T'_u$, and sets the status of this ship to “standby”, where T is the symbol period and N_p is the message length.

The system controller determines the number of active signals and fills the free seats in the virtual ship transmitters with the active signals. If the number of active signals exceeds the number of seats, then the number is truncated and the remaining signals are ignored since the number of seats is large enough.

5.3.8 Additional Functions

Some interesting parts of the modules of this C++ model can be replaced and extended for different simulation scenarios. Most of the modifications are not discussed in this chapter. However, they are listed as optional choices

1. The latitude of the satellite orbit can be changed. Most of the changes to channel effects are automatic except new input files for the path loss.
2. The satellite antennas can be replaced with directional antennas (See [5]). The antennas can have different types of wave polarization, where the path loss is changed accordingly. This effect can be even made dynamic if the satellite is spinning (See Chapter 4).
3. Different transmission intervals for ships can be used (See Chapter 4).
4. The message length can be different for different ships.
5. Sudden interrupts for the transmission from part of the FoV can be modeled, such as the interrupts caused by scintillation in ionosphere.
6. The parameters of the baseband filter in the satellite receiver can be changed (See Chapter 4).

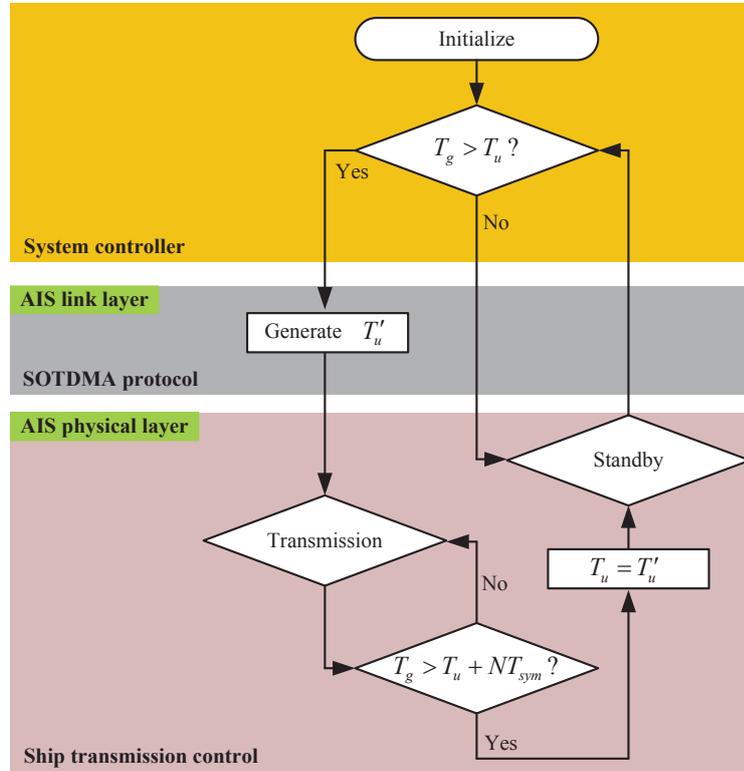


Figure 5.18: The flow chart of the ship transmission control.

7. Imperfection besides noise in the satellite receiver can be added, such as the imbalance of IQ paths and the incorrect DC offsets.

5.4 Simulation Results

In this section, first, the model is verified by comparing test outputs with the ideal output shown in Section 5.2. Second, the model is validated by simulating it with the receivers designed in MATLAB.

This model is configured as listed in Table 5.1. The modeling tools are the libraries, SystemC-AMS 1.0BETA2 and SystemC 2.2.0. The code is compiled and built in Visual Studio 2008 on Windows 7. The hardware platform is a PC with Intel Duo CPU 3.0 GHz and 4 GB ram.

An important evaluation factor of the model is the time cost of one run of the simulation. The time cost must be acceptable. The measure of this factor is the ratio of the time cost of one run T_{simu} to the modeled system time period T_{sys} ,

Table 5.1: *The configuration of the C++ model.*

| | |
|---|---|
| Carrier frequency $f_{carrier}$ | 162.025MHz |
| Channel bandwidth | 25 kHz |
| Receiver baseband filter BW | 15 kHz or 25 kHz |
| Modulation | 9.6 kbps GMSK |
| BT | 0.3 |
| Oversampling ratio P | 8 |
| Channel module activation freq. | $9.6P$ kHz |
| Satellite altitude h_{sat} | 600 km |
| Satellite speed v_{sat} | 7561.65 m/s |
| Orbit period T_{sat} | 5792.52 s |
| Number of ships in FoV | 10 to 10,000 |
| Number of cells | 5476 UTC synchronized SOTDMA cells |
| Radius of FoV r_{FoV} | 1437.74 nautical miles |
| Maximum ship visible time | 704 s (One satellite pass) |
| Ship report interval | 0.25 s, 1 s, 6 s, 60 s |
| Ship position updating interval | 10 s |
| AIS message length N_p | 256 bits |
| Ship transmitter power | 12.5 W |
| Ship antenna type | Perpendicular half-wave dipole |
| Received power on satellite | Look-up table |
| Number of satellite antennas m | 4, 20 |
| Satellite antenna type | Omnidirectional circularly polarized half-wave dipole |
| Polarization loss | Fixed -3 dB |
| Satellite antenna spacing $\Delta\lambda$ | Half a wavelength 0.5λ |
| Satellite antenna gain G_a | 0 dB |
| Virtual gain G_{AGC} | 97 dB |
| Uplink noise power | -121.30 dBm |

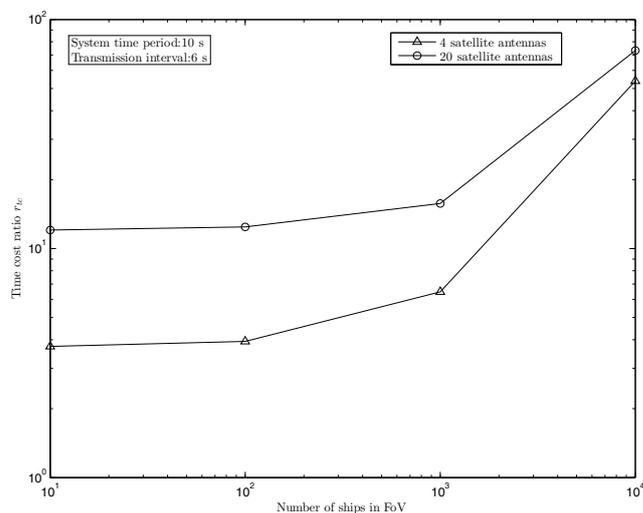


Figure 5.19: The time cost ratio as a function of the number of ships in FoV for 4 and 20 satellite antennas.

which is given as

$$r_{tc} = \frac{T_{simu}}{T_{sys}}. \quad (5.23)$$

In order to simulate a clean SystemC-AMS model to collect the time cost, all output streams to files are shut down and no test algorithm in the satellite receiver is used. Fig. 5.19 shows the results of a real-time simulation for a system time period of 10 seconds for 10 to 10,000 ships in FoV and $m = \{4, 20\}$ satellite antennas. The ratio increases nonlinearly as a function of the number of ships in FoV, which was expected to have a linear growth. The explanation for this nonlinearity may be the undetermined loss of efficiency in the growth of stack. Nevertheless, in Fig. 5.19, the time cost of the worst case with 10,000 ships in FoV is acceptable.

5.4.1 Verification of the Model

Next, the code of the C++ model for implementing the mechanism introduced in Section 5.2 is tested for verifying its correctness. The verification process is made under the case when the ships are uniformly distributed in FoV. The output parameters from the model are recorded and compared with the figures in Section 5.2. To check the parameters of the channel effects, 10,000 ships are generated. The report interval is set to 1 second for all the ships and the simulated system time period is set to 2 seconds. Fig. 5.20 to Fig. 5.24 show the configured

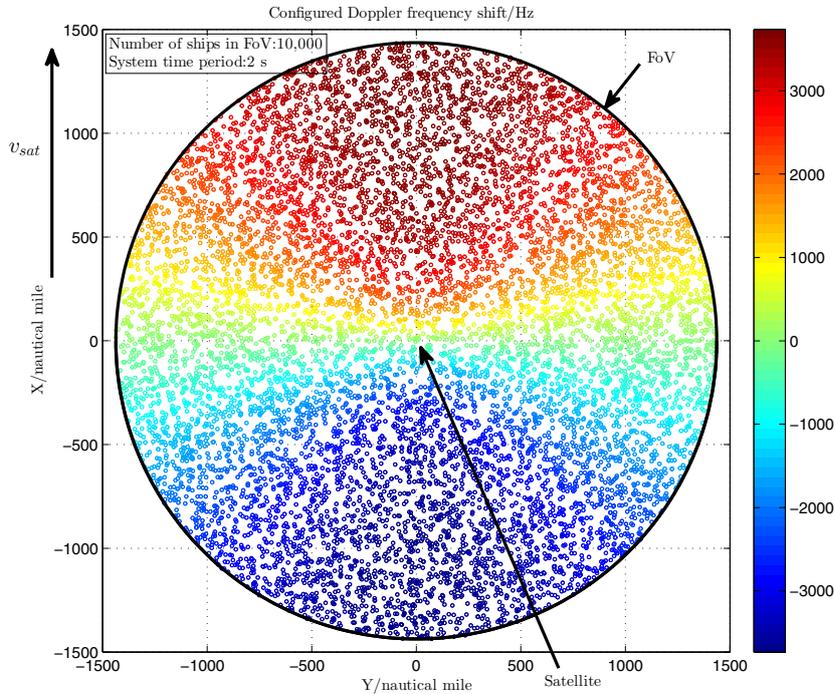


Figure 5.20: The recorded configured Doppler frequency shifts of signals transmitted by the ships in FoV.

Doppler frequency shifts, configured relative propagation time delays, configured received signal power and configured DOAs corresponding to Fig. 5.4, Fig. 5.5, Fig. 5.6 and Fig. 5.8, respectively. It is clearly seen that the results are identical and the modeling is correct.

To check the SOTDMA protocol, 10 ships are generated in FoV. The report interval is set to 0.25 seconds and the simulated system time period to 0.5 seconds. Fig. 5.25 illustrates the activity in the wireless channel seen by the satellite. In Fig. 5.25, the vertical dash lines are the divider line of the time slots. The colored rectangles are messages. The messages dyed the same color are from the same ship. The modeling of the report interval, the propagation time delays and the types of message collisions are reflected in the figure. Type 1 collision is caused by the messages transmitted from the same time slot. Type 2 collision is caused by the messages from adjacent time slots. The results show that the SOTDMA protocol is successfully modeled.

Most of the parameters of the model are dynamically changing with the posi-

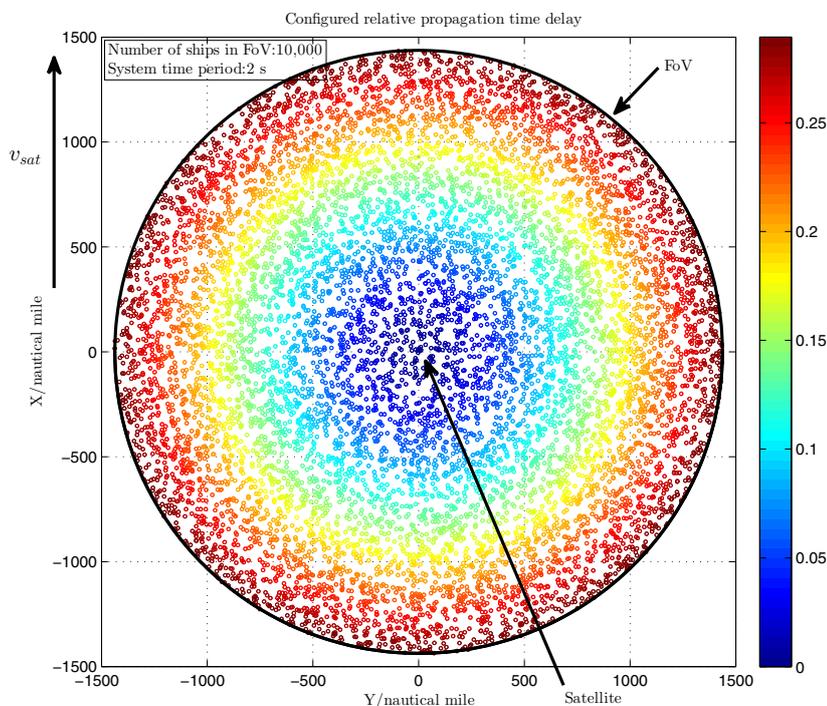


Figure 5.21: The recorded configured relative propagation time delays of signals transmitted by the ships in FoV in terms of a ratio of the length of the time slot.

tions of ships. The movement of ships must be checked. Again, assume that the number of ships is constant in FoV. Fig. 5.26 shows the movement of ships in a long time simulation that lasts for 704 seconds. The 5th ship (marked red) moves out of FoV and a new generated ship, the 11th ship (marked red), moves in FoV from the top margin. The other ships move in and out of the FoV in a similar way. It is clear that the implementation of the movement of ships is correct.

The model allows changes to the nonuniform distribution of ships in FoV. Fig. 5.27 shows a recorded nonuniform distribution of 10,000 ships in a simulated system time period for 60 seconds. This distribution is a simple case where the satellite is flying above the coastline at its right-hand side. The majority (95%) of the ships are located close to the coastline, which have 6 seconds report intervals. The remaining (5%) of the ships are far away from the coastline, which have 60 seconds report intervals.

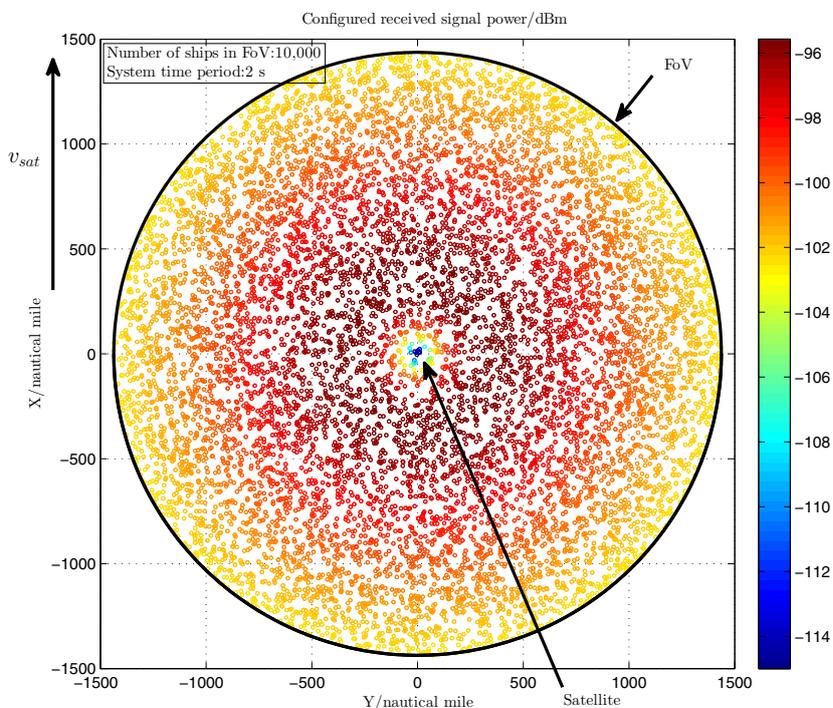


Figure 5.22: The recorded configured received power of signals transmitted by the ships in FoV.

5.4.2 Validation of the Model

Now, after the model is verified, it can be concluded that the model is consistent with the specifications in Section 2. In the following subsection, the model is used to generate test data for the receivers designed and implemented in MATLAB (See Fig. 5.28). The receivers have been tested in MATLAB with a transmitter and a simple noisy channel, but this test lacks of full information of the real scenario. The transmitter and the channel in MATLAB generate random block data in a single analysis window for testing the receivers. When the test in MATLAB is done, the block data from the output of the model are used to test the receivers in a more real way. The block data are collected from consecutive sliding windows on the test data. The adjacent windows can partially overlap each other.

Table 5.2 shows the comparison between the configured parameters and the corresponding received and estimated parameters. The test data is taken from the simulation shown in Fig. 5.26. It is evident that the test data from the model are

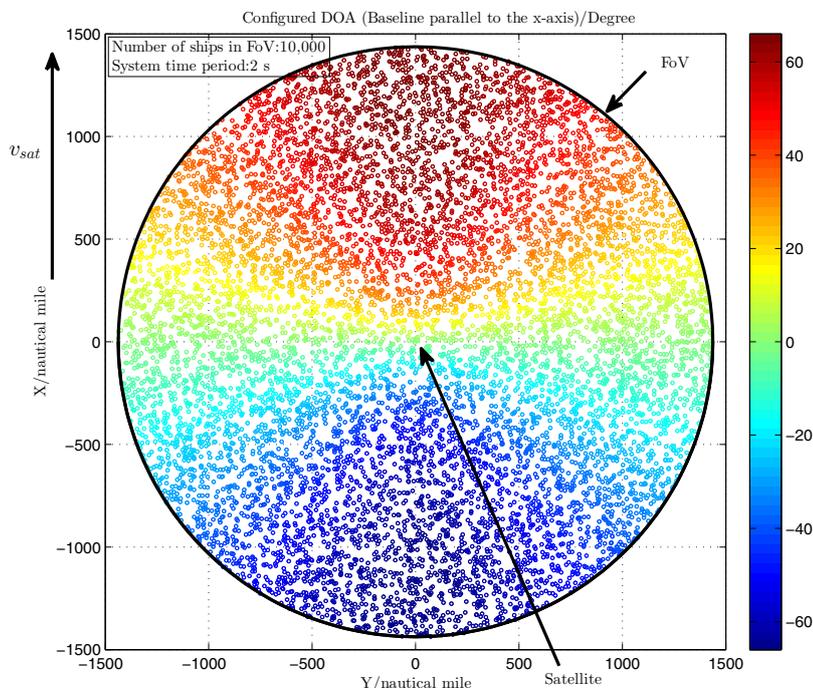


Figure 5.23: The recorded configured DOAs (baseline parallel to the x -axis) of signals transmitted by the ships in FoV.

successfully processed by the receivers in MATLAB, which validates the model.

In addition, an example of the separation of two collided messages is shown in Fig. 5.29, which exhibits the successful separation of the circled Type 1 collision shown in Fig. 5.25. Note that the bottom subfigure also shows the modeled power-up part, power-down part and empty buffers at the tail of the messages. This verifies the model once again.

Finally, different receivers are simulated in MATLAB using the block data from the C++ model. The performance of the receivers are compared. The simulation uses a system time period of 704 seconds and 10,000 ships in FoV. Fig. 5.30 shows the comparison between the ratios r_{meg} of the number of successfully detected messages to the number of total sent messages for different receivers. The performance grows when the ratio goes close to 1. In this comparison, the advanced multi-user receivers are MU (ACMA in Chapter 4.5), MU+SI (GSVD-SI+ACMA in Chapter 4.5), and MU+IFST (GSVD-T+ACMA in Chapter 4.5) [14–16, 20, 23, 24]. Here, SU is a single user receiver, as in Fig. 5.30,

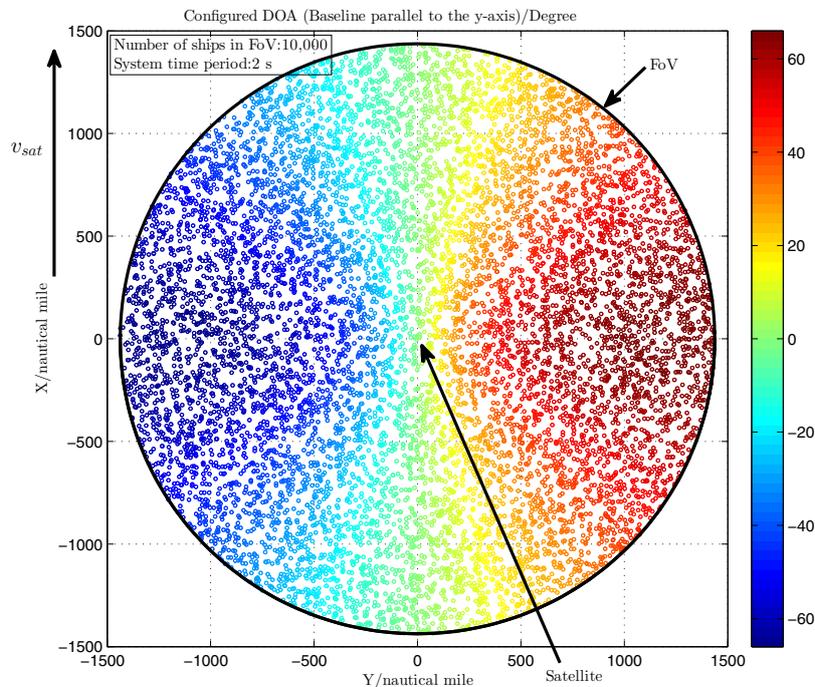


Figure 5.24: The recorded configured DOAs (baseline parallel to the y-axis) of signals transmitted by the ships in FoV.

corresponding to MU with 1 antenna. It is seen that the advanced multi-user receivers improve the detection of messages in Satellite AIS. This validates the function of the C++ model.

5.5 Conclusions

In this chapter, a SystemC-AMS model of a dynamic large-scale network of Satellite AIS is already presented, verified and validated. It is demonstrated how SystemC-AMS is used as an efficient tool to model such a complex system. The AIS link layer and the AIS physical layer are analyzed and abstracted for modeling. With the help of the TDF modeling style in SystemC-AMS and its seamless integration to SystemC domain, the activity of the both layers and a cross-layer system controller are precisely timed. The modular modeling style provides sufficient flexibility in modeling real objects, such as transmitters, physical channels and receivers. Many important factors of Satellite AIS are modeled in the TDF

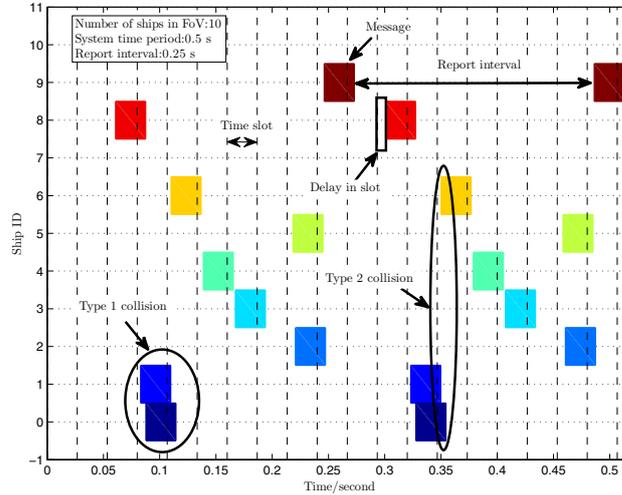


Figure 5.25: The recorded report time of messages. The report interval, the propagation time delays and the types of message collisions are shown.

domain, such as the Doppler frequency shift, the path loss, DOA, the propagation time delay, the real-time transmission, and the collision of messages. The open structure of this model allows further modification and extension for different simulation scenarios.

The modeling method introduced in this chapter can be applied to the modeling of other communication systems. These virtual models will help saving the time cost during the hardware implementation of future communication systems.

5.A Lists of Codes

Listing 5.1: The *syspara* class.

```

1 class syspara
2 {
3 public:
4     // System size
5     int m_num_user, m_num_trans, m_num_rec;
6     double m_oversampling, m_antenspacing;
7     // Global system time
8     double m_global_time, m_global_time_l;
9     // Ship data 'for' channel effects
10    double *p_Cafreq_u, *p_lamda_u;
11    double *p_Pathloss_u, *p_ar_angle_u;

```

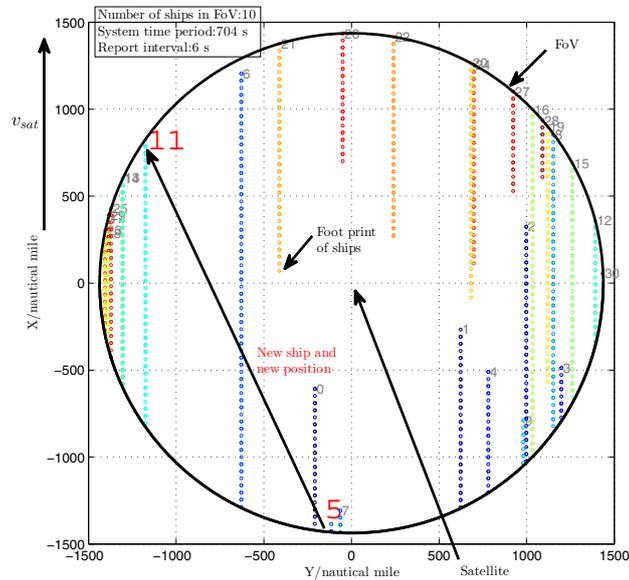


Figure 5.26: The movement of the ships in FoV in a long time simulation.

```

12     double *p_transint_u, *p_start_time_u;
13     double *p_delayed_start_time_u;
14     ...
15     // Satellite's orbit
16     double m_high_sat, m_velo_sat;
17     double m_T_sat, m_earth_r;
18     ...
19     // Received power
20     double *p_Pr;
21
22     // SOTDMA protocol manager
23     GroundCell *p_cellmanag;
24     ...
25     syspara();
26     ~syspara();
27     // Calculate the satellite orbit
28     void initpara();
29     // Calculate channel effects
30     double doppler_sat(double r_u, double phi_u);
31     void initPathloss();
32     // Distribute ships in FoV
33     void genuserdata();
34     // Uniform distribution
35     void gen_randpos_u(int o);
36     void gen_pos_from_margin_u(int o);

```

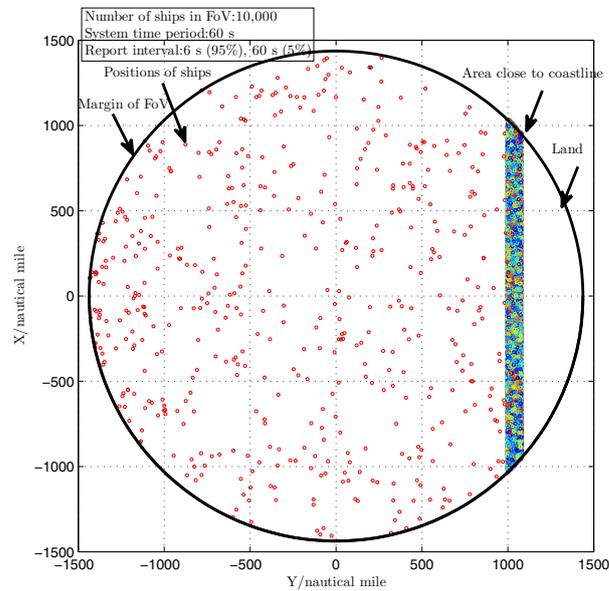


Figure 5.27: A recorded nonuniform ship distribution in FoV.

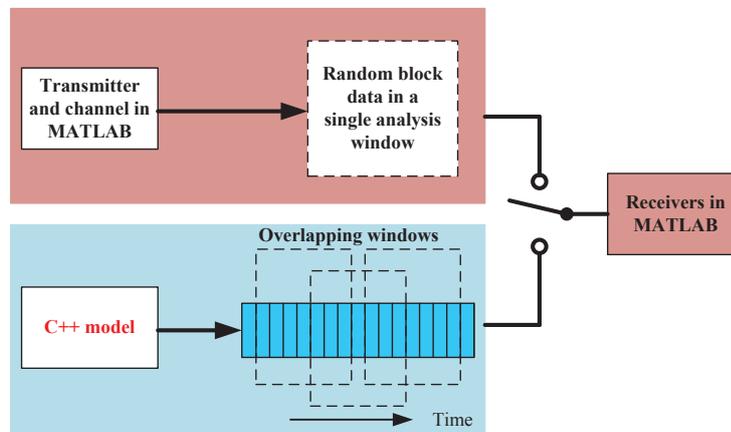


Figure 5.28: The validation of the model with receivers implemented in MATLAB.

```

37 // Nonuniform distribution
38 void gen_pos_in_scenario_u(double y_shore, double shore_width, double
    shore_ratio);
39 void gen_pos_in_scenario_from_margin_u(int o, double y_shore, double
    shore_width, double shore_ratio);

```

Table 5.2: Comparison between the configured parameters and the corresponding received and estimated parameters.

| Configured Parameters | | | | | | |
|-----------------------------------|-------------|-------------|-------------------|-------------------------|--------------------------|-------------|
| ID i | x_i (nmi) | y_i (nmi) | Δf_i (Hz) | θ_{iax} (degree) | θ_{ia_y} (degree) | P_i (dBm) |
| 0 | -608 | -209 | -3316.86 | -54.2538 | -16.1654 | -97.100229 |
| 1 | -267 | 623 | -1391.50 | -19.9068 | 52.7757 | -97.364434 |
| 2 | 324 | 998 | 1144.41 | 16.2620 | 59.5936 | -100.106270 |
| 3 | -488 | 1198 | -1405.18 | -20.1109 | 57.8039 | -101.685616 |
| 4 | -510 | 781 | -2007.83 | -29.4262 | 48.9064 | -99.287914 |
| 5 | -1386 | -115 | -3722.24 | -65.6172 | -4.32853 | -102.262895 |
| 6 | 1205 | -629 | 3311.13 | 54.1166 | -25.0078 | -102.081321 |
| 7 | -1308 | -63 | -3729.24 | -65.8561 | -2.49725 | -101.782703 |
| 8 | 852 | 1152 | 2220.27 | 32.9076 | 47.2941 | -102.510403 |
| 9 | -790 | 980 | -2340.18 | -34.9336 | 45.3231 | -101.476760 |
| Received and Estimated Parameters | | | | | | |
| 0 | -608 | -209 | -3317.03 | -54.2286 | -16.1598 | -97.604432 |
| 1 | -267 | 623 | -1391.44 | -19.9234 | 52.7654 | -97.885004 |
| 2 | 324 | 998 | 1143.67 | 16.2573 | 59.5862 | -100.658912 |
| 3 | -488 | 1198 | -1405.11 | -20.1014 | 57.8183 | -102.253472 |
| 4 | -510 | 781 | -2007.94 | -29.4308 | 48.8846 | -99.840751 |
| 5 | -1386 | -115 | -3722.43 | -65.6793 | -4.31531 | -102.791682 |
| 6 | 1205 | -629 | 3311.08 | 54.0675 | -25.0017 | -102.640752 |
| 7 | -1308 | -63 | -3729.47 | -65.7914 | -2.49338 | -102.296416 |
| 8 | 852 | 1152 | 2220.75 | 32.8995 | 47.2431 | -103.102978 |
| 9 | -790 | 980 | -2340.26 | -34.9444 | 45.3190 | -102.056188 |

```

40 // Move ships
41 void update_pos_u(double  $\Delta_T$ , int o);
42 void update_spatialpara_u(double  $\Delta_T$ , int o);
43 // Ship transmission control
44 void checktrans_u(int o);
45 void shutdowntrans_u(int o);
46 };
47 // Global declaration
48 extern syspara *g_syspara;

```

Listing 5.2: The GroundCell class.

```

1 class GroundCell
2 {
3 public:
4 // SOTDMA cells
5 double m_cell_width;
6 int m_num_grid_radius;
7 deque<double> *hor, *ver;
8 int **p_ShipatCell;
9
10 // AIS frame and AIS slots
11 int m_num_slots;
12 double m_slottimelen;
13 double m_frametimelen;

```

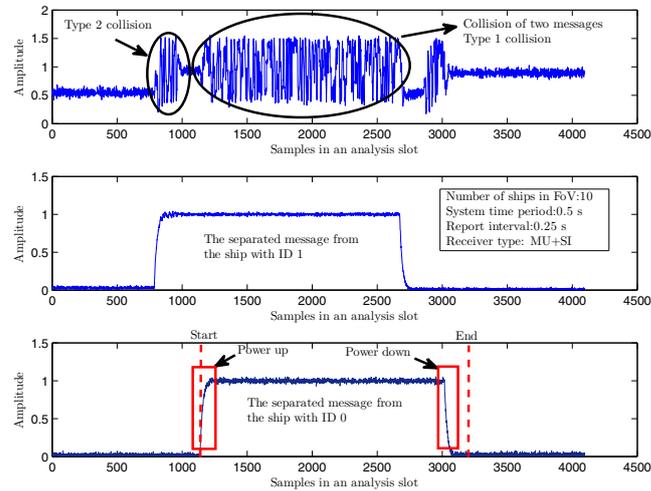


Figure 5.29: The collided messages and the separated messages for the circled Type 1 collision shown in Fig. 5.25.

```

14  int *p_VisibleShipslot;
15  deque<deque<double> *> *p_cell_frame_start_time;
16  // Slot assignment in cells
17  deque<deque<deque<cellunit> *> *> *cells;
18
19  explicit GroundCell(double localhorizon_in);
20  ~GroundCell();
21  // Create cells
22  void GenGrid();
23  // Move cells
24  void MoveGrid(double distant);
25  // Keep number of cells fixed
26  void Cellcirshift();
27
28  // AIS frames
29  void InitialCellFrameStartTime();
30  void UpdateCellFrameStartTime(double g_time_in);
31  // Add ships into AIS frames of cells
32  void InitialShipinCells(int *p_ID_u, double *p_r_u, double *p_phi_u, double *
    p_transint_u, double *p_start_time_u, double g_time_in);
33  void AddSingleShipinCell(int o_in, int ID, double r, double phi, double
    transint_u, double *p_start_time_u, double g_time_in);
34  void AddShip(int o_in, int ID_in, int reserveslot_in, double x, double y,
    double *p_start_time_u, double g_time_in);
35  // SOTDMA slot reservation
36  void ReserveNextSlot(int o_in, int ID_in, double *p_transint_u, double *
    p_start_time_u, double g_time_in);

```

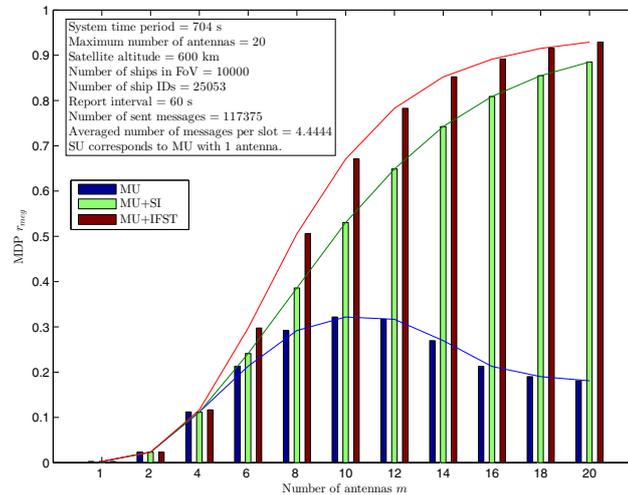


Figure 5.30: The performance of different receivers tested with the C++ model. MU, MU+SI and MU+IFST are ACMA, GSVD-SI+ACMA, and GSVD-T+ACMA in Chapter 4.5, respectively

```

37     void ReserveSlot(int o_in, int ID_in, int reserveslot_in, double *
38         p_start_time_u, double g_time_in, int v_index, int h_index);
39     // Dynamic scintillation
40     void UpdateScintinCells();
41     void InitialPathlosswithScint(double *p_Pathloss_u);
42     void PathlosswithScint(int o_in, double *p_Pathloss_u);
43 };

```

Listing 5.3: The multi_users_trans_b_ams class.

```

1  template<int NUMUSERS>
2  SC_MODULE(multi_users_trans_b_ams)
3  { // Actually a SCA_TDF_MODULE. Just a wrapper.
4      // Output port. It is an array of ports.
5      sca_tdf::sca_out<st_complex> o_trans[NUMUSERS];
6      // Single ship transmitter
7      user_b_ams **p_user_trans;
8      // Constructor
9      SC_CTOR(multi_users_trans_b_ams)
10     {
11         int i; char stbuf[20];
12         p_user_trans = new user_b_ams*[NUMUSERS];
13         for (i = 0; i < NUMUSERS; i++)
14             {

```

```

15         sprintf(stbuf, "user_trans(%d)", i);
16         p_user_trans[i] = new user_b_ams(stbuf, i);
17         p_user_trans[i]->o_GMSK_trans(o_trans[i]);
18     }
19 }
20 ~multi_users_trans_b_ams();
21 };

```

Listing 5.4: *The user_b_ams class.*

```

1  SCA_TDF_MODULE(user_b_ams)
2  {
3      // Output port 'for' AIS signals
4      sca_tdf::sca_out<st_complex> o_GMSK_trans;
5      int o;// Ship ID
6      // State machine
7      bool isPacketgened; int cnt;
8      // Actual message length
9      int PLen;
10     // GMSK modulation parameters
11     int L; int P;
12     double h; double *packetphase;
13     // Scaling 'for' RF power-up and power-down
14     double *RFPower;
15     // Module time step and port rate
16     void set_attributes()
17     {
18         double T = 1.0/(1.0 * int(USER_RATE));
19         set_timestep(T,sc_core::SC_SEC);
20         o_GMSK_trans.set_rate(P);
21     }
22     // Generate scaling 'for' RF power-up and power-down
23     void genRFPower();
24     // GMSK modulator
25     double* GMSK(double *packet, int PLen);
26     // Activation process
27     void processing();
28     { // Ship transmission control
29         if (g_syspara->p_status_u[o] == 0) // "Standby"
30         {
31             o_GMSK_trans.write(st_complex(0.0,0.0));
32             cnt = 0;
33         }
34         else // "Transmission"
35         {
36             if (cnt < g_syspara->p_packlen_u[o])
37             {
38                 if (cnt == 0 && isPacketgened == false)
39                 {
40                     // Generate AIS signals
41                 }
42                 if (cnt < 10) // Power up
43                 {
44                     for (i=0;i<P;i++)

```

```

45         {
46             o_GMSK_trans.write( st_complex(cos(packetphase[cnt*P+i]),
47                                     sin(packetphase[cnt*P+i])) * (1.0-RFPower[cnt*P+i]),
48                                     i);
49         }
50     }
51     else if (cnt < PLen){// Training,Start FFlag,Data,CRC...}
52     else if (cnt < PLen + 10){// Power down}
53     else{// Empty buffer}
54         cnt++;
55     }
56     else // Shut down
57     {
58         while (g_syspara->p_status_u[o] == 1)
59         {
60             g_syspara->shutdowntrans_u(o);
61         }
62     }
63 // Constructor
64 explicit user_b_ams(::sc_core::sc_module_name , int o_in = 0)
65     :o(o_in)
66     {
67         PLen = int(PACKAGE_LENGTH);
68         L= int(GMSK_L);
69         P = int(OVERSAMPLINGRATIO);
70         h = double(GMSK_H);
71         isPacketgened = false;
72         RFPower = new double[10*P];
73         genRFPower();
74     }
75     ~user_b_ams();
76 };

```

Listing 5.5: The st_complex class.

```

1 class st_complex
2 {
3 public:
4     double real, imag;
5     st_complex();
6     st_complex(double real_in, double imag_in);
7     st_complex(const st_complex &r);
8     inline st_complex &operator=(const st_complex &r);
9     inline bool operator==(const st_complex &r) const;
10    inline friend st_complex operator+(const st_complex &r1, const st_complex &r2)
11        ;
12    inline friend st_complex operator-(const st_complex &r1, const st_complex &r2)
13        ;
14    inline friend st_complex operator*(const st_complex &r1, const st_complex &r2)
15        ;
16    inline friend st_complex operator*(const st_complex &r, const double s);
17    inline friend st_complex operator/(const st_complex &r, const double s);

```

```

15 inline friend st_complex operator/(const st_complex &r1, const st_complex &r2)
16 ;
17 inline double norm_s() const;
18 inline double norm() const;
19 inline st_complex conj() const;
20 // Output stream
21 inline friend ostream& operator << (ostream& os, st_complex const &r)
22 {
23     os << "(" << r.real << ", "<< r.imag << ")";
24     return os;
25 }
26 // Trace 'function' 'for' SystemC
27 inline friend void sc_trace(sc_trace_file *tf, const st_complex &r, const std
28 ::string & NAME)
29 {
30     sc_trace(tf,r.real, NAME + ".real");
31     sc_trace(tf,r.imag, NAME + ".imag");
32 }
33 // This 'function' is not available in SystemC-AMS 1.0BETA2
34 inline friend void sca_trace(sca_util::sca_trace_file *tf, const st_complex &r
35 , const std::string & NAME)
36 {
37     sca_util::sca_trace(tf, r.real, NAME + ".real");
38     sca_util::sca_trace(tf, r.imag, NAME + ".imag");
39 }

```

Listing 5.6: *The channel_b_ams class.*

```

1 template<int NUMUSERS, int NUMTRECS>
2 SCA_TDF_MODULE(channel_b_ams)
3 {
4     // Input and output ports.
5     sca_tdf::sca_in<st_complex> i_sig[NUMUSERS];
6     sca_tdf::sca_out<st_complex> o_sig[NUMTRECS];
7     // Module time step
8     void set_attributes()
9     {
10         double T = 1.0/(int(OVERSAMPLINGRATIO) * 1.0 * int(USER_RATE));
11         int i;
12         for (i = 0; i < NUMUSERS; i++)
13         {
14             i_sig[i].set_timestep(T, sc_core::SC_SEC);
15         }
16     }
17     // Activation process
18     void processing()
19     {
20         int i,j;
21         st_complex signalout[NUMTRECS];
22         st_complex array_phase;
23         st_complex doppler_phase;
24         double phi;
25         for (i = 0; i < NUMUSERS; i++) // Process data from every ship

```

```

26     {
27         if (g_syspara->p_status_u[i] == 1)
28         { // Doppler phase shift
29             phi = 2 * M_PI * (g_syspara->p_Cafreq_u[i] - g_syspara->
30                 m_centralfreq) * sc_time_stamp().to_seconds();
31             doppler_phase = st_complex(cos(phi), sin(phi));
32             for (j = 0; j < NUMTRECS; j++)
33             { // Add the output to antennas
34                 if (j < NUMTRECS/2) // Arrival angle (Baseline along Y axis)
35                 { // Array phase shift
36                     phi = 2 * M_PI * g_syspara->m_antenspacing * j * sin(
37                         g_syspara->p_ar_angle_u[i]) * g_syspara->m_lamda /
38                         g_syspara->p_lamda_u[i];
39                 }
40                 else // Arrival angle (Baseline along X axis)
41                 {
42                     phi = 2 * M_PI * g_syspara->m_antenspacing * (j - NUMTRECS
43                         /2 + 10) * sin(g_syspara->p_ar_angle_u2[i]) *
44                         g_syspara->m_lamda / g_syspara->p_lamda_u[i];
45                     phi = phi - 2 * M_PI * g_syspara->m_antenspacing * 10 *
46                         sin(g_syspara->p_ar_angle_u[i]) * g_syspara->m_lamda
47                         / g_syspara->p_lamda_u[i];
48                 }
49                 array_phase = st_complex(cos(phi), sin(phi));
50                 signalout[j] = signalout[j] + i_sig[i].read() * g_syspara->
51                     p_Pathloss_u[i] * array_phase * doppler_phase;
52             }
53         }
54     }
55     double noise_mag = 0.707 * sqrt(pow(10.0, (g_syspara->m_noise_p + double(
56         AGCGAIN))/10.0));
57     st_complex Antenna_noise[NUMTRECS];
58     for (i = 0; i < NUMTRECS; i++)
59     { // Add noise
60         Antenna_noise[i] = st_complex(gaussian(0.0, noise_mag), gaussian(0.0,
61             noise_mag));
62         o_sig[i].write(signalout[i] + Antenna_noise[i]);
63     }
64 }
65 SCA_CTOR(channel_b_ams){}
66 };

```

Listing 5.7: The multi_antennas_fe.b class.

```

1  template<int NUMTRECS>
2  SC_MODULE(multi_antennas_fe_b)
3  {
4      // Input port in the TDF cluster
5      sca_tdf::sca_in<st_complex> i_sig[NUMTRECS];
6
7      // Input and output ports in SystemC
8      sc_in<bool> i_clk;

```

```

9   sc_in<bool> i_reset;
10  sc_out<st_complex> o_sig[NUMTRECS];
11  sc_out<int> o_d2;
12  // Internal signal
13  sc_signal<st_complex> s_lpf2rec[NUMTRECS];
14  // Simple Sampling module from TDF to DE
15  GMSK_b **p_rec;
16  // Baseband filter in the TDF cluster
17  lowpass_filter_fe_b **p_lpfilter;
18  // Place to put test algorithms. Outputs to files
19  detection2<int (NUM_RECS)> *p_dectect2;
20  // Constructor
21  SC_CTOR(multi_antennas_fe_b)
22  {
23      int i; char stbuf[200];
24      p_rec = new GMSK_b* [NUMTRECS];
25      p_lpfilter = new lowpass_filter_fe_b* [NUMTRECS];
26      for (i = 0; i < NUMTRECS; i++)
27      {
28          sprintf(stbuf, "lowpass_filter_fe_b(%d)", i);
29          p_lpfilter[i] = new lowpass_filter_fe_b(stbuf, 1, i);
30          p_lpfilter[i]->i_sig(i_sig[i]);
31          p_lpfilter[i]->o_sig(s_lpf2rec[i]);
32          sprintf(stbuf, "GMSK_b(%d)", i);
33          p_rec[i] = new GMSK_b(stbuf);
34          p_rec[i]->i_clk(i_clk);
35          p_rec[i]->i_reset(i_reset);
36          p_rec[i]->i_sig(s_lpf2rec[i]);
37          p_rec[i]->o_sig(o_sig[i]);
38      }
39      p_dectect2 = new detection2<int (NUM_RECS)> ("detection2");
40      p_dectect2->i_clk(i_clk);
41      p_dectect2->i_reset(i_reset);
42      for (i = 0; i < int (NUM_RECS); i++)
43      {
44          p_dectect2->i_sig[i] (s_lpf2rec[i]);
45      }
46      p_dectect2->o_d(o_d2);
47  }
48  ~multi_antennas_fe_b();
49  };

```

Listing 5.8: *The lowpass_filter_fe_b class.*

```

1  SCA_TDF_MODULE(lowpass_filter_fe_b)
2  {
3      // Input port in TDF cluster
4      sca_tdf::sca_in<st_complex> i_sig;
5      // Output port in SystemC
6      sca_tdf::sc_out<st_complex> o_sig;
7      //Constructor
8      lowpass_filter_fe_b( sc_core::sc_module_name, double h0_in = 1.0, int o_in =
9          0)
          : h0(h0_in), o(o_in) {}

```

```
10 // Initialization of the filter.
11 void initialize()
12 {
13     if (FILTERBANDWIDTH == 0) // 15kHz
14     {
15         // order 8
16         num(0) = 1.00112915039063e+31;
17         den(0) = 1.00112915039063e+31;
18         den(1) = 6.84215830592126e+27;
19         den(2) = 2.33811642908513e+24;
20         den(3) = 5.18419402882565e+20;
21         den(4) = 8.12795636893894e+16;
22         den(5) = 9216344940134.49;
23         den(6) = 738960254.130606;
24         den(7) = 38443.7317161226;
25         den(8) = 1.0;
26     } else if (FILTERBANDWIDTH == 1) // 25kHz
27     {
28         // order 8
29         num(0) = 5.96046447753908e+32;
30         den(0) = 5.96046447753908e+32;
31         den(1) = 2.44418663763190e+29;
32         den(2) = 5.01139495260014e+25;
33         den(3) = 6.66691618933339e+21;
34         den(4) = 6.27157127232942e+17;
35         den(5) = 42668263611733.7;
36         den(6) = 2052667372.58502;
37         den(7) = 64072.8861935377;
38         den(8) = 1.0;
39     } else { // All pass
40         num(0) = 1;
41         den(0) = 1;
42     }
43 }
44 // Activation process
45 void processing()
46 {
47     // DC offset
48     if DCOFFSET == 1
49         o_sig.write( st_complex(ltf_ndI( num, den, i_sig.read().real, h0 ) +
50                               g_syspara->m_dcoffset[2*o], ltf_ndQ( num, den, i_sig.read().imag, h0
51                               ) + g_syspara->m_dcoffset[2*o+1]));
52     else
53         o_sig.write( st_complex(ltf_ndI( num, den, i_sig.read().real, h0 ),
54                               ltf_ndQ( num, den, i_sig.read().imag, h0 )));
55     endif
56 }
57 private:
58     sca_tdf::sca_ltf_nd ltf_ndI; // Laplace transfer 'function'
59     sca_tdf::sca_ltf_nd ltf_ndQ; // Laplace transfer 'function'
60     sca_util::sca_vector<double> num, den; // Numerator and denominator
61     coefficients
62     double h0; // DC gain
63     int o; // Antenna no.
64 };
```

Listing 5.9: *The refresh_sys class.*

```

1  SC_MODULE(refresh_sys)
2  {
3      sc_in<bool> i_clk1;
4      sc_in<bool> i_clk2;
5      sc_out<int> o_num_sigs;
6      // The first updating process
7      void seq_proc_refresh_sys1()
8      {
9          if (i_clk1.read() == 1)
10         {
11             int i;
12             g_syspara->m_global_time = sc_time_stamp().to_seconds();
13             double Δ_time = g_syspara->m_global_time - g_syspara->m_global_time_l;
14 #if ORGANIZEDCELLS == 1 // Move SOTDFMA cells
15                 g_syspara->p_cellmanag->MoveGrid(Δ_time * g_syspara->m_velo_sat);
16 #endif
17 #if SCINTONOFF == 1 // Update ionosphere scintillation 'for' cells
18                 g_syspara->p_cellmanag->UpdateScintinCells();
19 #endif
20                 for (i = 0; i < g_syspara->m_num_user; i++)
21                 { // Move ships
22                     g_syspara->update_spatialpara_u(Δ_time, i);
23                     g_syspara->checktrans_u(i); // Ship transmission control
24                 }
25                 g_syspara->m_global_time_l = g_syspara->m_global_time;
26             }
27         }
28     }
29     // The second updating process
30     void seq_proc_refresh_sys2()
31     {
32         int count = 0;
33         if (i_clk2.read() == 1)
34         {
35             int i;
36             g_syspara->m_global_time = sc_time_stamp().to_seconds();
37 #if ORGANIZEDCELLS == 1 // Update AIS frames 'for' cells
38                 g_syspara->p_cellmanag->UpdateCellFrameStartTime(g_syspara->
39                     m_global_time);
40 #endif
41                 for (i = 0; i < g_syspara->m_num_user; i++)
42                 {
43                     g_syspara->checktrans_u(i); // Ship transmission control
44                     if (g_syspara->p_status_u[i] == 1)
45                         { // Count the number of active ships
46                             count++;
47                         }
48                 }
49                 o_num_sigs.write(count);
50             }
51         } //Constructor
52         SC_CTOR(refresh_sys)

```

```
53     {  
54         SC_METHOD(seq_proc_refresh_sys1);  
55         sensitive_pos<<i_clk1;  
56         SC_METHOD(seq_proc_refresh_sys2);  
57         sensitive_pos<<i_clk2;  
58     }  
59 };
```

Chapter 6

A Demonstration System of Advanced Receivers for Satellite AIS

6.1 Motivation and Overview

In the former chapters, a simulation model is developed for testing the advanced receivers for Satellite AIS. This work is done in a virtual environment. So far no real AIS signal is seen. This chapter steps further to set up a demonstration system of the advanced receivers (See Fig. 6.1). This is a minimum setup of a functional AIS network. The purpose is to test the receivers on real signals coming from the frequency bands and to validate the solution by visualizing the received data. In this system, the signals are not from the satellite in space but they are still useful for proving the following points.

1. The receivers work well on real signals.
2. The receivers work well on examples of message collision.

The signals are collected from the hardware receivers on the 17th floor of the EWI building, 68 meters high above the ground. The signals come from the real transmitters in several SOTDMA cells around the Hook of Holland and the harbor of Rotterdam.

Luckily, the two types of message collisions are observed. The received signal power varies in a range more than 40 dB. The only missing effect is the Doppler frequency shift because the received signals are not from a real LEO satellite running in orbit. This problem is left for future work, although the designed receivers are theoretically not sensitive to the Doppler frequency shifts. In fact, some transmitters have poor crystals for the local oscillators (LO) and frequency

drifts are distinct for different crystals and could be relatively large up to several kHz. This effect is equivalent to the Doppler frequency shifts.

Fig. 6.2 shows the block diagram of this demonstration system. The system consists of five major components.

1. The AIS receiver located in the D3Lab consists of an RF frontend for converting the AIS signals to baseband analog signals, an oscilloscope for digitizing the baseband analog signals, a computer (Computer 1) for separation and demodulation of the signals and decoding the encrypted messages to binary data in forms of NMEA sentences.
2. The message dispatching network includes a host computer (Computer 1) for sending NMEA sentences via UDP, routers for connecting the computers and routing the UDP packets, and a client computer (Computer 2) for analyzing the AIS data.
3. The AIS data analyzer (Computer 2) located in the room HB17.140 records the NMEA sentences, explains the embedded navigational information, and organizes these data into a database. It writes the processed information into a disk file and uploads it to a web server through FTP.
4. The web server draws the data from the file onto the overlays of a Google map and makes the graphical information available to all internet users.
5. In user terminals (such as desktops, laptops, tablets and smart phones), the dynamic information are shown on a web page opened in a web explorer.

6.2 Hardware Receiver

The AIS receiver is implemented partially in hardware and partially in software. The hardware part consists of four identical RF frontend boards and an oscilloscope used as programmable analog-to-digital converters (ADC). The RF frontend board is EV9920B, a product from CML. EV9920B is an evaluation board for the chip CMX992, which is a quadrature receiver. EV9920B has the following functions:

1. It is tunable to the AIS frequency band at 162 MHz.
2. It provides analog I/Q outputs at the baseband.
3. It accepts external local oscillators for expansion to an array.

Fig. 6.3 shows the block diagram of EV9920B. The RF signal coming from the air enters EV9920B at J9, goes through a bandpass filter, a low noise amplifier and an attenuation circuit network, and then is fed into the first mixer where the signal is down converted to the intermediate frequency band at 45 MHz. A 4-pole

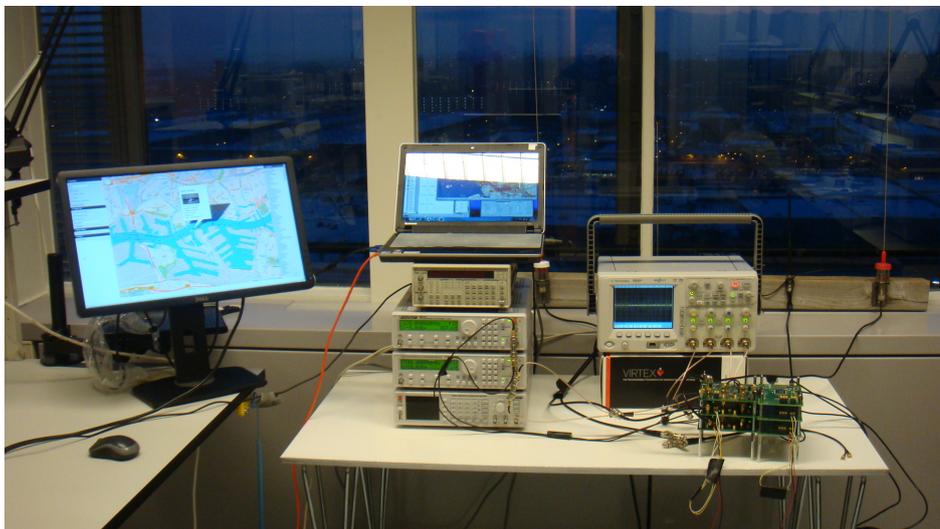


Figure 6.1: *The view of the AIS receiver in D3Lab and the map for the ship tracking (on a wide screen at the left side). Other components are not shown here.*

crystal filter with bandwidth 15 kHz filters the IF signal. A variable gain amplifier controls the power of the IF signal. The IF signal is then down converted again by a pair of I/Q mixers to provide baseband I/Q signals. The I/Q signals are filtered into a bandwidth of 100 kHz and available at the SMA connectors J23 and J22. EV9920B is controlled by PE0002, a configurable controller board from CML.

The original EV9920B uses the default configuration for working at a frequency band at 460 MHz. For a special configuration for the AIS frequency band at 162 MHz, the resistors, capacitors and inductors in the impedance matching network for the input of the first mixer are modified according to the manual [112](documented on page 36 in Section 7.5 in the manual).

The equation describing the relation of RFIN f_{RXIN} and RXLO f_{RXLO} is

$$f_{RXLO} = N_{RXLODIV} (f_{RXIN} + 45 \times 10^6), \quad (6.1)$$

where $N_{RXLODIV} = \{2, 4\}$ is the value of internal frequency divider in CMX992. To accept an external IFLO, the VCO in EV9920B is modified according to the manual [108](documented on page 28 in Section 6.2.3 in the manual). Table 6.1 shows the configuration of external LO frequencies of EV9920B for the AIS frequency band.

Four Metz AIS antennas are fixed on a wooden bar with separation of 0.92 meter. The baseband I channel, RXI, of each EV9920B is connected to one of the four observation channels of the Agilent oscilloscope DSO6104A. DSO6104A is mainly used to digitizing the analog signals. DSO6104A can be also used to

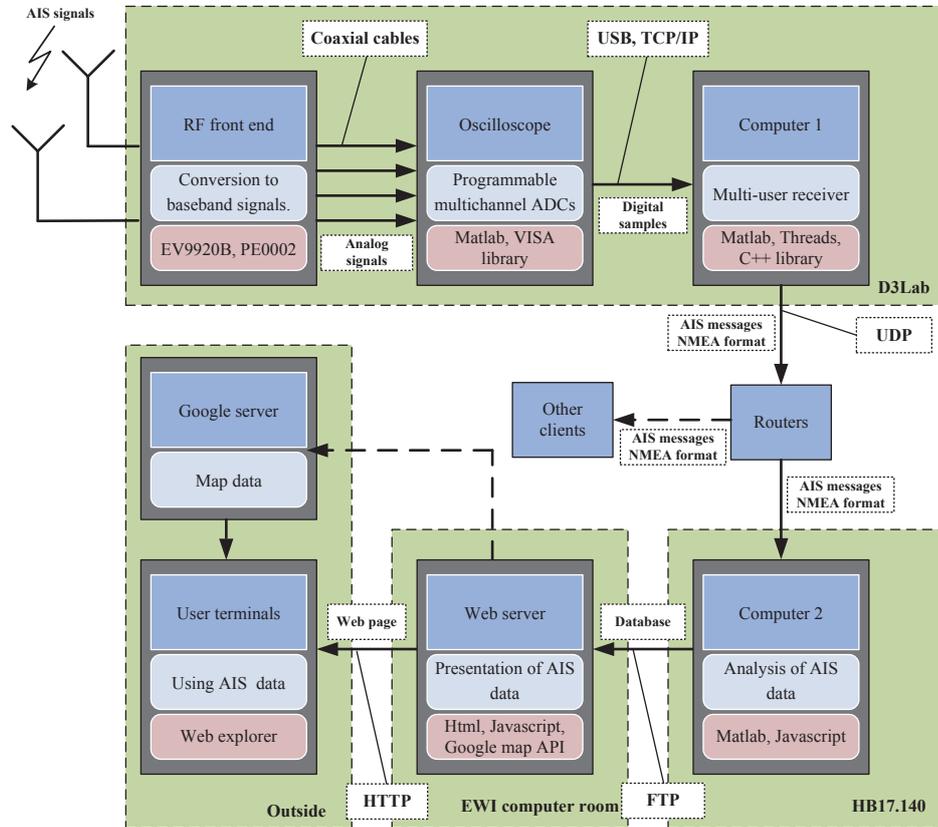


Figure 6.2: *The block diagram of the demonstration system.*

observe and to calibrate the array, such as measuring DC offsets, imbalance and noise levels of I/Q channels. DSO6104A provides the storage of samples visible on its screen to a USB stick or the transfer of blocks of samples to the PC through a USB or TCP/IP connection. A laptop computer is used as the host PC to get the ADC samples from DSO6104A. MATLAB routines are programmed upon the VISA library (provided by Agilent [113, 114]) to automatically control the sampling and the transfer of the ADC samples of the four observation channels in DSO6104A. DSO6104A allows a sampling speed up to 4 GSa/s. The processing and sampling is in realtime if DSO6104A collects a block of ADC samples for a time period of 10 seconds every time.

In MATLAB routines, the compensation for the DC offsets and the I/Q imbalance are carried out. The AIS receivers are implemented according to Chapter 3. The sampling in DSO6104A and the processing in MATLAB are parallel as a result of a careful sequencing the VISA commands (See Listing 6.1).

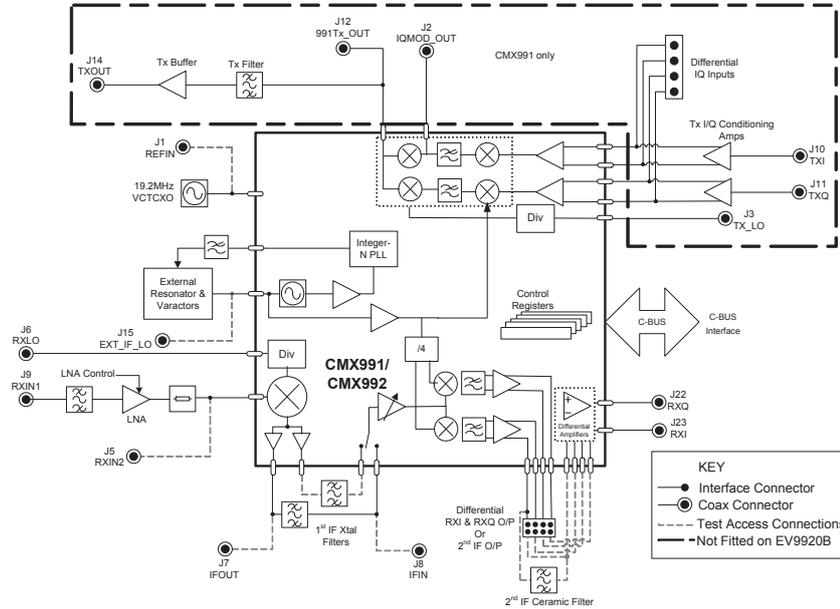


Figure 6.3: The block diagram of EV9920B (from [108]).

Table 6.1: The configuration of external LO frequencies of EV9920B for the AIS frequency band.

| | AIS 1, 161.975 MHz | AIS 2, 162.025 MHz |
|-------------------|-----------------------------------|-----------------------------------|
| External RXLO J6 | 207.015 N_{LODIV} MHz at -5 dBm | 207.065 N_{LODIV} MHz at -5 dBm |
| External IFLO J15 | 180 MHz at -10 dBm | 180 MHz at -10 dBm |

Listing 6.1: The MATLAB routine for AIS receivers.

```

1  % UDP connection to Computer 2
2  udpobj2 = udp('xxx.xxx.xx.xxx', 'RemotePort', 12346, 'LocalPort', 12344);
3  % ...
4  fopen(udpobj2);
5  % ...
6  % Set up VISA connection
7  %visaObj = visa('agilent', 'TCP/IP0:mso.d3lab:inst0:INSTR'); % TCP/IP
8  visaObj = visa('agilent', 'USB0::0x0957::0x1754::MY44008128::0:INSTR'); % USB
9  % ...
10 fopen(visaObj); % Open VISA connection
11 % ...
12 ACQruns = 10^5; % Total number of loops
13 runindex = 0; % Index for blocks of data
14 data = []; % Data matrix for processing
15 while runindex < ACQruns % loops for blocks of data
16     % Digitize 4 channels
17     fprintf(visaObj, 'DIGITIZE CHAN1,CHAN2,CHAN3,CHAN4');

```

Table 6.2: *The encoded AIS messages in NMEA format. The first character following “A,” is the type of messages.*

```

!AIVDM,1,1,,A,13aDB>0P000Cw6<Md=Qh0?wP08M3,0*60
!AIVDM,1,1,,A,233qb:5P@30CAh<MeQQbm:IT00S5,0*2B
!AIVDM,1,1,,A,33f2@<05@v0CeUpMdEHUJIHt05bi,0*6B
!AIVDM,1,1,,A,4h2E3MQulGcAMPcGqVMd@sG00L>L,0*23
!AIVDM,2,1,1,A,53aC2=T000010?G7G@0dEHTr0<0000000000001S0P?
44u:@C0Dhj2AA0CHh00,0*37
!AIVDM,2,2,1,A,0000000000,2*25
!AIVDM,1,1,,A,633'@p00U@oH06P0L0,4*6C
!AIVDM,1,1,,A,74SOpT00U@oL,0*2B
!AIVDM,1,1,,A,83aEPHhj2d<dteeMeinho?bP5tP0,0*6D
!AIVDM,1,1,,A,D02E35Pmtg6D000000000000000,2*4D

```

```

18 % Process data in parallel to the sampling period
19 % ...
20 clear data;
21 %...
22 % Get ADC samples in WORD format
23 fprintf(visaObj, 'WAV:DATA?');
24 %...
25 % Close VISA connection
26 fclose(visaObj);
27 %...
28 % Convert ADC samples in WORD format to floating point data
29 %...
30 % Advance one block
31 runindex = runindex + 1;
32 %...
33 % Open VISA connection
34 fopen(visaObj);
35 end
36 %...
37 fclose(visaObj); fclose(udpobj2);

```

The demodulated binary AIS messages are encoded into NMEA sequences for transmission in the network. Table 6.2 shows examples of NMEA sequences.

6.3 Analysis of AIS Data

The analysis of AIS data is done in a desktop computer (Computer 2) in room HB17.140 on the same floor of D3Lab. In this computer, the NMEA sequences collected from a UDP connection are recovered back to AIS binary sequences, which are parsed and classified. The navigational information of each ship decoded from the messages is stored in a structural database (See Fig. 6.4). The code is in MATLAB with Javascript developed from an online source, AIS Toolbox [115]. AIS Toolbox gives a good platform to process the messages. The main body of AIS Toolbox is reused but a lot of bugs in code are fixed. Additional

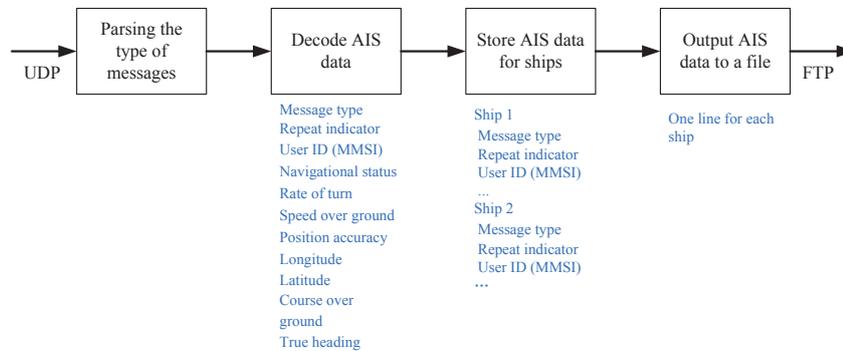


Figure 6.4: *The processing stages in the matlab routines for the analysis of AIS data.*

functions are added in the code to create the database in the format defined by ourselves. The frequency of uploading the data to the web server is set to every 10 seconds, equal to the processing time of the AIS receiver. Only the messages useful to Satellite AIS are processed including Type 1 to Type 4 messages for regular position reports.

6.4 Presentation of AIS Data

The illustration of AIS data on an online map is very attractive for demonstration. By using the Google map API, it is possible to do it (See Fig. 6.5). The web page automatically refreshes the AIS data every 20 seconds.

6.5 Some Results

Fig. 6.6 shows the waveform of the I/Q signals at baseband from EV9920B in a 10 seconds observation. The spikes are AIS messages. The noise floor is also visible. It is seen that the signal power of messages from near and far field has large difference in amplitude.

Fig. 6.7 shows the observed waveform of two partially overlapping AIS messages. Fig. 6.8 shows the amplitude of the signals. This collision is probably caused by unsynchronized transmission. The overlapping of messages of equal or different lengths are also observed.

6.6 Conclusions

In this chapter, a demonstration system of the advanced receivers for Satellite AIS is introduced. The system is made upon available devices, equipment and

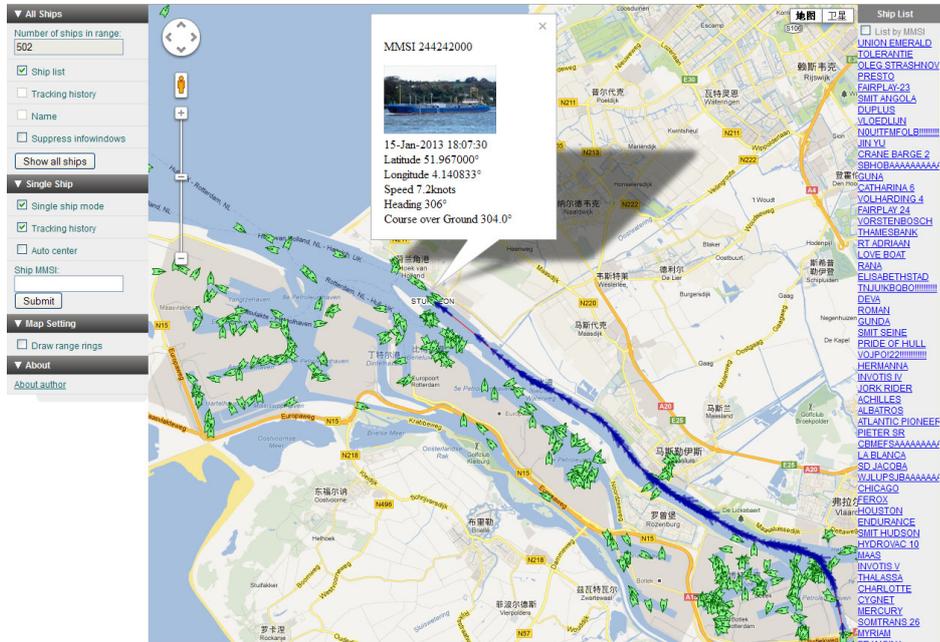


Figure 6.5: The web page presenting the AIS data on a Google map.

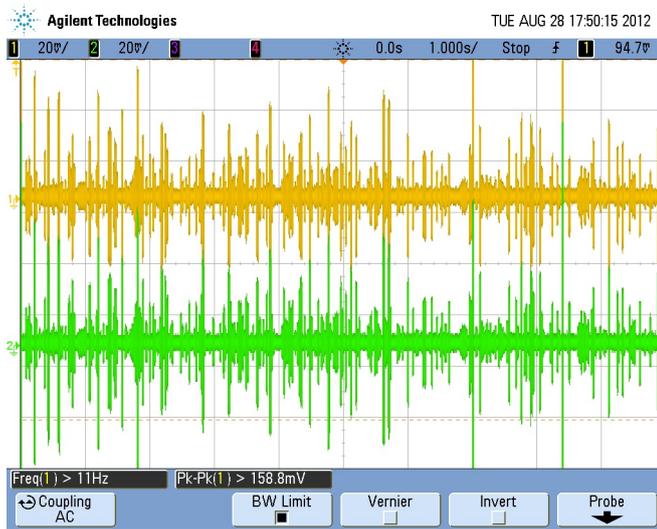


Figure 6.6: The waveform of the received I/Q signals at baseband in a 10 seconds observation.

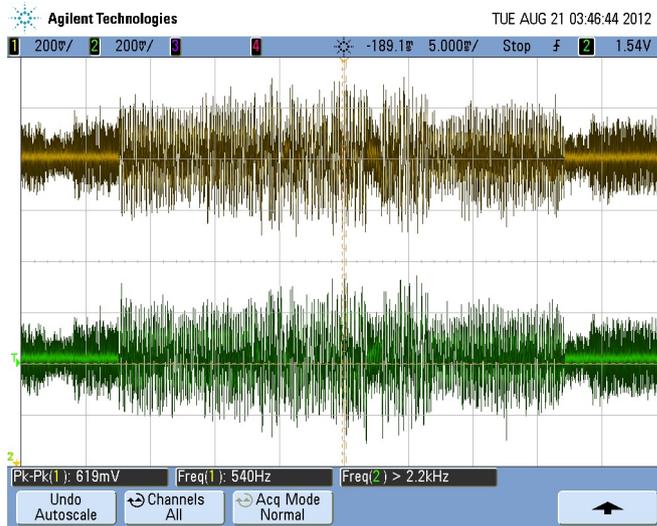


Figure 6.7: *The observed waveform (I/Q) of two partially overlapping messages.*

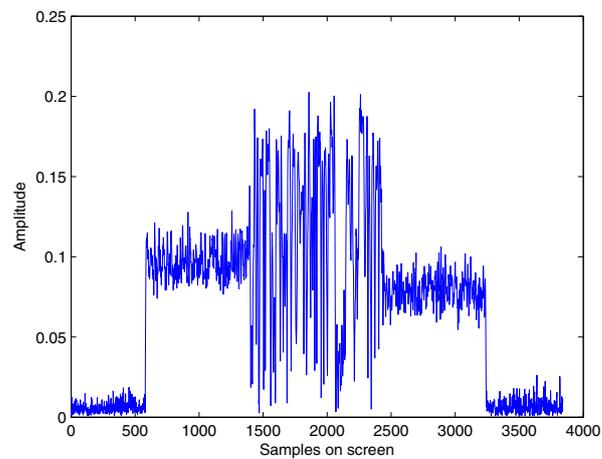


Figure 6.8: *The amplitude of the waveform shown in Fig. 6.7.*

software. The system provides real data for the design of receivers. The performance of the receivers are displayed on a map so that the developers and the users can watch it. The structure of this system can help building experimental systems for other communication systems.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The growing demands for expansion of the existing tracking systems to the global coverage call for innovative solutions. The new applications brings unprecedented challenges to the existing systems designed in the last century. The development of new techniques are required to keep the new systems compatible to the currently running systems without major updating of the user-side equipment. AIS is one example of this problem in which LEO satellites are used to track ships anywhere on the earth. This new application is called Satellite AIS. In Satellite AIS, the receivers face in-cell and inter-cell interfering sources from the system itself as well as other distortion to the signals. In this thesis, new blind beamforming techniques have been developed for the satellite receivers for Satellite AIS to combat these interferences. The new beamforming techniques are proposed through three consecutive chapters associated with their foundation, evolution, and application. A software testbed in SystemC-AMS with MATLAB has been developed to simulate the receivers. An experimental hardware platform has been constructed to test the receivers on real AIS signals and to visualize the results to developers and users.

The thesis has developed a new beamforming technique for principal subspace tracking in a block stationary scenario. The purpose of computing the principal subspace is to construct a beamformer for strengthening the signal against noise. The purpose of selecting the tracking form is to reduce the computational complexity in computing the subspace. In Chapter 2, a Hyperbolic URV decomposition and its updating algorithm (SURV) for subspace tracking is proposed. It is a non-iterative algorithm that provides exact rank estimates and very good principal subspace estimates in comparison with the SVD with the same rank threshold. The subspace is provided in terms of an orthogonal basis \mathbf{Q}_B . The proposed updating algorithm uses at most one hyperbolic rotation per vec-

tor update and provides a numerically stable computation of the “SSE-2” Schur subspace estimator (unlike the previously proposed algorithm in [38]).

The proposed updating algorithm has the computational complexity $O(m^2)$ per size- m update vector (similar to a QR update), and has constant $O(m^2)$ memory requirements. The algorithm is based on a sliding-window update scheme, but is easily tailored to exponential windowing by scaling matrix \mathbf{R} after every update. All operations in the proposed updating algorithm are local, consisting of elementary rotations and column permutations, and the computational flow is non-iterative, thus facilitating parallel implementations. The proposed algorithm can replace an SVD when only the subspace is of interest and the threshold on the noise power splitting the subspace is already given. The computational complexity is higher than some of the tracking algorithms with knowledge of the rank, such as PAST, but the prominent advantage is that the estimate is “exact” with known properties and does not rely on convergence.

Then the proposed beamforming technique is developed for a nonstationary scenario for suppressing partial overlapping interference in data blocks. In Chapter 3, blind beamforming techniques for suppressing asynchronous co-channel interference are proposed. The algorithms are based on subspace estimations using GSVD and GEVD. Subsequently, SURV is introduced as a computationally efficient replacement for the GSVD. Simulations have shown that these algorithms have essentially equal performance, close to that of a reference MMSE receiver with completely known target signals.

Expanding the scope of the application of beamforming techniques is the work in Chapter 4. In Chapter 4, a multi-user receiver for Satellite AIS is proposed. The proposed receiver uses a new blind beamforming technique based on GSVD to effectively improve the detection of AIS messages under asynchronous interferences. The beamforming algorithm can be easily implemented in a tracking form through SURV and the complexity is reduced. The whole receiver uses “blind” algorithms and requires less knowledge of the beam pattern, which facilitates the design of antenna arrays and satellites. Simulation results in two scenarios of Satellite AIS have confirmed the effectiveness of the proposed receivers and showed that the proposed receivers are superior to previous receivers.

In this thesis, not only theoretical supports for the proposed receivers are provided, but also the software testbed and the experimental hardware platform for testing the receivers, either in a virtual Satellite AIS or on real AIS signals, are developed. In Chapter 5, a SystemC-AMS model of a dynamic large-scale network of Satellite AIS is presented, verified and validated. It has been demonstrated how SystemC-AMS is used as an efficient tool to model such a complex system. The AIS link layer and the AIS physical layer are analyzed and abstracted for modeling. With the help of the TDF modeling style in SystemC-AMS and its seamless integration to SystemC domain, the activity of the both layers and a cross-layer system controller are precisely timed. The modular modeling style provides sufficient flexibility in modeling real objects, such as transmitters, physical channels and receivers. Many important factors of Satellite AIS are modeled

in the TDF domain, such as the Doppler frequency shift, the path loss, DOA, the propagation time delay, the real-time transmission, and the collision of messages. The open structure of this model allows further modification and extension for different simulation scenarios.

In Chapter 6, a demonstration system of the advanced receivers for Satellite AIS has been introduced. The system is made upon available devices, equipment and software. The system provides real data for the design of the receivers. The performance of the receivers are displayed on a map so that the developers and the users can watch it. The structure of this system can help building experimental systems for other communication systems.

7.2 Future Work

For the principal subspace tracking based on SURV, the future work can include the derivation of a spherical update, where the noise subspace is not tracked but replaced by an average property.

For the beamforming techniques for suppressing partially overlapping packets, the future discussion can focus on the theoretical analysis of the detection parameters for identifying the target or the interference signals.

For the proposed advanced receiver, the future work turn into the design of better antenna arrays, such as a miniaturized dual polarized VHF antenna array for AIS [116]. It is also interesting to develop similar receivers for other identification systems, such as ADS-B for tracking airplanes and systems for car driving assistance. Direct extensions to wideband signals (e.g. separating partially overlapping WIFI/Bluetooth signals, or two concurrent speech signals) are possible.

For the modeling method of SystemC-AMS from Chapter 5, the model can be extended when the system is upgraded. The same method can be applied to other (future) communication systems for saving the time cost during the hardware implementation.

For the construction of the hardware platform in Chapter 6, it is hoped that it can give readers some inspiration to build their own platforms.

Bibliography

- [1] M. Simon and C. Wang, "Differential detection of Gaussian MSK in a mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. 33, no. 4, pp. 307–320, Nov. 1984.
- [2] A. Yongacoglu, D. Makrakis, and K. Feher, "Differential detection of GMSK using decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 6, pp. 641–649, Jun. 1988.
- [3] I. Korn, "GMSK with differential phase detection in the satellite mobile channel," *IEEE Trans. Commun.*, vol. 38, no. 11, pp. 1980–1986, Nov. 1990.
- [4] *REPORT ITU-R M.2084: Satellite detection of automatic identification system messages*, ITU Std., 2006.
- [5] O. F. H. Dahl, "Space-based AIS receiver for maritime traffic monitoring using interference cancellation," Master's thesis, Norwegian University of Science and Technology, Department of Electronics and Telecommunications, 2006.
- [6] M. A. Cervera, A. Ginesi, and K. Eckstein, "Satellite-based vessel automatic identification system: A feasibility and performance analysis," *Int. J. Satell. Commun. Network.*, 2009.
- [7] T. Eriksen, G. Høy, B. Narheim, and B. J. Meland, "Maritime traffic monitoring using a space-based AIS receiver," *Acta Astronautica*, vol. 58, no. 10, pp. 537–549, 2006.
- [8] *RECOMMENDATION ITU-R M.2169: Improved Satellite Detection of AIS*, ITU Std., 2009.
- [9] D. J. Nelson and J. R. Hopkins, "GMSK co-channel demodulation," in *Proc. SPIE 7444*, 2009, p. 74440V.
- [10] R. Prevost, D. Bonacci, J. LeMaitre, J.-P. Millerioux, and J.-Y. Tourneret, "Partial CRC-assisted error correction of AIS signals received by satellite," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 1951–1955.

- [11] C. C. Paige and M. A. Saunders, "Towards a generalized singular value decomposition," *SIAM Journal on Numerical Analysis*, vol. 18, no. 3, pp. 398–405, Jun. 1981.
- [12] A.-J. van der Veen and A. Paulraj, "An analytical constant modulus algorithm," *IEEE Trans. Signal Process.*, vol. 44, no. 5, pp. 1136–1155, May 1996.
- [13] M. Picard, M. Oularbi, G. Flandin, and S. Houcke, "An adaptive multi-user multi-antenna receiver for satellite-based AIS detection," in *Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th*, 2012, pp. 273–280.
- [14] M. Zhou and A.-J. van der Veen, "Stable subspace tracking algorithm based on a signed URV decomposition," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3036–3051, Jun. 2012.
- [15] M. Zhou and A.-J. van der Veen, "Blind beamforming techniques for separating partially overlapping data packets," *submitted*, 2015.
- [16] M. Zhou and A.-J. van der Veen, "Robust multi-user receiver for satellite AIS using blind beamforming techniques and tracking," *submitted*, 2015.
- [17] M. Zhou, R. van Leuken, and H. J. L. Arriens, "Modeling a configurable resistive touch screen system using SystemC and SystemC-AMS," in *20th annual workshop on circuits, systems and signal processing-ProRISC*, Nov. 2009, pp. 393–398.
- [18] M. Zhou and R. van Leuken, "Modeling an uplink for multi-antenna satellite receivers in AIS-like networks using SystemC-AMS," in *21st annual workshop on circuits, systems and signal processing-ProRISC*, Nov. 2010.
- [19] M. Zhou and A.-J. van der Veen, "Stable subspace tracking algorithm based on signed URV decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 2720–2723.
- [20] M. Zhou and A.-J. van der Veen, "Improved subspace intersection based on signed URV decomposition," in *Proc. of the Asilomar Conf. on Signals, Syst., and Comput.*, Pacific Grove, California, USA, Nov. 2011.
- [21] M. Zhou and R. van Leuken, "SystemC-AMS model of a dynamic large-scale satellite-based AIS-like network," in *Forum on specification and Design Languages*, Oldenburg, Germany, Sept. 2011, pp. 24–31.
- [22] M. Zhou and A. van der Veen, "Improved blind separation algorithm for overlapping secondary surveillance radar replies," in *Proc. IEEE CAMSAP*, San Juan, Puerto Rico, Dec. 2011, pp. 181–184.

- [23] M. Zhou, A.-J. van der Veen, and R. van Leuken, "Multi-user LEO-satellite receiver for robust space detection of AIS messages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012.
- [24] M. Zhou and A.-J. van der Veen, "A subspace tracking algorithm for separating partially overlapping data packets," in *Proc. of the Seventh IEEE Sensor Array and Multichannel Signal Process. Workshop (SAM)*, Hoboken, NJ, USA, Jun. 2012.
- [25] A.-J. van der Veen and M. Zhou, "Subspace intersection tracking using GSVD and the signed URV algorithm," in *TUM-IAS Workshop on Novel Numerical Methods*, TUM Institute for Advanced Study (TUM-IAS), Lichtenbergstraße 2 a, Garching, July 2013.
- [26] M. Zhou and A.-J. van der Veen, "Bind beamforming techniques for automatic identification system using GSVD and tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014.
- [27] A.-J. van der Veen and M. Zhou, "Subspace intersection tracking using the signed URV algorithm," in *ITG Workshop on Smart Antennas*, TU-ilmeneau, Ilmenau, Germany, Mar. 2015.
- [28] A.-J. van der Veen, E. F. Deprettere, and A. L. Swindlehurst, "Subspace based signal analysis using singular value decomposition," *Proc. IEEE*, vol. 81, no. 9, pp. 1277–1308, Sep. 1993.
- [29] A.-J. van der Veen, "Algebraic methods for deterministic blind beamforming," *Proc. IEEE*, vol. 86, no. 10, pp. 1987–2008, Oct. 1998.
- [30] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, no. 8, pp. 1327–1343, Aug. 1990.
- [31] T. Chan, "Rank revealing QR factorizations," *Linear Algebra and its Applications*, vol. 88-89, pp. 67–82, 1987.
- [32] C. Bischof and G. Schroff, "On updating signal subspaces," *IEEE Trans. Signal Proc.*, vol. 40, no. 1, pp. 96–105, Jan. 1992.
- [33] G. W. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1535–1541, 1992.
- [34] G. W. Stewart, "Updating a rank-revealing ULV decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 2, pp. 494–499, 1993.
- [35] A.-J. van der Veen, "A Schur method for low-rank matrix approximation," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 1, pp. 139–160, 1996.

- [36] A.-J. Boonstra and A.-J. van der Veen, "Gain calibration methods for radio telescope arrays," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 25–38, Jan. 2003.
- [37] R. D. Fierro, P. C. Hansen, and P. S. K. Hansen, "UTV Tools: Matlab templates for rank-revealing UTV decompositions," *Numerical Algorithms*, vol. 20, no. 2-3, pp. 165–194, Aug. 1999.
- [38] A.-J. van der Veen, "Subspace tracking using a constrained hyperbolic URV decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, May 1998, pp. 1953–1956.
- [39] X. Doukopoulos and G. Moustakides, "Fast and stable subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 4, pp. 1452–1465, Apr. 2008.
- [40] I. Karasalo, "Estimating the covariance matrix by signal subspace averaging," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 34, no. 1, pp. 8–12, Feb. 1986.
- [41] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.
- [42] Y. Miao and Y. Hua, "Fast subspace tracking and neural network learning by a novel information criterion," *IEEE Trans. Signal Process.*, vol. 46, no. 7, pp. 1967–1979, Jul. 1998.
- [43] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *Signal Processing Letters, IEEE*, vol. 7, no. 3, pp. 60–62, Mar. 2000.
- [44] Y. Hua, Y. Xiang, T. Chen, K. Abed-meraim, and Y. Miao, "A new look at the power method for fast subspace tracking," *Digital Signal Processing*, vol. 9, no. 4, pp. 297–314, Oct. 1999.
- [45] B. Yang, "An extension of the PASTd algorithm to both rank and subspace tracking," *IEEE Signal Process. Lett.*, vol. 2, no. 9, pp. 179–182, Sep. 1995.
- [46] R. Badeau, G. Richard, and B. David, "Fast and stable YAST algorithm for principal and minor subspace tracking," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3437–3446, Aug. 2008.
- [47] G. W. Stewart, "On the stability of sequential updates and downdates," *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2642–2648, Nov. 1995.
- [48] Y.-J. J. Wu, "Downdating a rank-revealing URV decomposition," University of Maryland at College Park, College Park, MD, USA, Tech. Rep., 1995.
- [49] H. Park and L. Eldén, "Downdating the rank-revealing URV decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 1, pp. 138–155, 1995.

- [50] M. Stewart and G. W. Stewart, "On hyperbolic triangularization: Stability and pivoting," *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 4, pp. 847–860, 1998.
- [51] R. Badeau, K. Abed-Meraim, G. Richard, and B. David, "Sliding window orthonormal PAST algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, Apr. 2003, pp. 261–264.
- [52] R. Badeau, G. Richard, and B. David, "Sliding window adaptive SVD algorithms," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 1–10, Jan. 2004.
- [53] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2931–2941, Aug. 2005.
- [54] J. Götze and A.-J. van der Veen, "On-line subspace estimation using a Schur-type method," *IEEE Trans. Signal Process.*, vol. 44, no. 6, pp. 1585–1589, Jun. 1996.
- [55] J. Götze, M. Haardt, and J. A. Nossek, "Subspace estimation using unitary Schur-type methods," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1995, pp. 1153–1156.
- [56] A. Bunse-Gerstner, "An analysis of the HR algorithm for computing the eigenvalues of a matrix," *Lin. Alg. Applicat.*, vol. 35, pp. 155–173, 1981.
- [57] R. Onn, A. Steinhardt, and A. Bojanczyk, "The hyperbolic singular value decomposition and applications," *IEEE Trans. Signal Process.*, vol. 39, no. 7, pp. 1575–1588, Jul. 1991.
- [58] R. Horn and C. Johnson, *Matrix analysis*. Cambridge University Press, 1990.
- [59] A. Bojanczyk, S. Qiao, and A. O. Steinhardt, "Unifying unitary and hyperbolic transformations," *Lin. Alg. Applicat.*, vol. 316, pp. 183–197, 2000.
- [60] J. H. Wilkinson, Ed., *The algebraic eigenvalue problem*. New York, NY, USA: Oxford University Press, Inc., 1988.
- [61] A. Edelman, "The distribution and moments of the smallest eigenvalue of a random matrix of Wishart type," *Lin. Alg. Applicat.*, vol. 159, pp. 55–80, Dec. 1991.
- [62] C. A. Tracy and H. Widom, "On orthogonal and symplectic matrix ensembles," *Commun. Math. Phys.*, vol. 177, pp. 727–754, 1996.
- [63] M. Johnstone, "On the distribution of the largest eigenvalue in principal components analysis," *Ann. Statist.*, vol. 29, pp. 295–327, 2001.

- [64] J. Andrews, W. Choi, and R. Heath, "Overcoming interference in spatial multiplexing MIMO cellular networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 95–104, Dec. 2007.
- [65] S. Karimifar and J. Cavers, "Achieving high-capacity narrowband cellular systems by means of multicell multiuser detection," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 945–953, Mar. 2008.
- [66] N. Petrochilos and A.-J. van der Veen, "Algebraic algorithms to separate overlapping secondary surveillance radar replies," *IEEE Trans. Signal Process.*, vol. 55, no. 7, pp. 3746–3759, Jul. 2007.
- [67] A. M. Kuzminskiy and Y. I. Abramovich, "Second-order asynchronous interference cancellation: Regularized semi-blind technique and non-asymptotic maximum likelihood benchmark," *Signal Processing*, vol. 86, no. 12, pp. 3849–3863, 2006.
- [68] A. M. Kuzminskiy and Y. I. Abramovich, "Non-stationary multiple-antenna interference cancellation for unsynchronized OFDM systems with distributed training," *Signal Processing*, vol. 89, no. 5, pp. 753–764, 2009.
- [69] J. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *IEE Proc. F (Radar and Signal Processing)*, vol. 140, no. 6, pp. 362–370, Dec. 1993.
- [70] C. Papadias, "Globally convergent blind source separation based on a multiuser kurtosis maximization criterion," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3508–3519, Dec. 2000.
- [71] R. Behrens and L. Scharf, "Signal processing applications of oblique projection operators," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1413–1424, Jun. 1994.
- [72] L. Scharf and M. McCloud, "Blind adaptation of zero forcing projections and oblique pseudo-inverses for subspace detection and estimation when interference dominates noise," *IEEE Tr. Signal Proc.*, vol. 50, no. 12, Dec. 2002.
- [73] N. Petrochilos, G. Galati, and E. Piracci, "Separation of SSR signals by array processing in multilateration systems," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 45, no. 3, pp. 965–982, 2009.
- [74] D. Callaerts, B. Moor, J. Vandewalle, W. Sansen, G. Vantrappen, and J. Janssens, "Comparison of svd methods to extract the foetal electrocardiogram from cutaneous electrode signals," *Medical and Biological Engineering and Computing*, vol. 28, no. 3, pp. 217–224, 1990.

- [75] P. Hansen and S. Jensen, "Prewhitening for rank-deficient noise in subspace methods for noise reduction," *IEEE Trans. Signal Process.*, vol. 53, no. 10, pp. 3718–3726, 2005.
- [76] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique based on second order statistics," *IEEE Trans. Signal Proc.*, vol. 45, no. 2, pp. 434–44, Feb. 1997.
- [77] G. Stewart, "An updating algorithm for subspace tracking," *IEEE Trans. Signal Proc.*, vol. 40, no. 6, pp. 1535–1541, Jun. 1992.
- [78] G. Stewart, "Updating a rank-revealing ULV decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 14, no. 2, pp. 494–499, Apr. 1993.
- [79] Z. D. Bai and Y. Q. Yin, "Limit of the smallest eigenvalue of a large dimensional sample covariance matrix," *Ann. Probab.*, vol. 21, no. 3, pp. 1275–1294, 1993.
- [80] K. Johansson, "Shape fluctuations and random matrices," *Communications in Mathematical Physics*, vol. 209, pp. 437–476, 2000.
- [81] C. A. Tracy and H. Widom, "Correlation functions, cluster functions, and spacing distributions for random matrices," *Journal of Statistical Physics*, vol. 92, pp. 809–835, 1998.
- [82] D. Paul, *Asymptotic distribution of the smallest eigenvalue of Wishart(N, n) when $N, n \rightarrow \infty$ such that $N/n \rightarrow 0$* . World Scientific, 2011.
- [83] *RECOMMENDATION ITU-R M.1371-4: Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band*, ITU Std., 2010.
- [84] M. Gallardo and U. Sorger, "Coherent receiver for AIS satellite detection," in *4th Int. Symp. on Commun., Control and Signal Process.*, Mar. 2010, pp. 1–4.
- [85] J. E. Hicks, J. S. Clark, J. Stocker, G. S. Mitchell, and P. Wyckoff, "AIS/GMSK receiver on FPGA platform for satellite application," in *Proc. SPIE 5819*, 2005, p. 403.
- [86] P. Burzigotti, A. Ginesi, and G. Colavolpe, "Advanced receiver design for satellite-based automatic identification system signal detection," *International Journal of Satellite Communications and Networking*, vol. 30, no. 2, pp. 52–63, 2012.
- [87] G. Colavolpe, T. Foggi, A. Ugolini, J. Lizarraga, S. Cioni, and A. Ginesi, "A highly efficient receiver for satellite-based automatic identification system signal detection," *International Journal of Satellite Communications and Networking*, 2014.

- [88] G. Eiden and T. Martinsen, "Technical note 4.1 vessel density mapping issue 4: Preparatory action for assessment of the capacity of spaceborne automatic identification system receivers to support eu maritime policy," Luxspace Sarl, Betzdorf, Luxembourg, Tech. Rep. No MARE/2008/06/SI2.517298, Oct. 2010.
- [89] Real-time vessel positions on Google Earth. [Online]. Available: <http://www.marinetraffic.com/cn/p/google-earth>
- [90] L. Foged, A. Giacomini, F. Saccardi, L. Tancioni, N. Fonseca, P. de Maagt, A. Di Cintio, G. Della Pietra, A. Caliumi, G. Duchini, and J. Baracco, "Design and testing of a miniaturized dual polarized VHF array element for AIS space application," in *Microwaves, Communications, Antennas and Electronics Systems (COMCAS), 2013 IEEE International Conference on*, Oct 2013, pp. 1–3.
- [91] M. Picard, M. Oularbi, G. Flandin, and S. Houcke, "An adaptive multi-user multi-antenna receiver for satellite-based AIS detection," in *Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th*, 2012, pp. 273–280.
- [92] Y. Lee, R. Chandrasekaran, and J. Shynk, "Separation of cochannel GSM signals using an adaptive array," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 1977–1987, Jul. 1999.
- [93] C. Budianu and L. Tong, "Channel estimation under asynchronous packet interference," *IEEE Trans. Signal Process.*, vol. 53, no. 1, pp. 333–345, Jan. 2005.
- [94] Aausat3. [Online]. Available: <http://www.space.aau.dk/aausat3/>
- [95] J. Li, P. Stoica, and Z. Wang, "On robust Capon beamforming and diagonal loading," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1702–1715, July 2003.
- [96] *RECOMMENDATION ITU-R M.1371-2: Technical characteristics for a universal shipborne automatic identification system using time division multiple access in the VHF maritime mobile band*, ITU Std., 2006.
- [97] M. A. Cervera and A. Ginesi, "On the performance analysis of a satellite-based AIS system," in *Signal Process. for Space Commun., 2008. SPSC 2008. 10th Int. Workshop on*, Oct. 2008, pp. 1–8.
- [98] A. Dembovskis, "Testbed for performance evaluation of SAT-AIS receivers," in *Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC), 2012 6th*, 2012, pp. 253–257.

- [99] *SystemC AMS extensions Users Guide*, OSCI Std., 2010.
- [100] K. Caluwaerts and D. Galayko, "SystemC-AMS modeling of an electromechanical harvester of vibration energy," in *Specification, Verification and Design Languages, 2008. FDL 2008. Forum on*, Sep. 2008, pp. 99–104.
- [101] E. Markert, M. Dienel, G. Herrmann, and U. Heinkel, "SystemC-AMS assisted design of an inertial navigation system," *Sensors Journal, IEEE*, vol. 7, no. 5, pp. 770–777, May 2007.
- [102] E. Markert, M. Dienel, G. Herrmann, D. Mueller, and U. Heinkel, "Modeling of a new 2D acceleration sensor array using SystemC-AMS," *Journal of Physics: Conference Series*, vol. 34, no. 1, p. 253, 2006.
- [103] P. Hartmann, P. Reinkemeier, A. Rettberg, and W. Nebel, "Modelling control systems in SystemC AMS-benefits and limitations," in *SOC Conference, 2009. SOCC 2009. IEEE International*, Sep. 2009, pp. 263–266.
- [104] M. Vasilevski, N. Beilleau, H. Aboushady, and F. Pecheux, "Efficient and refined modeling of wireless sensor network nodes using SystemC-AMS," in *Research in Microelectronics and Electronics, 2008. PRIME 2008. Ph.D.*, Apr. 2008, pp. 81–84.
- [105] M. Vasilevski, F. Pecheux, N. Beilleau, H. Aboushady, and K. Einwich, "Modeling refining heterogeneous systems with systemc-ams: application to wsn," in *Proceedings of the conference on Design, automation and test in Europe*, ser. DATE '08, 2008, pp. 134–139.
- [106] A. Vachoux, C. Grimm, and K. Einwich, "Analog and mixed signal modelling with SystemC-AMS," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 3, May 2003, pp. III–914–III–917.
- [107] S. Basu, E. MacKenzie, E. Costa, P. Fougere, J. Carlson, H., and H. Whitney, "250 MHz/GHz scintillation parameters in the equatorial, polar, and auroral environments," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 2, pp. 102–115, Feb. 1987.
- [108] *EV9910B/EV9920B Evaluation Kit User Manual*, UM9910/6, CML Microcircuits, 2010.
- [109] D. Randall, T. Ringler, R. Heikes, P. Jones, and J. Baumgardner, "Climate modeling with spherical geodesic grids," *Computing in Science and Engineering*, vol. 4, no. 5, pp. 32–41, Sep. 2002.
- [110] K. Sahr, D. White, and A. J. Kimerling, "Geodesic discrete global grid systems," *Cartography and Geographic Information Science*, vol. 30, no. 2, pp. 121–134, 2003.

- [111] Gridsphere MATLAB code. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/28842-gridsphere>
- [112] *CMX991/CMX992 RF Quadrature Transmitter and RF Quadrature/Low IF Receiver*, D/991_992/17, CML Microcircuits, 2012.
- [113] (2012) IO libraries suite 16.3. [Online]. Available: <http://www.home.agilent.com/>
- [114] V. Cherian. (2013) Capturing a waveform from an Agilent oscilloscope over a standard visa interface. [Online]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/28887>
- [115] R. T. Bye, “AIS toolbox for MATLAB v. 1.2,” Høgskolen i Ålesund, N-6025 Ålesund, Norway, Rapport Notat 2009/04, May 2009.
- [116] L. Foged, A. Giacomini, F. Saccardi, L. Tancioni, A. Di Cintio, G. Dell Pietra, A. Caliumi, G. Duchini, N. Fonseca, and J. Baracco, “Miniaturized dual polarized VHF array element for AIS space application,” in *Antennas and Propagation Society International Symposium (APSURSI), 2013 IEEE*, July 2013, pp. 980–981.

Summary

After the development in the past 120 years since the invention of the first radio transmission, worldwide wireless communication systems are nowadays part of daily life. Behind the shining and astonishing achievement of modern communication systems, the exhaustion of existing frequency spectrum resources has been a concern. In higher frequency bands, the most advanced techniques are in development for the fifth cellular mobile communication system (5G) to meet rapid growth in its applications. The 5G system sits in the millimeter-wave band and consumes a wider bandwidth to offer a higher data transmission speed and a larger system capacity. Frequency/time/code division multiple accesses (FD-MA/TDMA/CDMA) are successful techniques to reuse and to save the frequency spectrum. However, in lower frequency bands, existing communication systems face similar unprecedented demands to accommodate more users in new applications. These growing demands exceed the designed system capacity and thus call for innovative solutions while keeping compatibility to the current setup to reduce the cost of users. For example, in the automatic identification system (AIS), satellite receivers are being used for expanding the service coverage of ship tracking to the global range, and similarly in the automatic dependent surveillance-broadcast (ADS-B) system for aircraft tracking. These systems are narrowband and originally designed in the last century, but they will continue to run for at least another couples of years without major updating of the user-side equipment.

The new application of AIS considered in this thesis is Satellite AIS. The satellite runs in the low-earth orbit (LEO). On the satellite, receiving AIS signals becomes much more difficult than before: one has to combat in-cell and inter-cell interfering sources from the system itself. Interference suppression is the main topic of this thesis. Narrowband spatial beamforming techniques for antenna arrays are candidate solutions to this challenge. This thesis tries to develop new beamforming techniques with a simple structure and a low computational complexity. With these techniques, this thesis establishes a framework of multiuser reception for Satellite AIS.

The new beamforming techniques are proposed through three consecutive chapters associated with their foundation, evolution, and application.

In Chapter 1, the background and the issues brought by Satellite AIS are introduced. Related literature is reviewed. The contribution of this thesis is shown.

In Chapter 2, the beamforming problem for signals in additive white noise is discussed. As a basic tool for the proposed algorithms in this thesis, a signed URV algorithm (SURV) is proposed for the basic problem of principal subspace computation and tracking as a replacement of the singular value decomposition (SVD). The updating and downdating of SURV is direct and simple. SURV has no issue of numerical stability unlike previous algorithms in linear algebraic and shows consistent performance in both stationary and nonstationary cases. This chapter shows how SURV is derived and provides its theoretical support.

In Chapter 3, the beamforming techniques for interference suppression in nonstationary scenarios are discussed. New blind beamforming techniques are proposed for separating overlapping packets in such scenarios. The connections between subspace intersection, oblique projection, the generalized SVD (GSVD), the generalized eigenvalue decomposition (GEVD), and SURV are exposed. SURV is used as one of the basic tools for the beamforming techniques. Simulation and experimental results of the proposed algorithms are shown.

In Chapter 4, based on the proposed algorithms in Chapter 3, a special blind beamforming technique enabling tracking for the multi-user receiver for Satellite AIS is proposed. The proposed algorithm is based on SURV. Results of the receiver in a software simulation model and on a hardware platform are provided.

In the remaining part of the thesis, the work on developing the software simulation model and constructing the hardware platform is presented. The outputs of the work are used for the verification and validation of the proposed algorithms in this thesis.

In Chapter 5, the method of developing the software testbed (simulation model) is presented. This testbed is built by using several tools including SystemC-AMS and MATLAB. The software implementation of the receiver is done in MATLAB and then translated into C++. This chapter first shows a lightweight version of the testbed in the hope that readers can learn and construct their own simulation model from scratch. The chapter also shows the possibility that the lightweight version will be extended and reconfigured to a more sophisticated model for the practical global ship distribution and satellite orbit from launched satellites like what is done in Chapter 4.

In Chapter 6, the structure of the hardware platform is presented. This chapter gives an example on how to build array receivers from available equipment. This platform uses an array of modified commercial RF frontends to downconvert the AIS signals to baseband. Sampled data are fed into PC and processed in MATLAB. The decoded AIS messages are analyzed and visualized on maps.

Samenvatting

Na een ontwikkeling van 120 jaar sinds de uitvinding van de eerste radio-transmissie zijn wereldwijde draadloze communicatiesystemen nu onderdeel van ons dagelijks leven. Na de verbazingwekkende verwezenlijking van moderne communicatiesystemen is de uitputting van het bestaande frequentiespectrum een punt van zorg geworden. Voor de hogere frequentiebanden worden geavanceerde technieken ontwikkeld om met vijfde-generatie (5G) mobiele communicatie nieuwe toepassingen mogelijk te maken. Het 5G systeem gebruikt de millimeter golflengte en heeft een bredere bandbreedte nodig om hogere transmissie-snelheden en systeem-capaciteit te kunnen bieden. Deling van het spectrum door middel van frequentie/tijd/code (FDMA/TDMA/CDMA) wordt nu succesvol toegepast in systemen. Echter, de lagere frequentiebanden zien een overeenkomstige groei in aantallen gebruikers. Deze groei overschrijdt de beschikbare capaciteit en innovaties zijn nodig terwijl de systemen compatibel moeten blijven met de huidige systemen om de kosten voor gebruikers te minimaliseren. Bijvoorbeeld, in het Automatische Identificatie Systeem (AIS) voor schepen worden nu satellieten geïntroduceerd om schepen wereldwijd te kunnen volgen, en soortgelijke uitbreidingen zijn er voor vliegtuigen (ADS-B). Deze systemen zijn smalbandig en origineel ontworpen in de vorige eeuw, maar zullen tenminste nog een aantal jaren gebruikt worden zonder aanpassingen aan de technologie aan de gebruikerskant.

De nieuwe toepassing voor AIS die in dit proefschrift wordt bestudeerd is Satelliet-AIS. De satelliet draait in een lage baan om de aarde. Voor een satelliet is het ontvangen van AIS signalen moeilijker dan op de grond: er zijn storende AIS signalen van zowel dezelfde communicatiecel als ook vanuit naburige cellen. Het onderdrukken van deze stoorsignalen is het hoofdthema van dit proefschrift. We kunnen gebruikmaken van smalbandige bundelvormings-technieken en collecties van antennes. In dit proefschrift ontwikkelen we nieuwe bundelvormings-technieken met een simpele structuur en een lage numerieke complexiteit. Hiermee ontwerpen we een kader voor een meer-gebruikers systeem voor Satelliet-AIS.

De nieuwe bundelvormings-technieken worden gepresenteerd in drie opeenvolgende hoofdstukken, overeenkomend met hun fundament, evolutie, en toepassing.

In Hoofdstuk 1 introduceren we de achtergrond en problematiek rond Satelliet-

AIS. De beschikbare literatuur wordt behandeld en de bijdrage die het proefschrift geeft wordt aangegeven.

In Hoofdstuk 2 wordt het bundelvormingsprobleem voor meerdere signalen in witte ruis besproken. Als basisgereedschap voor de voorgestelde algoritmes ontwikkelen we een zogenaamd Signed URV (SURV) algoritme, waarmee we de belangrijkste deelruimten van een matrix kunnen bepalen en volgen, ter vervanging van de singuliere-waarden ontbinding (SVD). Toevoegen en verwijderen van vectoren uit de ontbinding is direct en eenvoudig. De SURV is numeriek stabiel (in tegenstelling tot eerdere algoritmes), zowel in stationaire als in niet-stationaire gevallen. Het hoofdstuk geeft de afleiding en theoretische onderbouwing van de SURV.

In Hoofdstuk 3 worden bundelvormingstechnieken voor storingsonderdrukking in niet-stationaire situaties besproken. Nieuwe ‘blinde’ bundelvormingstechnieken voor het scheiden van half-overlappende pakketten worden voorgesteld. De verbanden tussen de doorsnedes van deelruimtes, scheve projecties, de gegeneraliseerde SVD (GSVD), de gegeneraliseerde eigenwaarden-ontbinding (GEVD) en de SURV worden onthuld. De SURV wordt vervolgens gebruikt als basisgereedschap voor bundelvorming. Simulaties en experimentele resultaten voor de voorgestelde algoritmes worden getoond.

In Hoofdstuk 4 vervolgen we Hoofdstuk 3 en gebruiken we de voorgestelde algoritmes om een volgsysteem te maken voor een meer-gebruikers Satelliet AIS systeem. De algoritmes zijn gebaseerd op SURV. We geven resultaten van de ontvanger voor simulaties in software en voor een experimenteel platform.

De rest van het proefschrift gaat in op de ontwikkeling van het software simulatiepakket en het experimentele platform. De resultaten hiervan worden gebruikt voor de verificatie en validatie van de in dit proefschrift voorgestelde algoritmes.

In Hoofdstuk 5 presenteren we een methode om een software testbed (simulatiemodel) te ontwikkelen. Dit testbed is gebaseerd op diverse middelen waaronder SystemC-AMS en MATLAB. De software implementatie van de ontvanger is gedaan in MATLAB en dan vertaald naar C++. Dit hoofdstuk toont eerst een kleine versie van dit testbed, met de hoop dat lezers hieruit kunnen leren hoe ze hun eigen simulatiemodel kunnen opbouwen. Het hoofdstuk toont ook hoe deze kleine versie kan worden uitgebouwd en ontwikkeld in een geavanceerder model voor de wereldwijde verdeling van schepen en de trajecten van gelanceerde satellieten, overeenkomstig Hoofdstuk 4.

In Hoofdstuk 6 presenteren we de opzet van het experimentele platform. Dit hoofdstuk is een voorbeeld hoe een meer-antenne systeem gebouwd kan worden uit beschikbare apparatuur. Het platform gebruikt meerdere commercieel verkrijgbare RF voor-ontvangers om de AIS signalen te verschuiven naar basisband. De bemonsterde signalen worden aan een computer gegeven en in MATLAB verder verwerkt. De gedecodeerde AIS berichten worden geanalyseerd en getoond op een kaart.

Acknowledgements

The first thank goes to my promotor and supervisor, Prof. Alle-Jan van der Veen for his great help and patience in encouraging me to pursue my PhD.. His wisdom guided me through the hard time of my research. He spent enormous time on weekly discussion with me. Without his effort, I would not have a complete thesis.

My second thank goes to the PhD committee members, for their time and comments for improving the quality of the thesis.

My third thank goes to the main financial source, the China Scholarship Council from China.

I want to especially thank my family for their spiritual supports.

Many thanks goes to the people of the CAS group in Delft University of Technology.

*Delft,
Dec. 2015*

Mu Zhou

Curriculum Vitae

Mu Zhou was born in China in 1982. He received the B.Sc. and M.Sc. degrees from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2005 and 2008, respectively, all in electrical engineering.

Since 2008, he has been working towards the Ph.D. degree in the Circuits and Systems Group in the Department of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, The Netherlands. His current research interests include matrix decomposition for array signal processing, receiver design for nano-satellite communication and mixed signal system modeling

