Adaptive Antenna Arrays for Precision GNSS Receivers

Dissertation

Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the
Graduate School of The Ohio State University

By

Andrew J. O’Brien, M.S.

Graduate Program in Electrical and Computer Engineering
The Ohio State University
2009

Dissertation Committee:

Prof. Prabhakar H. Pathak, Co-Adviser
Prof. Inder J. Gupta, Co-Adviser
Prof. Lee Potter
Prof. Joel Johnson
ABSTRACT

Antenna arrays with adaptive filters are currently used to provide interference suppression capabilities for Global Navigation Satellite System (GNSS) receivers. An adaptive array allows greater performance over a single element antenna by providing beamforming/null steering in the directions of satellites and interference sources. Unfortunately, there are some important limitations to the GNSS adaptive arrays in use today. For example, in the process of suppressing interference, adaptive antennas may inadvertently distort the GNSS signal and introduce bias errors into the receiver’s position and time estimates. Furthermore, many systems produce sub-optimal interference suppression performance, which degrades the accuracy of the navigation solution. To overcome these limitations, this dissertation develops novel adaptive antenna algorithms and techniques suitable for precision GNSS receivers. Three primary contributions are made. First, it develops an approach for optimal suppression of interference. For a GNSS application, this corresponds to an adaptive filter that maximizes carrier-to-noise ratio ($C/N_0$). The second contribution is the development of approaches for preventing the adaptive antenna array from introducing errors into the GNSS receiver measurements. These techniques take the form of a special adaptive filter algorithm and additional receiver logic that mathematically guarantee zero antenna-induced error even during interference suppression. Since
mitigation of these errors requires accurate antenna manifold information, a calibration procedure is needed to obtain the antenna manifolds in an efficient and practical manner. Consequently, the third contribution is a novel self-calibration algorithm. This algorithm simultaneously estimates the antenna manifold and navigation information “on-the-fly”. Collectively, these contributions advance the state-of-the-art in GNSS adaptive antennas in terms of performance, precision and practicality.
ACKNOWLEDGMENTS

I would like to thank all of my friends, colleagues, and professors at the Ohio State University. In particular, I am indebted to the ElectroScience Lab for providing a truly special and unique environment for me to grow. I am also very grateful for the guidance of my advisors, Prof. Gupta and Prof. Pathak. Thank you.

This work was sponsored in part by the GPS Wing Los Angeles AFB, AFRL Sensors Directorate, and the Air Force Office of Scientific Research. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsement, either expressed or implied, of the U.S. Air Force or the U.S. Government.
VITA

January 24, 1981 ......................... Born - Akron, OH

June 2004 ................................. B.S. Electrical & Computer Engineering, The Ohio State University, Columbus, Ohio, USA

June 2007 ................................. M.S. Electrical & Computer Engineering, The Ohio State University, Columbus, Ohio, USA

June 2004-present ....................... Graduate Research Associate, Electro-Science Lab, The Ohio State University

PUBLICATIONS

Research Publications


A. OBrien and I.J. Gupta, “Mitigation of Adaptive Antenna-Induced Biases in GNSS Receivers,” *IEEE Transactions on Aerospace and Electronic Systems*, accepted for publication.


FIELDS OF STUDY

Major Field: Electrical and Computer Engineering

Studies in:

- Electromagnetics
- Signal Processing
- Antennas and Antenna Arrays
- Adaptive Filters
- GPS/GNSS Receivers
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapters:</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background: Antenna Arrays for GNSS Receivers</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Limitations of Existing Systems</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Contributions and Organization of This Document</td>
<td>6</td>
</tr>
<tr>
<td>2. Antenna-Induced Biases in GNSS Receivers</td>
<td>11</td>
</tr>
<tr>
<td>2.1 Overview of Satellite Navigation Signals</td>
<td>12</td>
</tr>
<tr>
<td>2.2 GNSS Receiver Model</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Antenna-Induced Errors in GNSS Measurements</td>
<td>16</td>
</tr>
<tr>
<td>2.4 Calibration of GNSS Antennas</td>
<td>19</td>
</tr>
<tr>
<td>2.5 Summary &amp; Conclusions</td>
<td>21</td>
</tr>
<tr>
<td>3. GNSS Adaptive Antenna Arrays</td>
<td>22</td>
</tr>
<tr>
<td>3.1 Overview of the Interference Threat</td>
<td>23</td>
</tr>
<tr>
<td>3.2 GNSS Adaptive Antenna Arrays</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Adaptive Antenna Array Model</td>
<td>27</td>
</tr>
</tbody>
</table>
7. Self-Calibration Algorithm for GNSS Antennas .......................... 126
  7.1 Introduction ......................................................... 127
  7.2 Overview of the Proposed System .................................... 129
  7.3 Antenna Calibration Data ............................................. 133
  7.4 Measurement Model ................................................... 136
    7.4.1 GNSS Geometric Model ......................................... 136
    7.4.2 Linearization of Measurements .................................. 139
    7.4.3 Differential GNSS Measurement Model .......................... 143
  7.5 Isolating the Antenna-Induced Pseudorange Biases ................. 149
    7.5.1 Antenna-Induced Bias Residual ............................... 150
    7.5.2 Solving for the Antenna-Induced Pseudorange Bias .......... 152
    7.5.3 The Unknown Phase Center Offset .............................. 154
  7.6 Isolating the Antenna-Induced Carrier Phase Biases ............... 156
  7.7 Antenna Calibration Algorithm ...................................... 157
  7.8 Implementation ....................................................... 160
  7.9 Example Results ..................................................... 163
  7.10 Summary & Conclusions ............................................... 177

8. Self-Calibration Algorithm for GNSS Antenna Arrays ................... 178
  8.1 Introduction ......................................................... 179
  8.2 Overview of Proposed System ........................................ 180
  8.3 Antenna Array Manifold Model ...................................... 182
  8.4 Collecting Measurements ............................................. 183
  8.5 Solving for the Antenna Array Manifold ............................ 185
  8.6 Example Results ..................................................... 188
  8.7 Summary & Conclusions ............................................... 199

9. Summary & Future Work .................................................. 200

Bibliography .............................................................. 203
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>GNSS receiver model with antenna and front-end electronics.</td>
<td>17</td>
</tr>
<tr>
<td>3.1</td>
<td>Example GNSS adaptive antennas.</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>Model of an antenna array with a space-time adaptive filter.</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>6-element circular array of patch antennas.</td>
<td>43</td>
</tr>
<tr>
<td>3.4</td>
<td>SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the absence of interference.</td>
<td>47</td>
</tr>
<tr>
<td>3.5</td>
<td>SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the presence of 3 wideband interferers.</td>
<td>48</td>
</tr>
<tr>
<td>3.6</td>
<td>SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the presence of 3 offset, partial-band interferers.</td>
<td>49</td>
</tr>
<tr>
<td>4.1</td>
<td>The uniform linear array of 7 isotropic elements used for simulations in this study.</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Pre-Correlation SINR (left) and post-correlation SINR (right) vs. STAP filter length using the Max SINR and Max C/N algorithms.</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>Adaptive antenna system response magnitude in SOI direction for different STAP filter lengths using the Max SINR algorithm (left) and maximum C/N₀ algorithm (right). Note that, in the absence of interference, all of the responses on the right are the same regardless of filter length.</td>
<td>70</td>
</tr>
<tr>
<td>4.4</td>
<td>Pre-Correlation SINR (left) and post-correlation SINR (right) vs. System Bandwidth using the Max SINR and Max C/N algorithms.</td>
<td>71</td>
</tr>
</tbody>
</table>
4.5 Adaptive antenna system response magnitude in SOI direction for different system bandwidths using the Max SINR algorithm (left) and the maximum $C/N_0$ algorithm (right). Note that, in the absence of interference, all of the responses on the right are the same regardless of system bandwidth. .......................................................... 71

4.6 Pre-correlation SINR performance of different STAP algorithms in the presence of an interfering signal. .......................................................... 75

4.7 Receiver post-correlation SINR performance of different STAP algorithms in the presence of an interfering signal. DC, MMSE and Max C/N methods have near identical performance. ................................. 75

4.8 Processing gain of different STAP algorithms. ........................................ 76

4.9 Normalized power spectral density of the noise at the output of the adaptive antenna for different STAP algorithms when the SOI is incident from $(\theta, \phi) = (0, 0)$. ................................................................. 76

4.10 Pre-correlation SINR performance of different STAP algorithms in the presence of 7 interfering signals. MMSE, Max SINR, and Max C/N methods have very similar performance. ............................ 77

4.11 Receiver post-correlation SINR performance of different STAP algorithms in the presence of 7 interfering signals. DC and Max C/N methods have very similar performance. ............................ 77

5.1 $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle in the absence of interference. ................................................................. 96

5.2 $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle for an interference scenario containing 3 partial-band interferers. ................. 97

5.3 $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle for an interference scenario containing 3 partial-band and 3 wideband interferers. ................................................................. 98
5.4 Distribution of $C/N_0$ in terms of upper-hemispherical coverage for multiple scenarios with 3 partial-band interferers (top), scenarios with 3 wideband interferers (middle), and scenarios containing 3 partial-band interferers and 3 wideband interferers (bottom). ........................................ 99

5.5 Distribution of $C/N_0$ in terms of upper-hemispherical coverage over multiple scenarios containing 3 partial-band and 3 wideband interferers with adaptive filter lengths of 3 taps (top), 5 taps (middle), and 9 taps (bottom). .......................................................... 100

6.1 The general, simplified GNSS receiver tracking loop model used in this study showing the delay lock loop (DLL) and carrier phase lock loop (PLL). ................................................................. 103

6.2 Three GPS signal types and their respective power spectra (top), autocorrelation functions (middle), and S-curves (bottom) for the particular discriminators. ..................................................... 108

6.3 S-curves for the C/A-code (top), P-code (middle), and M-code (bottom) tracking loops showing the effects of the antenna in the presence and absence of interference using the simple null steering algorithm. Also shown is the ideal S-curve which neglects antenna effects. .... 112

6.4 Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle using the simple null steering algorithm. .................... 113

6.5 The proposed bias estimation and correction mechanism for a GNSS receiver equipped with an interference suppressing adaptive antenna array . ................................................................. 114

6.6 An example of an S-curve, distorted by an antenna, showing the two bias correction terms. ............................................................. 116

6.7 Receiver code phase tracking loop model. The tracking loop bias correction term, $\tau_1$, and the navigation processor bias correction term, $\tau_2$, are both shown; however, only one type of correction would need to be applied in a specific implementation. ...................... 117
6.8 Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle. The adaptive algorithm is the simple power minimization method.

6.9 Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle. The adaptive algorithm is the single directional constraint method.

7.1 Integration of the proposed self-calibration logic with a standard receiver system.

7.2 Geometry of receiver positions and satellites.

7.3 Geometry of a differential position between two points in space.

7.4 Antenna-induced bias where $j$ is the reference point location, $i$ is the initial position/time estimate, and $\xi$ is the biased position solution.

7.5 Simulated crossed-slot antenna on aircraft model.

7.6 Code delay bias (top), carrier phase bias (middle) and amplitude (bottom) for the antenna on a groundplane (left) and on an aircraft (right).

7.7 True position (top) and orientation (bottom) of the simulated user receiver.

7.8 Sky plots of the GPS satellite angle-of-arrivals – accounting for user position and orientation – from left to right and top to bottom: 5 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, and 4 hours.

7.9 The total number of measurements versus time (top), the percent upper hemisphere coverage of those measurements (bottom).

7.10 Code delay calibration after 4 hours. The original bias (a), the angular regions that don’t have measurements (b), the estimated phase (c) and residual error (d), and the estimated phase with phase center offset correction (e) and residual error with phase center offset correction (f).
7.11 Code delay calibration after 4 hours for an elevation cut ($\phi = 45^\circ$). The original bias (top left), the angular regions that don’t have measurements (top right), the estimated phase and residual error (middle), and the estimated phase and residual error with phase center offset correction (bottom). .................................................. 173

7.12 Carrier phase calibration after 4 hours. The original bias (a), the angular regions that don’t have measurements (b), the estimated phase (c) and residual error (d), and the estimated phase with phase center offset correction (e) and residual error with phase center offset correction (f). 174

7.13 Carrier phase calibration after 4 hours for an elevation cut ($\phi = 75^\circ$). The original bias (top left), the angular regions that don’t have measurements (top right), the estimated phase and residual error (middle), and the estimated phase and residual error with phase center offset correction (bottom). .................................................. 175

7.14 Mean absolute error after calibration for the code delay (top) and carrier phase (bottom) for three cases: no noise, standard noise levels at 10 Hz (Noise 1) measurement rate, and standard noise levels at 1 Hz (Noise 2). .................................................. 176

8.1 An extension of the self-calibration logic proposed in Chapter 7 in order to make it applicable to adaptive antenna arrays. .................. 181

8.2 Simulated 7-element circular array of crossed-slot antennas on an aircraft. .................................................. 189

8.3 Original antenna manifold in the ($\theta, \phi$) = (60, 0) direction (left) and estimated antenna manifold (right). .................. 192

8.4 Original antenna manifold in the ($\theta, \phi$) = (40, 100) direction (left) and estimated antenna manifold (right). .................. 192

8.5 Elevation cut in the $\phi = 30^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1). .................. 193

8.6 Elevation cut in the $\phi = 30^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2). .................. 194
8.7 Elevation cut in the \( \phi = 50^\circ \) plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1). ......................................................... 195

8.8 Elevation cut in the \( \phi = 50^\circ \) plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2). ......................................................... 196

8.9 Elevation cut in the \( \phi = 120^\circ \) plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1). ......................................................... 197

8.10 Elevation cut in the \( \phi = 120^\circ \) plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2). ......................................................... 198
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Interference Properties</td>
<td>44</td>
</tr>
<tr>
<td>5.1 Adaptive Filter Algorithms</td>
<td>92</td>
</tr>
</tbody>
</table>
CHAPTER 1

Introduction

1.1 Background: Antenna Arrays for GNSS Receivers

In recent years, satellite navigation has revolutionized positioning and timing. Although the term “satellite navigation” was once synonymous with the Global Positioning System (GPS), the proposal of new constellations by other nations has necessitated creation of a new term: Global Navigation Satellite Systems (GNSS). GNSS can be used to refer to the United States’ GPS, the European Union’s Galileo, Russia’s GLONASS, China’s Compass, and a variety of others, each of which are currently at various stages of development or upgrade [1]. These satellite systems are fundamentally similar, and one can typically develop technology that applies generally to all of them. The research presented in this work, which focuses on the receiver antenna and its associated signal processing, is – generally speaking – applicable to any GNSS system so long as the antenna and antenna electronics are designed for the required frequencies. In this work, we will frequently use the term “GNSS antenna” to refer to an antenna designed for any one or multiple satellite navigation systems.

To those outside the navigation community, it often comes as a surprise that there is still significant, on-going research into GNSS receiver systems. This research
is largely focused on increasing the precision, integrity and robustness of navigation systems which use GNSS. Large sponsors of such research are military and aviation organizations who are eager to utilize satellite navigation but have strict requirements on both accuracy and reliability. Arguably, one of the most important factors driving research is fear of encountering GNSS-denied environments. In particular, the weak power of received GNSS satellite signals makes them vulnerable to radio frequency interference (RFI) [2–4]. Interference can originate from a variety of sources. For example, a GNSS receiver could be mounted in an environment with other electronics that unintentionally produce RFI [5,6]. In a more difficult case, the interference could come from hostile jamming [7, 8]. It is difficult to overemphasize the importance of navigation capabilities to military and aviation users, and great effort and expense has been directed toward overcoming interference environments. Fundamentally, system designers are faced with two options. The first is to pursue so-called “alternative navigation” technology in order to supplement and provide failover in GNSS-denied environments. The second option – which this research addresses – is to design interference suppression systems for these receivers.

As it stands today, adaptive antenna arrays are the only feasible candidate for allowing reception of GNSS signals in harsh and sustained interference environments. Although a variety of interference suppression approaches have been proposed for single-element receivers, they are often only applicable to a narrow class of interference types [9–12]. In general, antenna arrays offer greater performance over these systems by providing spatial degrees of freedom with which to separate the desired GNSS signals from undesired interference. Typically, the patterns of these antenna arrays are controlled via adaptive digital filtering which perform beamforming/null
steering in specific directions. An adaptive finite impulse response (FIR) filter is placed behind each antenna element which adds a temporal filtering aspect to the spatial filtering, and the result is commonly referred to as a space-time adaptive processor (STAP) [13]. If each filter only has a single weight, it is referred to as a space-only processor (SOP). In either case, the digital filter weights are chosen to destructively cancel interference while constructively preserving the navigation signals. The filter is adaptive (its weights are determined at run time based on received signals) so that it does not require a priori information or assumptions about the interference environment. STAP-based systems offer superior performance over SOP-based ones in scenarios that contain wideband interferers and, consequently, is the preferred choice for modern adaptive antenna arrays [14–16].

Historically, STAP-based adaptive antennas have been widely used for radar and communications applications [13, 17–19]. The design of adaptive filtering algorithms for these systems has been primarily motivated by the need to provide adequate signal-to-interference-plus-noise ratio (SINR) at the output of the adaptive antenna. When demand arose for interference suppression in GPS applications, existing adaptive filter technology was applied to the new applications with little consideration for the subtleties of satellite navigation [20, 21]. While current STAP systems provide interference suppression that is adequate for many GNSS receivers, they do not produce the ideal or optimal behavior. Consequently, the adaptive antenna arrays in use today have some important shortcomings when applied to the most demanding navigation applications. The next section will go into detail regarding the limitations of existing systems and the failure of previous research to adequately address these issues. Before we go any further, it should be noted that this work is focused on
the signal processing aspects of adaptive antenna arrays. Although the design of the physical antenna array is an important aspect, it will not be explored. It is intended that this work be applicable to any choice of antenna array.

1.2 Limitations of Existing Systems

There are three predominant limitations of GNSS antenna arrays in use today. First, their adaptive filters have not been explicitly designed to maximize carrier-to-noise ratio ($C/N_0$), the primary performance metric for GNSS receivers. Secondly, these antenna systems can introduce significant bias errors into the GNSS receiver measurements [22–24]. This is especially true during interference suppression, when there is greater potential for the adaptive filters to distort the GNSS signals. The amount of bias depends on many factors including the adaptive algorithm, the physical antenna array and front-end antenna electronics; however, code phase errors on the order of meters and carrier phase biases of hundreds of degrees are not uncommon. This is unacceptable for precision applications [25, 26]. Thirdly, traditional antenna calibration procedures cannot be used with adaptive antenna arrays. In order to obtain sufficient information about the antenna array manifold, there still exists a need for a practical calibration technique.

Proposals made by other research groups have failed to adequately address these limitations. Rather, upon closer inspection, one finds that their solutions are impractical or incomplete. For example, some believe that antenna-induced biases should be prevented by careful design of the physical antenna array. In [27], the author concludes that the adaptive antenna can only produce small biases if isotropic elements are used; for non-isotropic antennas, the user must select whether they want
interference suppression or precision navigation. It is claimed that antenna designers should design the elements of their antenna arrays to be as isotropic as possible. This requirement is clearly impractical since real antenna arrays will never satisfy that condition. More importantly, bias mitigation methods should never depend on the antenna array design. Not only should array design be motivated by more important factors (size and cost, for instance), but there is also the fact that mounting the array on a complex platform, such as an aircraft, will change its effective pattern.

Other authors suggest adding significant amounts of additional filtering. For example, in [28] the author proposes that the signal from each antenna element be fed into a filter bank. This filter would apply the inverse of the antenna response, and effectively make each response that of an isotropic element. Since anywhere between 6 - 12 GNSS signals are received at once, one would require a corresponding number of filters per element for this to work. After all of this pre-processing, the output signals would then be fed into the actual adaptive filter for interference suppression. It should be quite obvious that this approach has significant computational requirements. In a different work, [29] suggests adding post-processing filters. In this case, the idea is that a post-processing filter placed at the output of the adaptive antenna could compensate for all of the distortion introduced by the antenna, antenna electronics and adaptive filter. This would require less filters than the above approach; however, the computational complexity of these filters would be very significant. They would have to be updated at the same rate as the adaptive filters, and would incur a complexity on the same order as the adaptive filters themselves.

Some authors choose to implement special adaptive filtering algorithms which are designed to prevent antenna-induced errors; however, many of these approaches suffer
drawbacks. For instance, the algorithm described in [30] attempts to prevent distortion of the satellite signal by imposing a significant number of constraints across the frequency bandwidth. These constraints limit the degrees of freedom for interference suppression and increase the computational requirements. In a different work, [31] designs an algorithm that minimizes biases produced by the adaptive filter, but does not account for the errors introduced by the physical antenna array and antenna electronics. Thus, it represents only a partial solution. Others, such as [14], initially concluded that adaptive antenna-induced biases are small and will not affect precision GNSS measurements; however, today, this claim is understood to be incorrect.

1.3 Contributions and Organization of This Document

If adaptive antennas are to be developed for precision GNSS receivers, there are certain goals that must be achieved. First these systems must be designed to yield optimal $C/N_0$ so as to minimize noise errors. Secondly, they must be designed to reduce antenna-induced biases down to a few centimeters in code phase and a few degrees in carrier phase. The bias should be mitigated without degrading the interference suppression performance or placing any requirements on the physical antenna array design. Third, these systems must require only a simple antenna calibration procedure, or, ideally, the system could self-calibrate using only GNSS measurements. Finally, these requirements should be met with practical proposals that are efficient and implementable. To this end, any proposed solutions should be incremental to existing systems in order to hasten industry adoption. The research presented in this dissertation aims to achieve all of these goals.
The next two chapters provide the requisite introductory material. Chapter 2 begins with an introduction to single-element GNSS antennas. It reviews the structure of GNSS signals and discusses how the antenna affects GNSS receiver measurements. Next, Chapter 3 extends the discussion to GNSS adaptive antenna arrays. A detailed mathematical model for adaptive antenna arrays is provided which is the basis for all later chapters. Example simulation results are shown which demonstrate the performance of these systems in interference environments and highlight their limitations.

Chapter 4 takes the first step towards improving these systems by deriving a novel adaptive filtering algorithm which achieves the optimal interference suppression performance for a GNSS receiver. The chapter begins by reviewing how GNSS receiver performance is measured in terms of the receiver $C/N_0$. The mathematical and qualitative distinction is made between $C/N_0$ and the SINR at the adaptive antenna output, and it is demonstrated that SINR is an inadequate indication of GNSS receiver $C/N_0$ performance. Consequently, an adaptive filtering algorithm is derived which maximizes $C/N_0$. The performance of common STAP algorithms is compared in the context of this performance bound.

Chapter 5 discusses the problem of the adaptive antenna distorting received GNSS signals during interference suppression. This distortion introduces bias errors into the GNSS receiver’s measurements of code delay and carrier phase. For precision GNSS applications, these antenna-induced biases are a significant source of error. In Chapter 5, the results of Chapter 4 are extended to derive a novel adaptive algorithm that simultaneously optimizes $C/N_0$ while preventing the adaptive antenna from introducing errors into the satellite signals. These qualities represent the optimal behavior for an adaptive antenna in GNSS applications. It is the only algorithm which is able to
mathematically guarantee zero bias and simultaneously achieve optimal $C/N_0$ performance over all interference scenarios. Most importantly, the proposed algorithm has implementation requirements comparable to algorithms in use today.

Chapter 6 also focuses on the problem of preventing adaptive antenna-induced biases. Since system designers have different circumstances, it might be the case that implementing a special adaptive filter algorithm – as proposed in Chapter 5 – is not practical. To provide an alternative, this chapter develops a novel bias estimation and correction technique in which additional logic is added to the receiver in order to provide run time compensation for antenna-induced biases. The technique is general in the sense that it can be applied to a wide variety of adaptive antenna and receiver implementations. It utilizes a computationally efficient bias estimation equation, and the receiver can compensate for the bias errors either in the navigation processor or in the tracking loop of the GNSS receiver.

Thus far, Chapters 4, 5 and 6 have developed novel techniques and algorithms for GNSS adaptive antennas which optimally suppress interference without antenna-induced errors in the satellite signal. While this represents the ideal behavior, these techniques have been developed under the assumption that the receiver has perfect knowledge of the antenna array manifold. Unfortunately, obtaining a precise antenna calibration using existing techniques is a task that is commonly tedious, inconvenient or impractical. In order to address this issue, the next two chapters develop techniques for obtaining precise antenna calibration information.

Chapter 7 discusses the feasibility of extending a GNSS receiver to perform self-calibration: simultaneous estimation of the navigation parameters and antenna calibration information. This self-calibration can be performed on-the-fly (i.e. during
navigation) instead of being restricted to special calibration environments and procedures. This is accomplished using a differential GNSS configuration with a short baseline. As the user receiver moves, the antenna-induced biases of the proposed system will steadily converge to zero as the antenna is exposed to sufficient diversity in the angles of arriving satellite signals. In order to simplify the introduction of this self-calibration algorithm, this chapter will assume that the antenna is a simple, single-element antenna.

Finally, Chapter 8 studies how this self-calibration approach can be used to improve a recently proposed technique [32] for calibrating adaptive antenna arrays. Calibration of a GNSS adaptive antenna array requires determination of the entire antenna array manifold over the frequencies of interest. Interestingly, the calibration technique proposed in [32] accomplishes array calibration utilizing only standard, differential GNSS receiver measurements. However, it also requires that the differential position of the antenna be known very precisely. This drawback can be overcome by applying the self-calibration approach described in Chapter 7. The implementation of this combined approach is described, and it is shown that it is both computationally efficient and easily integrated with existing receivers. Simulations are performed using a GPS antenna array with space-time adaptive processing (STAP) mounted on a moving aircraft. After application of the proposed self-calibration approach, simulations demonstrate that these biases gradually converge given sufficient time. Since the antenna array is calibrated in-situ, both platform effects and mutual coupling between array elements will be incorporated automatically.
Ultimately, this research advances the state of the art in GNSS adaptive antenna arrays. This work allows GNSS adaptive antenna arrays to provide interference suppression capabilities to even the most precise and demanding GNSS applications. Without these contributions, one must choose either robust interference suppression or precision navigation – not both. A summary and conclusion is provided in Chapter 9 along with a brief discussion of possible future work.
CHAPTER 2

Antenna-Induced Biases in GNSS Receivers

As the precision of GNSS receivers has improved, the effects of the antenna on the received satellite signals have become a non-negligible source of error [33]. This is especially true with GNSS adaptive antenna arrays, which tend to be large and involve complicated filtering. Even when the antenna is a conventional, single-element GNSS antenna, the antenna-induced errors can be significant [?]. This chapter discusses how the GNSS antenna affects the receiver measurements. Section 2.1 begins with a brief introduction to GNSS systems: their satellites, signals, and spectra. Only the aspects of satellite navigation that are necessary to understand the antenna will be introduced. Section 2.2 introduces a simple GNSS receiver model and describes its two primary measurements: each satellite signal’s code delay and carrier phase. Section 2.3 provides an overview of GNSS antennas. Direction-dependent delay and phase shifts caused by the antenna introduce errors into the receiver’s measurements. Expressions for these antenna-induced biases are derived. For precision GNSS applications, the antenna is often calibrated to reduce these errors. Section 2.4 reviews common GNSS antenna calibration procedures. The scope of this chapter will be limited to a single-element antenna while the next chapter will introduce GNSS antenna arrays.
2.1 Overview of Satellite Navigation Signals

A full GNSS constellation, such as GPS, is composed of 24-30 satellites. They are distributed over a number of orbital planes so that 8-12 are visible at almost any location on Earth. The satellite locations are precisely tracked by a network of ground stations. These ground stations upload precise orbital information to the satellites, and the satellites then broadcast this information to user receivers so that the position of each satellite can be precisely calculated. The range between the satellite to a user receiver is approximately 20,000 km and is determined by estimating the relative propagation time of known ranging signals sent synchronously by all satellites. While it is only necessary to track 4 satellites to determine the receiver’s position and time, modern receivers are able to track all satellites in view. It is desirable to track satellites at a variety of elevations and azimuths in order to improve geometric diversity, and, for best performance, GNSS antennas are expected to provide adequate coverage over the entire sky.

Each GNSS satellite broadcasts multiple navigation signals on each of multiple carrier frequencies. Some signals are designated for civilian use while others are encrypted military signals with special features. Receiving signals on multiple frequencies allow the receiver to cancel certain propagation errors and to combine measurements in other advantageous ways. In general, each ranging signal has three components: a carrier, a spread-spectrum pseudorandom code, and a navigation message. The navigation message is a low bit-rate signal which disseminates the satellite location, the signal transmit time, and other information to the GNSS receiver. On the other hand, the spreading code is a high bit-rate, pseudorandom code that is known to the receiver. For example, in current GPS signals, a binary spread-spectrum code
and a binary data message are combined and modulated onto the carrier using binary phase shift keying (BPSK). This results in a sinc-squared power spectra whose bandwidth depends on the chipping rate of the spreading code. Other GNSS systems use a variety of other modulation schemes with different spectra and behaviors; however, the concepts are similar. Typically, the satellites of a particular GNSS constellation all transmit on the same set of frequencies. For example, modern GPS satellites all transmit signals at 1.575, 1.227, and 1.176 GHz, each occupying a bandwidth of 24 MHz. Each satellite uses a different, orthogonal pseudorandom code so that the receiver can distinguish measurements from different satellites, and each signal will have minimal interference with the others.

Fundamentally, the GNSS receiver makes three primary measurements of each signal: the phase of the carrier, the Doppler shift of the carrier, and the code delay of the navigation signal. These three measurements are tracked and combined in a variety of ways to ultimately calculate a position and time solution. Since the way in which position and time can vary significantly between different receiver implementations, it is not beneficial to quantify the antenna effects on these solutions. Rather, for our purposes, it will be sufficient to consider the effects of the antenna on the carrier phase and code delay measurements, which have a more universal meaning among GNSS receiver systems.

2.2 GNSS Receiver Model

In the first processing stages of a GNSS receiver, the received signal is cross-correlated with a locally generated reference signal in order to estimate the code delay of the signal and the phase of the carrier relative to the receiver’s local clock [1,34].
Consider a single GNSS signal that enters the receiver after reception, down-conversion, and analog-to-digital conversion. This complex digital signal is denoted $y[n]$ and contains two components: a desired satellite signal component $y_d[n]$ and a noise component $y_n[n]$. This signal is written as

$$y[n] = y_d[n] + y_n[n], \quad (2.1)$$

where

$$y_d[n] = \sqrt{C_d} d(nT_0 - \tau_r) e^{-j\psi}. \quad (2.2)$$

Here, $C_d$ is the power of the incident signal, $d(t)$ is the pseudorandom spreading code, $T_0$ is the sampling period, $\tau_r$ is the propagation delay between the satellite and the receiver, and $\psi$ is the carrier phase shift over the same distance. The spreading code $d(t)$ is often assumed to be real and unit power. The noise component $y_n[n]$ will be modeled as white, zero-mean Gaussian. We will not consider propagation effects, Doppler effects, data bits, or other complications since these aspects are independent of antenna-related effects. Note that $n$ will be used to denote both the discrete time index and the noise component. It should be obvious which meaning is intended based on the context.

In the receiver, a reference signal $r(t)$ is generated which is an exact replica of the transmitted spreading code

$$r(t) = d(t - \hat{\tau}_r), \quad (2.3)$$

where $\hat{\tau}_r$ is the receiver’s initial estimate of the received signal delay. The receiver also performs carrier wipe-off of the received signal, and the phase of the local oscillator will be denoted $\hat{\psi}$. The receiver performs a finite cross-correlation between the received signal $y[n]$ and the reference signal $r(t)$ over $N$ samples to produce a
cross-correlation estimate

\[ \hat{R}_{yr}(\tau) = \frac{1}{N} \sum_{n=1}^{N} y[n]r(nT_0 + \tau) \quad (2.4) \]

\[ = \frac{1}{N} \sum_{n=1}^{N} y[n]d(nT_0 + \tau - \hat{\tau}_r). \quad (2.5) \]

The mean of (2.5)

\[ R_{yr}(\tau) = E\{\hat{R}_{yr}(\tau)\} \quad (2.6) \]

contains only the desired signal component and is given analytically as

\[ R_{yr}(\tau) = E\{y[n]r(nT - \tau)\} \quad (2.7) \]

\[ = \sqrt{C_d} E\left\{d(nT - \tau_r)d(nT - \hat{\tau}_r - \tau)e^{j(\psi - \hat{\psi})}\right\} \quad (2.8) \]

\[ = \sqrt{C_d} R_{dd}(\tau - \tau_r + \hat{\tau}_r)e^{j(\psi - \hat{\psi})}, \quad (2.9) \]

where \( R_{dd}(\tau) \) is the auto-correlation function of the spreading code

\[ R_{dd}(\tau) = \int G_d(f)e^{j2\pi f\tau} df. \quad (2.10) \]

At any instant, the receiver is interested in estimating the relative delay \( \tau_0 \) between the incident and reference signals and the relative phase \( \psi_0 \) between the incident signal and local oscillator. These are, respectively,

\[ \tau_0 = \tau_r - \hat{\tau}_r \quad (2.11) \]

\[ \psi_0 = \psi - \hat{\psi}, \quad (2.12) \]

and can be estimated from the cross-correlation in (2.9) using a well-known approach [35]

\[ R_{yr}(\tau) = \sqrt{C_d} R_{dd}(\tau - \tau_0)e^{j\psi_0}. \quad (2.13) \]
Since $R_{dd}(\tau)$ has a peak at $\tau = 0$, the new peak location is the delay $\tau_0$ which the receiver is interested in estimating

$$\tau_0 = \arg\max_\tau |R_{yr}(\tau)| .$$  \hspace{1cm} (2.14)

The receiver will adjust the delay of the reference signal until the relative delay between it and the incident signal is $\tau_0$. However, the carrier phase is adjusted independently. The desired carrier phase $\psi_0$ corresponds to the measured carrier phase when the signal is actually aligned (that is, at $\tau = 0$), which is given by

$$\psi_0 = \angle R_{yr}(0).$$ \hspace{1cm} (2.15)

Thus far, a single GNSS signal has been considered in isolation, which will be referred to as the signal-of-interest (SOI). It is implied that the receiver would track each of the available satellite signals simultaneously.

### 2.3 Antenna-Induced Errors in GNSS Measurements

The GNSS antenna model used in this study is depicted in Fig. 2.1. The antenna response $A(f, \theta, \phi)$ completely characterizes the antenna on a specific platform for a single polarization (which is implied to be RHCP). This response is relative to a specific point known as the phase reference point or antenna reference point, which corresponds to a physical location easily visible on the exterior of the antenna. When the antenna effects are included in the GNSS model, the position estimated by the receiver will correspond to this point. Directly attached to the antenna is the front-end hardware, which down converts the signal to baseband and performs analog-to-digital conversion (ADC). The RF channel typically begins with a low-noise amplifier (LNA), which is the origin of the dominant thermal noise in the received signal. $F(f)$
is used to denote the response of the front-end electronics. Note that multiple GNSS signals are simultaneously received from multiple frequency bands, and the front-end is commonly segmented so that each band has a separate output. For convenience, we will consider the digital output of each front-end channel as complex even though it is typically implemented as two separate I and Q channels.

To measure the antenna-induced biases, it is necessary to consider the effects of the antenna on the receiver cross-correlation. Without loss of generality, let us assume that the incident and reference signals are synchronized. That is, \( \tau_r = 0 \) and \( \psi = 0 \) in (2.13). In this case, the reference signal corresponds to the GNSS signal received by an isotropic antenna located at the phase reference point of the antenna pattern \( A(f, \theta, \phi) \). Any code or carrier phase measured from the cross-correlation corresponds solely to antenna-induced biases. The responses of the antenna \( A(f, \theta, \phi) \) and front-end antenna electronics \( F(f) \) can be combined into a single direction-dependent response \( H(f, \theta, \phi) \),

\[
H(f, \theta, \phi) = A(f, \theta, \phi)F(f) \ . \tag{2.16}
\]

\[\text{Figure 2.1: GNSS receiver model with antenna and front-end electronics.}\]
Since the incident signal is filtered by $H(f, \theta, \phi)$, the cross-correlation will be as well

$$R_{yd}(\tau, \theta, \phi) = \sqrt{C_d} h(t, \theta, \phi) * R_{dd}(\tau),$$  \hspace{1cm} (2.17)

or equivalently in the frequency domain,

$$R_{yd}(\tau, \theta, \phi) = \sqrt{C_d} \int H(f, \theta, \phi) G_d(f) e^{-j2\pi f \tau} df.$$  \hspace{1cm} (2.18)

Here, we have denoted it $R_{yd}$ instead of $R_{yr}$ since the reference signal is now an exact replica of the incident signal. The filtering of the cross-correlation in (2.18) can result in distortion from the ideal auto-correlation expected by the receiver. This results in biased estimates of the GNSS signal’s carrier phase and code delay. If there was no antenna-induced bias, then (2.18) would have a peak at $\tau = 0$. However, the adaptive filter weights can distort and delay the SOI and cause the peak to be shifted. The new peak location is the antenna-induced code delay bias $\tau_0$ and is given by

$$\tau_0(\theta, \phi) = \arg\max_\tau |R_{yd}(\tau, \theta, \phi)|.$$  \hspace{1cm} (2.19)

The receiver will adjust the delay of the reference signal until the relative delay between it and the incident signal is $\tau_0$. However, the carrier phase is adjusted independently. The antenna-induced carrier phase bias $\psi_0$ corresponds to the measured carrier phase when the signal is actually aligned (that is, at $\tau = 0$), which is given by

$$\psi_0(\theta, \phi) = \angle R_{yd}(0, \theta, \phi).$$  \hspace{1cm} (2.20)

The antenna also affects the relative amplitude $g_0$ of the received signal, which is given by

$$g_0(\theta, \phi) = \frac{1}{\sqrt{C_d}} |R_{yd}(\tau_0, \theta, \phi)|.$$  \hspace{1cm} (2.21)
These antenna-induced biases $\tau_0(\theta, \phi)$, $\psi_0(\theta, \phi)$ and $g_0(\theta, \phi)$ only apply to a single GNSS signal type on a specific band.

As mentioned earlier, $A(f, \theta, \phi)$ represents the in-situ response of the antenna relative to some known phase reference point. This reference point is chosen to correspond to a known physical location on the antenna. As a result, the cross-correlation function given in (2.18) is equivalent to the cross-correlation between a signal processed by the antenna response and the same signal received by an isotropic receiver located at the reference point. This seemingly simple fact has two very important consequences. First, one can measure the antenna-induced biases by using the cross-correlation function given by equation (2.18). Secondly, these antenna-induced biases are understood to be relative to this antenna reference point.

### 2.4 Calibration of GNSS Antennas

GNSS antennas used in precision applications are commonly calibrated in order to mitigate antenna-induced errors. The effect of a single-element GNSS antenna on a specific signal is completely characterized by the antenna-induced code delay bias (2.19), carrier phase bias (2.20), and amplitude (2.21) relative to some known antenna reference point. The complete characterization of the antenna would require a set of functions for each GNSS signal type on each frequency band. Acquisition of these functions constitutes calibration of the GNSS antenna. In the past, the GNSS community has used a variety of different antenna models for calibration purposes, and all can be understood as approximations of these functions. The most basic consider only the carrier phase bias and represent the angular variation as a simple phase center offset model [36,37]. Some assume the phase center varies with elevation.
angle only [38–40], while others consider the phase variation over all angles, but use a small set of polynomials or spherical harmonics as the basis [41–45]. Some consider both code and carrier phase errors, but only a few are concerned with determining the effect of the antenna on the C/N0 [46]. The choice of antenna model will depend on the antenna and the requirements of the particular application.

Currently, there are a variety of antenna calibration procedures, each with different levels of convenience, cost, and requirements. These procedures can be divided into two general categories: chamber calibration and field calibration. In the first case, an antenna pattern measurement is performed in a special measurement chamber [33,43]. The gain and phase variations are measured over the entire GNSS frequency bandwidth as the antenna is rotated and tilted. The antenna pattern is then processed in order to determine the code and carrier biases. The chamber provides a clean, isolated environment with the potential for precise measurements. Unfortunately, it also requires access to expensive equipment and facilities, which makes it unavailable to many system designers. Furthermore, the calibration is often performed with the antenna on a ground plane, so the resulting data is unable to account for platform effects.

Alternatively, a field calibration can be performed using live GNSS signals [47–49]. In this case, the only equipment required is a differential GNSS receiver system with a reference antenna and an antenna under test. Both antennas are setup outdoors, in an environment away from potential obstructions, and separated by a short baseline (several meters). If the relative position between the reference antenna and the antenna under test is known, then standard GNSS receiver measurements can be
post-processed in order to determine the antenna-induced biases. Often, this measurement process is performed over a long period of time so that the satellites change their position in the sky and eventually provide measurements over all angles. In some cases, the antenna is mounted on its intended platform (i.e. aircraft or vehicle), which allows the calibration to account for platform effects. Note that field calibrations often yield calibration data that is relative to reference receiver antenna. However, absolute field calibrations have been demonstrated by precisely moving the antenna during calibration and post-processing the data in order to remove the effects of the reference antenna, to remove local multipath effects, and to speed the calibration process [43, 44]. Field calibration techniques can be highly accurate and have been widely used to calibrate geodetic antennas for precision GNSS applications [38, 39].

2.5 Summary & Conclusions

This chapter provided a brief review of antenna effects on GNSS signals. In the GNSS receiver model used in this work, we will focus primarily on the code delay and carrier phase measurements. The antenna-induced biases on these measurements were found by incorporating the response of the antenna into the receiver’s cross-correlation function and observing its effects in isolation. These bias errors can be mitigated by using antenna calibration procedures. While these effects are straightforward for simple, single-element GNSS antennas, there will be some important new considerations when dealing with GNSS antenna arrays. The next chapter will introduce our adaptive antenna array model and highlight these differences.
CHAPTER 3

GNSS Adaptive Antenna Arrays

The weak received power of GNSS satellite signals leaves them vulnerable to interference. As it stands today, antenna arrays are the only solution for allowing reception of GNSS signals in harsh interference environments. The content of later chapters will focus almost exclusively on adaptive antenna arrays, and this chapter provides the requisite context and mathematical model for these systems. Section 3.1 begins with an overview of interference. Here we establish the interference scenarios in which we will operate and discuss the motivation behind our use of arrays over other alternatives. Section 3.2 introduces GNSS adaptive antenna arrays, and Section 3.3 defines a mathematical model for a general adaptive antenna array which allows one to analytically study the performance and behavior of these systems. In Section 3.4, a variety of common adaptive filtering algorithms are reviewed. Section 3.5 discusses how to incorporate our adaptive antenna model into the traditional GNSS receiver model in order to understand how the antenna will affect GNSS measurements. Finally, Section 3.6 shows example results that compare the interference suppression performance of common adaptive filtering algorithms. These results demonstrate how these systems
distort the received satellite signal, introduce errors into receiver measurements, and cannot be calibrated using traditional approaches.

3.1 Overview of the Interference Threat

Unintentional radio-frequency interference on GNSS frequencies can originate from a variety of sources, including aircraft avionics and ground-based sources such as radar and aeronautical emitters [7, 8, 50]. In some cases, navigation signals share spectrum with other signals [7, 30]. In a more difficult case, the interference could come from hostile jamming. There are a range of interference and jamming waveforms including continuous wave (CW), pulsed CW, narrowband and broadband Gaussian noise, and phase-shift keyed pseudo-noise [3]. To understand the detrimental effects of interference, it is necessary to appreciate the weak signal strength of GNSS satellite signals. After accounting for path-loss and receiver hardware, the power of the received GNSS satellite signal is 20-30 dB below the noise floor for typical GNSS receivers. However, since the ranging signal is known to the receiver, it can be coherently integrated with a locally generated replica. Any noise or interference is destructively canceled in proportion with the integration length, and, after a typical 10 ms integration, the result is a processing gain of 54 dB for a GPS P-Code signal [1, 8]. Unfortunately, even if the level of interference is tolerable to the receiver, the additional noise will degrade the range measurements and navigation solution [4, 51, 52]. In the worst case, when interference overcomes the receiver’s processing gain, the GNSS receiver would not be able to acquire or track GNSS signals at all.
One option for overcoming interference is to use alternative navigation sensors to supplement and provide failover for GNSS capabilities. The measurements from different sensors can be synergistically combined to form navigation solutions. For example, an inertial measurement unit can be combined with measurements from optical or lidar imaging sensors. While GNSS reception is available, these alternative sensor measurements are fused with GNSS measurements in order to provide improved navigation accuracy [53]. When, interference degrades the availability of GNSS signals, these alternative sensors attempt to provide sufficient navigation information on their own. These systems have been the focus of considerable research, and while there is little doubt in the benefit of additional sensors, integrating a package of sensors capable of replacing GNSS still poses a challenging problem. GNSS exists, in part, because its features are not easily duplicated. Often alternative navigation systems are only useful for short outages, and GNSS reception is ultimately required to meet navigation objectives. This leads to today’s current demand for interference suppression capabilities in GNSS receiver systems.

For GNSS receivers with a single antenna element, the only form of interference suppression available is temporal processing. For example, the antenna front-end electronics provide some level of interference protection by band-limiting received signals. Additionally, the automatic gain control (AGC) in the front-end can be designed to minimize the effects of interference. Time-domain pulse blanking and frequency-domain notch filters are also used [9–12]. Since the GNSS signal is well below the noise floor, it is straightforward to detect which frequencies contain interference. Unfortunately, there are two primary problems with these approaches. First, these approaches can distort the satellite signal. Second, temporal processing can
only be applied in cases where the interference has a narrow bandwidth relative to the GNSS signal. For environments with wideband interferers which occupy the entire satellite signal bandwidth, there is very little temporal processing can accomplish.

3.2 GNSS Adaptive Antenna Arrays

State-of-the-art interference suppression systems for GNSS receivers utilize a multi-element antenna array with an adaptable reception pattern. Fig. 3.1 shows a number of example antenna arrays being developed today. Conventional arrays are composed of 4 to 7 patch elements arranged circularly on a planar surface as shown in Fig. 3.1 (a). Larger, non-planar arrays have also been developed, as shown in Fig. 3.1 (b) [54–56]. The diameter of these conventional arrays are approximately 12 inches and are often mounted on large aircraft or ships. In order to increase their use on smaller platforms, miniaturized arrays have been recently developed that are only 4 or 6 inches in diameter [57,58]. Some examples are shown in Fig. 3.1 (c) and (d). In order to cover a wider range of GNSS frequencies, these arrays are being composed of spiral elements instead of patches. In the past, GNSS arrays were often criticized for being large and bulky; however, these miniaturized arrays offer good interference suppression with aperture sizes that rival current, single-element antennas.

Adaptive arrays allow greater performance over a single element antenna by virtue of their ability to provide beamforming/null steering in specific directions. This gives the system spatial degrees of freedom with which to separate the desired and undesired signals. Interference suppression performance can be improved by adding an adaptive FIR filter behind each antenna element. This adds a temporal filtering aspect to the spatial filtering, and the resulting filter is commonly referred to
Figure 3.1: Example GNSS adaptive antennas.

as a space-time adaptive processor (STAP). Using an adaptive algorithm, the filter weights of the adaptive antenna are chosen to constructively preserve a signal-of-interest (SOI) while destructively suppressing any incident interference and noise. There are numerous algorithms that can be used to determine these weights, each
with their own requirements and benefits [19]. Recall that interference suppression methods for single-element antennas are applicable only to a limited class of interference scenarios (those involving a small number of narrowband interferers). However, STAP-based adaptive antennas are capable of handling harsh interference scenarios with multiple wideband interferers. While these systems are clearly more complex and costly than single antenna systems, there are many users who are willing to accept these drawbacks since they offer the only means to receive GNSS signals in the widest class of interference environments.

3.3 Adaptive Antenna Array Model

The adaptive antenna model used in the present work is depicted in Fig. 3.2. It consists of an array of \( K \) individual antenna elements, where \( A_k(f, \theta, \phi) \) represents the response of the \( k \)th antenna element in the \((\theta, \phi)\) direction. The response of each element is relative to a common antenna reference point, and each response is measured or simulated \textit{in-situ} (on the platform, in the presence of the other elements). In this way, it is implied that any effects of the platform (i.e. scattering from platform structures) are automatically incorporated into the antenna response, and any mutual coupling between the elements in the array is also included. The complete set of \( K \) antenna element responses \( \{A_k(f, \theta, \phi), k = 1 \ldots K\} \) is referred to as the antenna array manifold and is used extensively in the rest of this work.

Directly attached to each antenna element is the front-end hardware, which downconverts the signal to baseband and performs ADC. Each antenna element in the array has its own RF channel with independent components which typically begin with a
LNA. Subsequently, the signal passes through mixers and band-pass filters that down-
convert the signal to various intermediate frequencies (IF). The front-end hardware
determines the system bandwidth, and $F_k(f)$ is used to denote the response of the
front-end electronics behind the $k$th element. For our simulations, it will be assumed
that $F_k(f)$ is simply unity and has no effect on the received signals in order to focus
on the the physical antenna and the adaptive filtering. Note that multiple GNSS
signals are simultaneously received from multiple frequency bands, and the front-end
is commonly segmented so that each band is output separately. For convenience, the
digital output of each front-end channel is considered to be complex even though it
is typically implemented as two separate I and Q channels [1].

The digital signal at the output of each channel enters an adaptive filter, one
for each antenna element. This set of $K$, $L$-tap FIR filters is known as a space-
time adaptive processor. The outputs of all filters are summed to produce a single
adaptive antenna output signal. Older systems tend to use a small number of adaptive

![Figure 3.2: Model of an antenna array with a space-time adaptive filter.](image-url)
filter taps (i.e. 3, 5 or 7 taps); however, increased processing power in more recent implementations allow for much longer filters [59]. If there is only a single tap, then the system is a traditional phased array with a single complex weight behind each element. Note that in Fig. 3.2, the output of each front-end channel enters a single adaptive FIR filter. However, in practice, the same front-end output is often sent to multiple adaptive filter channels, which allows each adaptive filter to perform beamforming independently in different satellite directions. For our discussion, we will focus on a single satellite signal and a single channel. For adaptive algorithms that provide beamforming, it is implied that there is a separate channel for each satellite signal. Similarly, it is implied that there are separate channels for signals on different frequency bands.

3.3.1 Signals in the Adaptive Antenna

The digital output of the $k$th front-end channel is denoted by $x_k[n]$, and the instantaneous signal snapshot on the $L$ taps of the $k$th filter is denoted

$$x_k[n] = [ x_k[n] \ldots x_k[n - L + 1] ]^T.$$ (3.1)

The snapshots of each of the $K$ filters are combined into a single received signal snapshot vector $x[n]$,

$$x[n] = \begin{bmatrix} x_1[n] \\ \vdots \\ x_K[n] \end{bmatrix}.$$ (3.2)

Each complex weight in the STAP filter is designated $w_{kl}$, which corresponds to the $l$th weight of the $k$th filter. The $k$th filter is represented by the $Lx1$ vector

$$w_k = [ w_{k1} w_{k2} \ldots w_{kL} ]^T.$$ (3.3)
Typically, all of the filter weights are combined into a single stacked vector $\mathbf{w}$,

$$
\mathbf{w} = \begin{bmatrix}
\mathbf{w}_1 \\
\vdots \\
\mathbf{w}_K 
\end{bmatrix}.
$$

(3.4)

The output $y_k[n]$ of the $k$th filter is

$$
y_k[n] = \mathbf{x}_k[n] \mathbf{w}_k,
$$

(3.5)

and all of the individual output signals are summed to produce the single output signal $y[n]$ of the entire adaptive antenna,

$$
y[n] = \sum_{k=1}^{K} y_k[n] = \sum_{k=1}^{K} \mathbf{x}_k[n] \mathbf{w}_k = \mathbf{x}[n] \mathbf{w} = \mathbf{w}^T \mathbf{x}[n].
$$

(3.7)

In a real system, the weights $\mathbf{w}$ will change in time as the adaptive processor reacts to a changing signal scenario. In this case, we should introduce a time dependence on the weights; however, for much of our analysis, this will be not necessary. In a stationary signal environment, the weights will converge to a steady state value. Since this work only deals with adaptive filters in the steady-state, the weights will be written as if they are independent of time.

For our analysis, all of the signals in the adaptive antenna are decomposed into three components: the desired satellite signal, the thermal noise and the interference. Only a single satellite signal will be considered, while the interference component is the sum over all interference sources. All three components are assumed to be independent of each other. The signal snapshot vector is written as the sum

$$
\mathbf{x}[n] = \mathbf{x}_d[n] + \mathbf{x}_n[n] + \mathbf{x}_i[n],
$$

(3.8)
where $x_d[n]$, $x_n[n]$, and $x_i[n]$ are the desired, noise, and interference components, respectively. Sometimes it will be useful to refer to a single undesired component $x_u[n]$, 

$$x_u[n] = x_i[n] + x_n[n].$$

(3.9)

It follows that the output signal can also be decomposed into components,

$$y[n] = x^T[n]w$$

$$= (x_d[n] + x_n[n] + x_i[n])^T w$$

$$= y_d[n] + y_n[n] + y_i[n],$$

(3.10)

where $y_d[n]$, $y_n[n]$ and $y_i[n]$ correspond to the respective signal components. Note that we will use $n$ as both a discrete time index and as a subscript indicating the noise component. Context should make it clear which meaning is intended.

### 3.3.2 Power Spectra

In our study, the incident signals are completely characterized by their power spectral densities. The SOI has an incident power $C_d$, a normalized power spectral density $G_d(f)$, and an incidence angle of $(\theta_d, \phi_d)$. There are $M$ interferers where the $m$th interference signal is incident from $(\theta_{i,m}, \phi_{i,m})$, has a power $C_{i,m}$, and has a normalized power spectral density $G_{i,m}(f)$. The thermal noise from the $k$th front-end channel has power $C_{n,k}$ and normalized power spectral density $G_{n,k}(f)$. The power spectral densities of the SOI, interference and noise components are then

$$\bar{S}_d(f) = C_d G_d(f)$$

(3.11)

$$\bar{S}_i(f) = \sum_{m=1}^{M} C_{i,m} G_{i,m}(f)$$

(3.12)

$$\bar{S}_{n,k}(f) = C_{n,k} G_{n,k}(f) ,$$

(3.13)
respectively.

Based on the adaptive antenna model depicted in Fig. 3.2, the responses of the antenna, front-end and STAP filters can be combined into a single direction-dependent adaptive antenna response $H(f, \theta, \phi)$,

$$H(f, \theta, \phi) = \sum_{k=1}^{K} W_k(f) F_k(f) A_k(f, \theta, \phi).$$  \hspace{1cm} (3.14)

Here, the frequency response $W_k(f)$ of the $k$th adaptive filter is

$$W_k(f) = \sum_{l=1}^{L} w_{kl} e^{-j2\pi f(l-1)T_0},$$  \hspace{1cm} (3.15)

where $T_0$ is the sampling period.

From the filter response in (3.14), the power spectra at the output of the adaptive antenna can be found. The power spectral density of the desired component at the output of the adaptive antenna is

$$S_d(f) = C_d |H(\theta_d, \phi_d, f)|^2 G_d(f),$$  \hspace{1cm} (3.16)

while the output power spectral density of the undesired component is

$$S_u(f) = S_n(f) + \sum_{m=1}^{M} S_{i,m}(f)$$  \hspace{1cm} (3.17)

where $S_n(f)$ is the power spectral density of the thermal noise and, for $M$ interferers, $S_{i,m}(f)$ is the power spectral density of the $m$th interferer. The power spectral density of the total interference component is

$$S_i(f) = \sum_{m=1}^{M} S_{i,m}(f)$$  \hspace{1cm} (3.18)

$$= \sum_{m=1}^{M} C_{i,m} |H(\theta_m, \phi_m, f)|^2 G_{i,m}(f).$$  \hspace{1cm} (3.19)
The power spectral density of the thermal noise component must be treated in a different manner. Noise is generated independently in the antenna electronics of each antenna element. Furthermore, the noise does not go through the antenna, only the front-end and the adaptive filter. All of the noise can be considered as a single signal by defining the effective system response $H_n(f)$ imposed on the noise component as

$$H_n(f) = \sqrt{\sum_{k=1}^{K} |W_k(f)F_k(f)|^2},$$  \hspace{1cm} (3.20)

so that the noise spectral density at the output of the antenna is

$$S_n(f) = |H_n(f)|^2 C_n.$$  \hspace{1cm} (3.21)

Finally, the output power of each component is

$$P_d = \int S_d(f) df$$  \hspace{1cm} (3.22)

$$P_i = \int S_i(f) df$$  \hspace{1cm} (3.23)

$$P_n = \int S_n(f) df,$$

where it is implied the integrals are over the bandwidth of the system. These equations allow one to analytically analyze the interference suppression performance and to study the output spectra of the various signal components. The signal-to-interference-plus noise ratio (SINR) $\rho_0$ is given by

$$\rho_0 = \frac{P_d}{P_i + P_n}$$  \hspace{1cm} (3.25)

is often used to characterize the signal quality at the output of the adaptive antenna.

### 3.3.3 Correlation Matrices

In order to adapt the digital filter weights, the adaptive processor must first acquire information about the incident signal scenario. To accomplish this, the adaptive
processor forms a correlation matrix between the signals on each tap of each antenna element. The received signal correlation matrix $\Phi$ is given by

$$\Phi = E \left\{ x^*[n]x_T[n] \right\}, \quad (KL \times KL)$$

where the signal scenario is assumed static, and $x[n]$ is assumed to be wide sense stationary. $\Phi$ is composed of submatrices

$$\Phi = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1K} \\ \vdots & \ddots & \vdots \\ \Phi_{K1} & \cdots & \Phi_{KK} \end{bmatrix},$$

(3.27)

where each submatrix $\Phi_{rs}$ is the correlation between the taps of the $r$th and $s$th elements

$$\Phi_{rs} = E \left\{ x_r^*[n]x_s^T[n] \right\}. \quad (L \times L)$$

(3.28)

Since the components of the received signal in (3.8) are defined as independent, $\Phi$ can be decomposed as

$$\Phi = E \left\{ x^*x^T \right\}$$

(3.29)

$$= E \left\{ (x_d + x_n + x_i)^* (x_d + x_n + x_i)^T \right\}$$

(3.30)

$$= E \left\{ x_d^*x_d^T \right\} + E \left\{ x_n^*x_n^T \right\} + E \left\{ x_i^*x_i^T \right\}$$

(3.31)

$$= \Phi_d + \Phi_n + \Phi_i,$$

(3.32)

where $\Phi_d$, $\Phi_n$ and $\Phi_i$ correspond to the desired, thermal noise and interference components, respectively. $\Phi_u$ will be used to refer to the total undesired component,

$$\Phi_u = \Phi_n + \Phi_i.$$
A simple expression for the total signal power $P$ at the output of the adaptive array is

$$ P = \frac{1}{2} E \{ y[n] y^*[n] \} \quad (3.34) $$

$$ = \frac{1}{2} w^H E \{ x^*[n] x^T[n] \} w \quad (3.35) $$

$$ = \frac{1}{2} w^H \Phi w , \quad (3.36) $$

and the power of each component is given similarly as

$$ P_d = \frac{1}{2} w^H \Phi_d w \quad (3.37) $$

$$ P_i = \frac{1}{2} w^H \Phi_i w \quad (3.38) $$

$$ P_n = \frac{1}{2} w^H \Phi_n w . \quad (3.39) $$

This allows SINR, defined in (3.25), to be written as

$$ \rho_0 = \frac{w^H \Phi_d w}{w^H \Phi u w} . \quad (3.40) $$

In practice, the STAP processor forms an estimate of the received signal correlation matrix using a variety of possible implementations [18, 19]. However, this study will focus only on the steady-state performance of the adaptive filters, so the correlation matrices will be formed analytically, where the $(p,q)$ entry of each submatrix for each component are given by

$$ [\Phi_{d,r,s}]_{pq} = \int \bar{S}_d(f) F_r^*(f) F_s(f) A_r^*(f, \theta_d, \phi_d) A_s(f, \theta_d, \phi_d) e^{-j2\pi f(q-p)T_0} df \quad (3.41) $$

$$ [\Phi_{n,r,s}]_{pq} = \begin{cases} 
\int \bar{S}_n(f) |F_r(f)|^2 e^{-j2\pi f(q-p)T} df & r = s \\
0 & r \neq s 
\end{cases} \quad (3.42) $$

$$ [\Phi_{i,r,s}]_{pq} = \int \left( \sum_{m=1}^{M} \bar{S}_{i,m}(f) F_r^*(f) F_s(f) A_r^*(f, \theta_{i,m}, \phi_{i,m}) A_s(f, \theta_{i,m}, \phi_{i,m}) \right) e^{-j2\pi f(q-p)T_0} df . \quad (3.43) $$

35
3.4 Correlation Vector

This section will revisit the GNSS receiver model introduced in Section 2.2 in order to define an important quantity called the correlation vector.

Consider a GNSS receiver which has acquired a GNSS signal and has entered tracking mode. The receiver cross-correlates the output of the adaptive antenna with a locally generated reference signal. Let us assume the weight vector \( \mathbf{w} \) is effectively constant (in the steady state) over some interval \( n \in \{1, ..., N\} \) corresponding to the integration length. Then we can write the estimated receiver cross-correlation function \( \hat{R}_{yd}(\tau) \) in (2.5) has the form

\[
\hat{R}_{yd}(\tau) = \frac{1}{N} \sum_{n=1}^{N} y[n] d(nT_0 + \tau - \tau_{ref}) (3.44)
\]

\[
= \left[ \frac{1}{N} \sum_{n=1}^{N} x^T[n] d(nT_0 + \tau - \tau_{ref}) \right] \mathbf{w} (3.45)
\]

where \( d(t) \) is the receiver reference signal, and \( \tau_{ref} \) is

\[
\tau_{ref} = \left\lfloor \frac{L}{2} \right\rfloor T_0, 
\]

which corresponds to the delay of the received signal on the center tap (reference tap) of the adaptive filter. The effect of \( \tau_{ref} \) is to shift the time reference of the correlation function \( R_{yd}(\tau) \). The purpose of this is made clear in Section 3.6, when antenna-induced delays are considered. In this study, the reference signal \( d(t) \) is an exact replica of the incident GNSS signal, and there is an important implied relation in (3.44) between the timing reference and received signals. That is, the reference signal corresponds to the GNSS signal received by an isotropic antenna located at the phase reference point of the antenna array patterns \( A_k(f, \theta, \phi) \). Thus, the cross-correlation functions between the received and reference signals captures the effects of
the adaptive antenna on the signal, and any code delay, phase shift, of other distortion is attributed solely to the antenna.

To consider the bias errors caused by the adaptive antenna on the GNSS signal, we will use the mean of the cross-correlation in (3.45). In this case, the finite summation is replaced by an expectation that will be defined as a vector \( \mathbf{s}(\tau) \)

\[
\mathbf{s}(\tau) = E \{ \mathbf{x}^*[n]d(nT_0 + \tau - \tau_{ref}) \} \tag{3.47}
\]

in order to yield a convenient, vector form for the cross-correlation function,

\[
R_{yd}(\tau) = E \{ \hat{R}_{yd}(\tau) \} = \mathbf{s}^H(\tau)\mathbf{w}. \tag{3.48}
\]

\( \mathbf{s}(\tau) \) is known as the correlation vector. This \( KLx1 \) vector is composed of \( K \), \( Lx1 \) correlation vectors corresponding to each antenna element,

\[
\mathbf{s}(\tau) = \begin{bmatrix}
\mathbf{s}_1(\tau) \\
\vdots \\
\mathbf{s}_K(\tau)
\end{bmatrix}, \tag{3.50}
\]

where the \( l \)th entry of the \( k \)th vector is given by

\[
[ s_k(\tau) ]_l = \sqrt{C_d} \int G_d(f) F_k(f) A_k(f, \theta, \phi) e^{-j2\pi f(lT_0 + \tau - \tau_{ref})} df. \tag{3.51}
\]

Since the correlation vector depends on the incident signal angle, it will sometimes be written as \( \mathbf{s}(\tau, \theta, \phi) \).

Evaluation of (3.51) requires the power spectrum of the SOI, the front-end response of each channel, and the response of each antenna element in the array. For most of the research carried out in this work as well as for calculating the code delay and carrier phase biases (see Chapter 2), the incident SOI power \( C_d \) is unnecessary.
and can be ignored. All of the remaining quantities in (3.51) are known and do not depend on the adaptive filter weights. Therefore, the correlation vector for a particular relative delay, $\tau$, can be pre-processed and stored in a look-up table. This can be done for all angles-of-arrival and signal spectra. Some of the STAP algorithms used in this study will utilize the correlation vector, and it is implied that these algorithms use a pre-calculated and stored version of this vector. Note that, when the correlation vector is evaluated at $\tau = 0$, it is referred to as the reference correlation vector and is denoted simply as $s$.

### 3.5 Common Adaptive Filtering Algorithms

Three different STAP algorithms will be used repeatedly for simulations throughout this work. They will only be discussed briefly here since the purpose of this work is not to analyze any particular one in detail. They were chosen collectively to be diverse and representative of popular STAP techniques. While there are many other different algorithms available, they tend to be incremental variations on the ones presented here.

#### 3.5.1 Power Inversion

The first STAP algorithm, commonly implemented for its simplicity, constrains the array such that the output power is minimized while a specified reference tap remains on. This method is called power inversion and only performs interference suppression. It does not provide gain by forming a beam in the SOI direction; however, it also does not require knowledge of the antenna or SOI. This method suppresses all strong signals indiscriminately, and it should only be used when the SOI is sufficiently weak,
as is the case in GNSS systems. Its weights are given by

\[ \mathbf{w} = \mu \Phi^{-1} \mathbf{u}, \]  

(3.52)

where \( \mu \) is an inconsequential constant, \( \mathbf{u} \) corresponds to the steering vector,

\[ \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_k \end{bmatrix}, \]  

(3.53)

and the \( k \)th subvector \( \mathbf{u}_k \) is

\[ \mathbf{u}_k = \begin{cases} \begin{bmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 & 0 \end{bmatrix}^T, & \text{if } k = k_r \\ \mathbf{0}, & \text{if } k \neq k_r \end{cases} \]  

(3.54)

Here, \( k_r \) is the index of the reference element. For our simulations, the center element of the circular array will be chosen to be the reference element, as is typically the case. A more detailed explanation is given in [60]. Figures will denote it as 'PI'.

### 3.5.2 Beam Steering

The next adaptive filtering algorithm is commonly referred to as simple beam steering. It constrains the array to provide gain in the SOI direction while minimizing the total output power of the STAP filter. The weights are given in an identical form as (3.52), except that

\[ \mathbf{u}_k = \begin{bmatrix} 0 & 0 & \cdots & u_k^* & \cdots & 0 & 0 \end{bmatrix}^T \]  

(3.55)

and \( u_k^* \) (located at the reference tap) represents the conjugate of the voltage that would be induced on the \( k \)th element due to the SOI. As a result, this method requires knowledge of the SOI direction and the antenna array pattern. This method is also known as directionally constrained minimum power and similar methods are dealt with in [19, pp. 513-516]. Figures will denote it as 'DC'.

39
3.5.3 MMSE

The third adaptive filtering algorithm constrains the adaptive antenna such that the output signal produces the minimum mean-squared error (MMSE) with respect to some reference signal. The weights are given by

\[ w = \mu \Phi^{-1}s, \]  

where \( s \) is the reference correlation vector defined in Section 3.4. In many implementations, the correlation vector is determined at run time by performing the cross-correlation in (3.47) directly. However, since all of the quantities in (3.51) can be acquired beforehand, the correlation vector can be pre-calculated and stored. When MMSE is used in this work, we will assume that the correlation vector has been pre-calculated. The MMSE method provides both beamforming and null steering, and its error minimizing behavior helps reduce distortion in the received signal. However, it also has the greatest requirements since formation of \( s \) requires knowledge of the entire antenna array manifold as well as the SOI power spectrum. Figures will denote it as 'MMSE'.

3.6 Adaptive Antenna-Induced Errors in GNSS Measurements

Unfortunately, the effects of an antenna on received GNSS signals are not negligible for precision applications. This is especially true when the antenna is a large antenna array with complicated filtering, such as the case with a SOP- or STAP-based adaptive antenna. In a manner similar to Section 2.3, this section formulates equations for the adaptive antenna-induced biases on the GNSS receiver measurements.
To determine the adaptive antenna-induced biases, it is necessary to consider the effects of the adaptive antenna on the receiver cross-correlation. Without loss of generality, let us assume that the incident and reference signals are synchronized. In this way, any code or carrier phase measured from the cross-correlation corresponds solely to antenna-induced biases. The responses of the antenna, front-end and STAP filters can be combined into a single direction-dependent adaptive antenna response given in (3.14) as

$$H(f, \theta, \phi) = \sum_{k=1}^{K} W_k(f) F_k(f) A_k(f, \theta, \phi).$$  \hspace{1cm} (3.57)

It follows that the cross-correlation is filtered by the adaptive antenna response,

$$R_{yd}(\tau, \theta, \phi) = \sqrt{C_d} \int H(f, \theta, \phi) G_d(f) e^{-j2\pi f (\tau - \tau_{ref})} df.$$  \hspace{1cm} (3.58)

where $G_d$ is the normalized power spectral density of the SOI. Depending on the length of the filters and the adaptive algorithm, the response (3.57) can potentially distort the cross-correlation (3.58) in a way that is dependent on the the incident signal scenario.

The adaptive antenna-induced biases are found using a method that is nearly identical to the single-element antenna in the last chapter. A GNSS receiver finds the peak of the magnitude of the cross-correlation function in order to measure the code delay of the received signal relative to its local clock using (3.49). If there was no antenna-induced bias, then it would have a peak at $\tau = 0$. However, the adaptive filter weights can distort and delay the SOI and cause the peak to be shifted. The new peak location is the antenna-induced code delay bias $\tau_0$ and is given by

$$\tau_0(\theta, \phi) = \arg\max_{\tau} |R_{yd}(\tau, \theta, \phi)|.$$  \hspace{1cm} (3.59)
The receiver will adjust the delay of the reference signal until the relative delay between it and the incident signal is $\tau_0$. However, the carrier phase is adjusted independently. The antenna-induced carrier phase bias $\psi_0$ corresponds to the measured carrier phase when the signal is actually aligned (that is, at $\tau = 0$), which is given by

$$\psi_0(\theta, \phi) = \angle R_{yd}(0, \theta, \phi).$$  \hspace{1cm} (3.60)

As mentioned earlier, $A_k(f, \theta, \phi)$ represents the *in-situ* response of the $k$th antenna element relative to some common phase reference point. This reference point is chosen to correspond to a known physical location on the antenna. As a result, the cross-correlation function given in (3.49) is equivalent to the cross-correlation between a signal processed by the antenna response and the same signal received by an isotropic receiver located at the reference point. These antenna-induced biases are understood to be relative to this antenna reference point.

### 3.7 Example Results

Simulations were performed in order to demonstrate the typical interference suppression performance of commonly used algorithms. At the same time, it is shown that these algorithms introduce significant bias errors in the GNSS receiver measurements and that these biases cannot be accounted for using traditional GNSS antenna calibration techniques.

The antenna array used in the simulations is a 6-element circular array of patches mounted on an infinite ground plane as shown in Fig. 3.3. A patch array was chosen since they are commonly used for GNSS applications. The antenna pattern was simulated at the GPS L1 band (1.576 GHz) using the computational electromagnetics program FEKO [61]. The front-end thermal noise is assumed to be white, zero-mean
Figure 3.3: 6-element circular array of patch antennas.

Gaussian. The front-end filters are assumed to be flat and equalized for all elements so that they are effectively neglected. The system bandwidth and the corresponding sampling rate were chosen to be 24MHz, and the STAP filter has 7 taps (unless otherwise noted). The SOI is a BPSK signal with 10MHz chipping rate which will give it a sinc-squared power spectrum. The SOI has an incident signal power of -30 dB SNR at 24 MHz system bandwidth, and its angle-of-arrival will vary. Receiver performance metrics are evaluated using the analytic equations provided earlier in this chapter, and the STAP filter weights will be analytically computed assuming the weights have converged to a steady state in a stationary signal scenario. In our coordinate system, $\phi$ is the azimuthal angle and $\theta$ is the elevation angle, where $\theta = 0$ corresponds to boresight.

The interference bandwidths, relative powers, and incidence angles are provided in Table 3.1. Wideband interferers occupy the entire system bandwidth while partial-band interferers have a narrower bandwidth and are offset from the L1 carrier frequency. All interferers are strong with a 50 dB interference-to-noise ratio (INR) and are incident from angles near the horizon. Three different scenarios will be simulated.
Table 3.1: Interference Properties

<table>
<thead>
<tr>
<th>Interferer</th>
<th>AOA (θ, φ)</th>
<th>Bandwidth</th>
<th>Freq. Offset</th>
<th>INR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(80°, 0°)</td>
<td>24 MHz</td>
<td>-</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>(86°, 120°)</td>
<td>24 MHz</td>
<td>-</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>(78°, 270°)</td>
<td>24 MHz</td>
<td>-</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>(89°, 60°)</td>
<td>0.5 MHz</td>
<td>+2 MHz</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>(83°, 200°)</td>
<td>0.5 MHz</td>
<td>-4 MHz</td>
<td>50</td>
</tr>
<tr>
<td>6</td>
<td>(86°, 330°)</td>
<td>0.5 MHz</td>
<td>+1 MHz</td>
<td>50</td>
</tr>
</tbody>
</table>

The first has no interference, the second has 3 wideband interferers, and the last contains 3 partial-band interferers. Note that the choice of offset partial-band interferers merely serves to emphasize the degradation of certain algorithms in use today.

In the first signal scenario, no interference is present. Fig. 3.4 shows the resulting $C/N_0$, carrier phase bias, and code delay bias as the elevation angle $\theta$ of the SOI is varied from -90 to 90 degrees in the $\phi = 0$ plane. The small variation in $C/N_0$ versus angle is solely due to pattern of the antenna elements, which exhibit less gain at lower elevations. The results show that the MMSE algorithm achieves zero carrier phase bias, zero code delay bias, and has very good $C/N_0$ performance. The simple directional constraint method also has good $C/N_0$ performance and zero carrier phase bias; however, it also has significant code delay bias errors. The simple power minimization method has suboptimal $C/N_0$ since it does not provide beam-forming. Furthermore, it has significant bias errors in both carrier phase and code delay. This defines our baseline performance.
In the second signal scenario, there are 3 strong, wideband interferers. They have a bandwidth of 24 MHz and a 50 dB SNR. One interferer lies in the cut at $\theta = 80$ deg. while the other interferers originate from low elevation angles outside the cut. Fig. 3.5 shows the resulting $C/N_0$, carrier phase bias, and code delay bias vs. angle. In this case, we observe that the presence of interference has caused a general reduction in $C/N_0$ around the interferer. Furthermore, two of the algorithms exhibit increased bias errors. The MMSE algorithm, however, still yields very little antenna-induced bias. This is true even at angles very close to the interference.

The third signal scenario has 3 partial-band interferers. They have a bandwidth of 0.5 MHz and a 50 dB SNR. Their center frequencies are spread randomly in the band in order to create a temporally complex environment. Fig. 3.6 shows the resulting $C/N_0$, carrier phase bias, and code delay bias vs. angle. Now, in this temporally complex scenario, all three algorithms suffer severe antenna-induced biases in the code delay measurements. However, two of the algorithms are able to prevent carrier phase bias errors. As we will see in Chapter 5, carrier phase errors are easier to prevent than code delay errors.

From these results it is clear that all of the algorithms presented here produce antenna-induced biases. It is often assumed that algorithms such as MMSE are distortionless since they minimize the error relative to a reference signal. However, it is clear that it is easy to find interference scenarios in which this is not true. None of these existing algorithms have been specifically designed to prevent antenna-induced biases. It should be noted that if the adaptive antenna were to be calibrated in the absence of interference, then this calibration data would not necessarily be applicable in the presence of interference. This can be understood by comparing Fig.
3.4 with Fig. 3.6. We conclude that, although these algorithms suppress interference adequately, they are not suitable for precision GNSS receivers.
Figure 3.4: SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the absence of interference.
Figure 3.5: SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the presence of 3 wideband interferers.
Figure 3.6: SINR, code delay bias, and carrier phase bias for the different STAP algorithms in the presence of 3 offset, partial-band interferers.
3.8 Summary & Conclusion

This chapter introduced the adaptive antenna array model which will be used throughout this work. This analytic model incorporates the effects of the adaptive antenna on the receiver cross-correlation function and the delay and phase estimates that are based on it. A variety of common adaptive filtering algorithms were simulated both in the presence and absence of interference. It was shown that they exhibit different interference suppression performance. Additionally, it was shown that these adaptive antennas introduce significant bias errors into the GNSS receiver measurements, and that these errors cannot be removed using traditional GNSS antenna calibration procedures. These results define the baseline performance of GNSS adaptive antenna systems in use today. The next three chapters will develop new adaptive antenna algorithms and techniques to overcome the deficiencies presented here.
CHAPTER 4

Optimal $C/N_0$ of GNSS Receivers with Adaptive Antennas

In order to achieve optimal GNSS receiver accuracy, it is necessary to develop an adaptive filtering algorithm that can optimally suppress interference. Many existing adaptive algorithms are commonly optimized to maximize output signal-to-interference-plus-noise ratio (SINR); however, GNSS receiver performance is primarily dependent on a different metric: receiver post-correlation carrier-to-noise ratio ($C/N_0$). In this chapter, the distinction between SINR and $C/N_0$ will be established, and a novel adaptive algorithm that maximizes $C/N_0$ will be derived. The performance achieved by this algorithm defines the theoretical performance bound for interference suppression for a GNSS receiver with an adaptive antenna. Section 4.1 begins by defining $C/N_0$ and its relation to receiver performance. It also compares the mathematical forms of SINR and $C/N_0$ in order to demonstrate their differences. Section 4.2 derives a novel vector formula for $C/N_0$, and Section 4.3 uses it to develop a novel adaptive filtering algorithm that maximizes $C/N_0$. Section 4.4 shows the performance of common STAP algorithms in the presence of interference, and compares them to this performance bound. The results demonstrate that adaptive filters which optimize SINR behave very differently than those that optimize $C/N_0$. 

51
4.1 $C/N_0$ and SINR

In GNSS receivers, signal quality is measured by $C/N_0$ [51], while the interference suppression performance of an adaptive array is measured using the SINR at the output of the antenna array. It is commonly assumed that these two performance metrics are directly proportional to each other. Unfortunately, this is only approximately true. SINR only measures the relative powers of the SOI, interference, and noise components of the antenna output signal. Receiver $C/N_0$ is a post-correlation performance metric which depends on processing gain, which, in turn, depends on the relative power spectral densities of the SOI, interference and noise. As a result, there can be significant discrepancies between the two metrics. Since adaptive antennas are being actively developed for GNSS applications, it is very important to understand the differences between SINR and $C/N_0$ performance.

The first stage of signal processing in the GNSS receiver involves correlation of the antenna output signal with a reference signal. Therefore, the SINR at the adaptive antenna output is referred to as pre-correlation (or pre-integration) SINR. Pre-correlation SINR itself is defined simply as the ratio of output powers, and was given in Section 3.2 as

$$\rho_0 = \frac{P_d}{P_n + P_i},$$

(4.1)

where $P_d$, $P_n$, and $P_i$ are the desired GNSS signal power, thermal noise power, and interference power, respectively, at the output of the adaptive antenna. It was shown that this can also be expressed in terms of the adaptive filter weights and correlation
matrices as

\[
\rho_0 = \frac{\mathbf{w}^H \Phi_d \mathbf{w}}{\mathbf{w}^H (\Phi_n + \Phi_i) \mathbf{w}} \quad (4.2)
\]

\[
= \frac{\mathbf{w}^H \Phi_d \mathbf{w}}{\mathbf{w}^H \Phi_u \mathbf{w}}. \quad (4.3)
\]

Equivalently, the power of each component can be represented using integrals of their respective power spectral densities, where their power spectral densities were shown in Chapter 3 to be

\[
S_d(f) = C_d |H(\theta_d, \phi_d, f)|^2 G_d(f) \quad (4.4)
\]

\[
S_n(f) = |H_n(f)|^2 G_n(f) \quad (4.5)
\]

\[
S_i(f) = \sum_{m=1}^{M} C_{i,m} |H(\theta_m, \phi_m, f)|^2 G_{i,m}(f). \quad (4.6)
\]

The allows the pre-correlation SINR to be represented in terms of integrals

\[
\rho_0 = \frac{C_d \int |H(\theta_d, \phi_d, f)|^2 G_d(f) df}{\int \left( S_n(f) + S_i(f) \right) df}. \quad (4.7)
\]

While pre-correlation SINR is a convenient means of understanding STAP performance, the performance analysis of GNSS receivers involves observing the post-correlation SINR. As introduced in Chapter 3, the receiver cross-correlation involves a finite integration between the STAP output, \( y[n] \), and the reference signal, \( d(t) \), is

\[
\hat{R}_{yd}(\tau) = \frac{1}{N} \sum_{n=1}^{N} y[n] d(nT_0 + \tau - \tau_{ref}), \quad (4.8)
\]

where \( T_0 \) is the sample spacing and \( \tau_{ref} \) was defined in (3.46). The post-correlation SINR \( \tilde{\rho}_0 \) is defined as the squared mean of the correlator output divided by its variance \[51\]

\[
\tilde{\rho}_0 = \left| \frac{E\{\hat{R}_{yd}(\tau_0)\}}{\text{var}\{\hat{R}_{yd}(\tau_0)\}} \right|^2. \quad (4.9)
\]
Here, \( \tau_0 \) is the peak of \( R_{yd}(\tau) \) and corresponds to the relative delay between the incident GNSS signal and the locally generated reference signal, including any biases introduced by the adaptive antenna. Since the noise was defined to be zero mean,

\[
E\{\hat{R}_{yd}(\tau)\} = R_{yd}(\tau), \tag{4.10}
\]

and the numerator in (4.9) represents the post-correlation desired component power \( \tilde{P}_d \), which is simply the square of the desired component of the cross-correlation function,

\[
\tilde{P}_d(\tau) = |R_{yd}(\tau)|^2 \\
= \left| \sqrt{C_d} \int H(\theta_d, \phi_d, f)G_d(f)e^{j2\pi f(\tau-\tau_{ref})} \, df \right|^2 \\
= C_d \left| \int H(\theta_d, \phi_d, f)G_d(f)e^{j2\pi f(\tau-\tau_{ref})} \, df \right|^2 \tag{4.11}
\]

Since the reference signal is independent of the output noise, it is well understood that, the correlation acts as a filter with the same spectrum as the reference signal and is independent of \( \tau \) [62]. Therefore, the post-correlation power spectral density of the output noise \( \tilde{S}_n(f) \) and interference \( \tilde{S}_i(f) \) are

\[
\tilde{S}_n(f) = \frac{1}{T}G_d(f)S_n(f) \tag{4.12}
\]

\[
\tilde{S}_i(f) = \frac{1}{T}G_d(f)S_i(f) \tag{4.13}
\]

and the total undesired component \( \tilde{S}_u(f) \) is

\[
\tilde{S}_u(f) = \tilde{S}_n(f) + \tilde{S}_i(f). \tag{4.14}
\]
Since the desired component represents the mean, the variance is caused by the undesired components and it follows that the post-correlation noise power is

\[
\tilde{P}_u = \text{var}\{\tilde{R}_{yd}(\tau_0)\} = \int \tilde{S}_u(f) \, df = \frac{1}{T} \int \left( G_d(f)S_n(f) + G_d(f)S_i(f) \right) \, df,
\]

which is independent of \(\tau\). This yields an equation for the post-correlation SINR

\[
\tilde{\rho}_0 = \frac{\tilde{P}_d(\tau_0)}{\tilde{P}_u} = T \frac{C_d \left| \int H(\theta_d, \phi_d, f)G_d(f)e^{j2\pi f(\tau_0 - \tau_{ref})} \, df \right|^2}{\int \left( G_d(f)S_n(f) + G_d(f)S_i(f) \right) \, df}.
\]

(4.16)

\(C/N_0\) is related to post-correlation SINR by the integration length \(T\)

\[
\frac{C}{N_0} = \frac{\tilde{\rho}_0}{T}.
\]

(4.17)

Since \(T\) has units of seconds, \(C/N_0\) has units of Hz or, in log scale, dB-Hz. By comparing the pre-correlation integrals in (4.7) to the post-correlation case in (4.16), we see that the post-correlation undesired powers take into account the reference signal spectrum and the noise spectrum. One way to quantify this effect is to measure the ratio of post-correlation SINR to pre-correlation SINR,

\[
G_p = \frac{\tilde{\rho}_0}{\rho_0}.
\]

(4.18)

Since this gain is related to the despreading of the satellite ranging signal, it can be thought of generally as processing gain. If the STAP output noise is white \((S_u(f) = N_0)\) and the total system response on the desired signal is constant \((H(\theta_d, \phi_d, f) = \alpha)\),
\( \tau_0 = 0 \), then the processing gain is simply

\[
G_p = \frac{\tilde{\rho}_0}{\rho_0} = T \left| \int S_d(f) e^{j2\pi f(\tau_0 - \tau_{ref})} \, df \right|^2 \frac{\int S_u(f) \, df}{\int S_d(f) \, df} = T \left| \sqrt{C_d} \int H(\theta_d, \phi_d, f) G_d(f) e^{j2\pi f(\tau_0 - \tau_{ref})} \, df \right|^2 \frac{\int N_0 \, df}{\int \left| H(\theta_d, \phi_d, f) \right|^2 G_d(f) \, df} = \frac{TC_d \alpha^2 \left| \int G_d(f) e^{j2\pi f(\tau_0 - \tau_{ref})} \, df \right|^2 N_0 B}{N_0 C_d \alpha^2 \int G_d(f) \, df \int G_d(f) \, df} = T/T_0, \tag{4.19}
\]

which is the common definition of processing gain for a signal in white noise. However, the conditions on the noise spectrum and desired signal system response that allow for this result are not true in a realistic adaptive antenna. It will be shown via simulations in Section 4.5 that different adaptive filtering algorithms can produce a variety of noise power-spectral densities and system responses in the desired signal direction. While this is not enough to prevent SINR from being a useful metric, it will be shown that they are enough to change the relative performance of different adaptive filtering algorithms.

### 4.2 Vector Formula for C/N₀

The pre-correlation SINR at the output of the adaptive antenna is commonly expressed in a form given by (4.2). It will be useful to derive an equation for post-correlation SINR in a similar form.
Beginning with (4.8), if the STAP weight vector is assumed constant over the interval \( n \in \{1, \ldots, N\} \), then we can write the cross-correlation function in the form

\[
\hat{R}_{yd}(\tau) = \frac{1}{N} \sum_{n=1}^{N} y[n] d(nT_0 + \tau - \tau_{ref})
\]

\[
= \frac{1}{N} \sum_{n=1}^{N} (x^T[n]w) d(nT_0 + \tau - \tau_{ref})
\]

\[
= \left[ \frac{1}{N} \sum_{n=1}^{N} x^T[n] d(nT_0 + \tau - \tau_{ref}) \right] w
\]

\[
= \hat{s}^H(\tau)w
\]

(4.20)

where we have defined the correlation signal vector \( \hat{s}(\tau) \) as

\[
\hat{s}(\tau) = \frac{1}{N} \sum_{n=1}^{N} x^*[n] d(nT_0 + \tau - \tau_{ref}).
\]

(4.21)

(4.20) is notable in that it demonstrates that a STAP filter can be equivalently applied to post-correlation signals to produce the same cross-correlation values.

Starting with (4.21), if \( x[n] \) is separated into its independent components, \( \hat{s} \) can be represented as independent components similarly as

\[
\hat{s}(\tau) = \frac{1}{N} \sum_{n=1}^{N} x^*[n] d(nT_0 + \tau - \tau_{ref})
\]

(4.22)

\[
= \frac{1}{N} \sum_{n=1}^{N} (x_d^*[n] + x_n^*[n] + x_i^*[n]) d(nT_0 + \tau - \tau_{ref})
\]

(4.23)

\[
= \hat{s}_d(\tau) + \hat{s}_n(\tau) + \hat{s}_i(\tau)
\]

(4.24)

where \( \hat{s}_d, \hat{s}_n \) and \( \hat{s}_i \) are the desired, thermal noise, and interference components, respectively. Since \( x_d \) contains the desired signal, \( \hat{s}_d(\tau) \) will quickly converge for \( \tau \) near \( \tau_0 \) and can be approximated as its mean,

\[
s(\tau) = E\{\hat{s}_d(\tau)\}
\]

\[
= E\{x_d^*[n] d(nT_0 + \tau - \tau_{ref})\}.
\]
If $\tau = 0$, then $s(\tau)$ is commonly referred to as the reference correlation vector. However, this work will consider the correlation vector at arbitrary $\tau$. It is a $KLx1$ vector that can be decomposed into $K$, $Lx1$ subvectors corresponding to each antenna element

$$s(\tau) = \begin{bmatrix} s_1(\tau) \\ \vdots \\ s_K(\tau) \end{bmatrix}.$$  \hfill (4.27)

The $l$th entry of each subvector corresponds to the $l$th adaptive filter tap and is given by

$$[s_k(\tau)]_l = \sqrt{C_d} \int G_d(f) A_k^*(\theta_d, \phi_d, f) F_k^*(f) e^{j2\pi f (IT_0 + \tau - \tau_{ref})} df.$$  \hfill (4.28)

The equation for a single estimate of $R_{yd}(\tau)$ can be decomposed as

$$\hat{R}_{yd}(\tau) = \hat{s}^H(\tau)w$$

$$= (s_d^H(\tau) + \hat{s}_n^H(\tau) + \hat{s}_i^H(\tau))w$$

$$= s_d^H(\tau)w + (\hat{s}_n^H(\tau) + \hat{s}_i^H(\tau))w$$

$$= R_{yd}(\tau) + \hat{R}_{yd}^u(\tau),$$ \hfill (4.29)

where $R_{yd}$ and $\hat{R}_{yd}^u$ are the desired and undesired components, respectively. Since the correlation in (4.21) is finite, $\hat{s}$ is still a random variable and, as a result, $\hat{R}_{yd}(\tau)$ is a random variable as well. Since the noise and interference have zero mean, then the mean of (4.29) is based on the desired component and the variance is the noise power at a particular $\tau$. We have defined the noise and interference components as uncorrelated with the reference signal, $d(t)$, and it is thereby independent of $\tau$. Generalizing (4.9) to be a function of $\tau$, the SINR of the correlator output is then
given by

\[
\tilde{\rho}_0(\tau) = \frac{\left| E\{ \hat{R}_{yd}(\tau) \} \right|^2}{\text{var}\{ \hat{R}_{yd}(\tau) \}} \\
= \frac{|R_{yd}(\tau)|^2}{E\{ \left| \hat{R}^\text{w}_{yd}(\tau) \right|^2 \}} \\
= \frac{|s^H(\tau)w|^2}{E\{ |(\hat{s}_n(\tau) + \hat{s}_i(\tau))^Hw|^2 \}} \\
= \frac{w^Hs(\tau)s^H(\tau)w}{E\{w^H(\hat{s}_n + \hat{s}_i)(\hat{s}_n + \hat{s}_i)^Hw\}} \\
= \frac{w^Hw E\{(\hat{s}_n + \hat{s}_i)(\hat{s}_n + \hat{s}_i)^H\}w}{w^Hw}.
\]

If we define the respective correlation matrices of the post-correlation components as

\[
\Psi_d(\tau) = s(\tau)s^H(\tau) \quad (4.30)
\]

\[
\Psi_n = E\{\hat{s}_n\hat{s}_n^H\} \quad (4.31)
\]

\[
\Psi_i = E\{\hat{s}_i\hat{s}_i^H\} \quad (4.32)
\]

as well as the undesired component \(\Psi_u\) and total correlation matrix \(\Psi\)

\[
\Psi_u = \Psi_n + \Psi_i \quad (4.33)
\]

\[
\Psi = \Psi_d(\tau) + \Psi_n + \Psi_i \quad (4.34)
\]

then the post-correlation SINR can be written as

\[
\tilde{\rho}_0(\tau) = \frac{w^H\Psi_d(\tau)w}{w^H\Psi_u w} \quad (4.35)
\]

The undesired correlation matrices \(\Psi_n\) and \(\Psi_i\) have simple interpretations that can be understood from (4.14). The two matrices have forms similar to (3.42) and (3.43) except with an additional response, \(G_d(f)/T\), which represents the effect of correlation.
with the reference signal. Combining (4.35) and (4.17) yields a simple vector form for $C/N_0$

$$C/N_0(\tau) = \frac{1}{T} \frac{w^H \Psi(\tau)w}{w^H \Psi w}$$

(4.36)

which has the same form as (4.3). This form is convenient for optimizing the filter since the filter weights are clearly separated in the equation.

The correlation matrix $\Psi$ can be thought of as being composed of $L \times L$ submatrices

$$\Psi = \begin{bmatrix} \Psi_{11} & \cdots & \Psi_{K1} \\ \vdots & \ddots & \vdots \\ \Psi_{1K} & \cdots & \Psi_{KK} \end{bmatrix}$$

(4.37)

where each submatrix $\Psi_{rs}$ is the correlation between the taps of the $r$th and $s$th elements. The $(p,q)$ entry of the $(r,s)$ submatrix for each component are

$$[\Psi_{n,rs}]_{pq} = \begin{cases} \frac{1}{T} \int G_d(f) \bar{S}_n(f) |F_r(f)|^2 e^{-j2\pi (q-p)T_0} df & r = s \\ 0 & r \neq s \end{cases}$$

(4.38)

$$[\Psi_{i,rs}]_{pq} = \frac{1}{T} \int G_d(f) \left( \sum_{m=1}^{M} \bar{S}_{i,m}(f) F_{r\star}(f) F_s(f) A_{r\star}(f, \theta, \phi) A_s(f, \theta, \phi) \right) e^{-j2\pi (q-p)T_0} df$$

(4.39)

This provides a means to analytically evaluate the correlation matrices.

4.3 Adaptive Filtering Algorithm for Maximum $C/N_0$

In adaptive filtering literature, the algorithm that maximizes SINR is well known and provides a useful bound with which to objectively compare the performance of different STAP algorithms [19]. However, since post-correlation performance is a more appropriate metric for GNSS receiver systems, it would be preferable to develop an algorithm that maximizes $C/N_0$ instead. In this section, we will derive this algorithm
using two different approaches. Both approaches serve to illuminate important aspects of our optimal algorithm.

**Derivation 1: Eigenvector Approach**

In this section, the weights which maximize $C/N_0$ will be derived using an eigenvector-based approach. This derivation is nearly identical to the well-known derivation of the maximum SINR algorithm [19]. Our goal is to find the weights $w_0$ which maximize $C/N_0$ given in (4.36)

$$w_0(\tau) = \arg\max_w \frac{1}{T} \frac{w^H \Psi_d(\tau) w}{w^H \Psi_u w}$$ (4.40)

Note that the optimal weights depend on the relative delay $\tau$ at which the $C/N_0$ is to be maximized. Since the weights can be scaled by an arbitrary complex constant without affecting the $C/N_0$, we can maximize the fraction by holding the numerator constant while minimizing the denominator

$$w_0(\tau) = \arg\min_w \frac{w^H \Psi_u w}{w^H \Psi_d(\tau) w} \quad \text{s.t.} \quad w^H \Psi_d(\tau) w = 1$$ (4.41)

where the factor $1/T$ is inconsequential. This is a standard optimization problem which has a quadratic form with a quadratic constraint. Using the method of Lagrange multipliers, we define the cost function $J$,

$$J = w^H \Psi_u w + \lambda \left( w^H \Psi_d(\tau) w - 1 \right)$$ (4.42)

and take the gradient with respect to $w$,

$$\nabla_w J = \Psi_u w + \lambda \Psi_d(\tau) w$$ (4.43)
A critical point occurs when (4.43) is equal to zero

\[ \Psi_u w + \lambda \Psi_d(\tau)w = 0 \]  \hspace{1cm} (4.44)

\[ w + \lambda \Psi_u^{-1} \Psi_d(\tau)w = 0 \]  \hspace{1cm} (4.45)

\[ \Psi_u^{-1} \Psi_d(\tau)w = -\frac{1}{\lambda} w \]  \hspace{1cm} (4.46)

which follows since \( \Psi_u \) is invertible. This is a standard eigenvalue equation, and the weight vector which satisfies this equation is an eigenvector of \( \Psi_u^{-1} \Psi_d(\tau) \). The Lagrange multiplier \( \lambda \) is chosen to satisfy the condition in (4.41); however, this choice does not affect the \( C/N_0 \). Rather, in order to minimize \( w^H \Psi_u w \), \( w_0 \) is chosen to be

\[ w_0(\tau) = \mathcal{P}\{\Psi_u^{-1} \Psi_d(\tau)\}, \]  \hspace{1cm} (4.47)

where \( \mathcal{P}\{\} \) is an operator which returns the principal eigenvector. Since this derivation is identical in form to the derivation of the maximum pre-correlation SINR algorithm, the maximum \( C/N_0 \) weights can be understood as the maximum SINR algorithm applied to the “post-correlation” correlation matrices.

In practice, there is no need to directly solve the eigenvalue problem in (4.47) to find the optimal filter weights. Rather, by simplifying the equation, the weights are shown to be equivalent to

\[ w_0 = \Psi_u^{-1} s(0), \]

\[ w_0(\tau) = \Psi_u^{-1} s(\tau), \]

where \( s \) is the reference correlation vector defined in Equation 4.26. To show this, first note that the matrix \( \Psi_u^{-1} \Psi_d \) is Hermitian and positive semi-definite. Since \( \Psi_d \) has the form \( \Psi_d(\tau) = s(\tau)s^H(\tau) \), it follows that both \( \Psi_d \) and \( \Psi_u^{-1} \Psi_d \) are rank 1. Therefore,
\( \Psi_u^{-1}\Psi_d \) has only 1 non-zero eigenvalue which is also the maximum eigenvalue, and the weights can be put into the form

\[
\begin{align*}
\lambda w_0 &= \Psi_u^{-1}s(\tau)s^H(\tau)w_0 \quad (4.48) \\
\lambda w_0 &= \Psi_u^{-1}s(\tau)(s^H(\tau)w_0) \quad (4.49) \\
w_0 &= \left( \frac{s^H(\tau)w_0}{\lambda} \right) \Psi_u^{-1}s(\tau) \quad (4.50) \\
w_0 &= \mu \Psi_u^{-1}s(\tau), \quad (4.51)
\end{align*}
\]

where \( \mu \) is some complex scalar. The scalar does not affect the C/N\(_0\) performance of the adaptive filter since scaling the filter scales all components equally. The filter, \( w_0 \), is chosen at a particular \( \tau \), which optimizes the C/N\(_0\) for that delay.

Note that correlation with the reference signal causes the wideband satellite signal to have a rank 1 correlation matrix \( \Psi_d \). This property, used in the above conclusion, is the same used to define the maximum pre-correlation SINR weights for CW desired signals. This is a useful connection since there has been extensive adaptive filter research which involves CW desired signals. As a result, much of the mathematics used to analyze the SINR of CW desired signals will analogously apply to the post-correlation SINR of wideband signals in GNSS.

**Derivation 2: Minimum MSE Approach**

An alternative derivation of the optimal C/N\(_0\) weights follows from the minimization of mean-squared error (MSE) on the cross-correlation function. Consider the error \( \epsilon \) between the ideal cross-correlation function and the estimated one at a particular delay, \( \tau \),

\[
\begin{align*}
\epsilon(\tau) &= R_{dd}(\tau) - \hat{R}_{yd}(\tau) \\
&= R_{dd}(\tau) - s^H(\tau)w. \quad (4.53)
\end{align*}
\]
Let us define the cost function $J$ as the mean square error

$$ J = E \{ |\epsilon(\tau)|^2 \} $$

$$ = |R_{dd}(\tau)|^2 + w^H \Psi w - 2 \text{Re}\{R_{dd}(\tau)s^H(\tau)w\}.$$  \hspace{1cm} (4.54)

Taking the gradient with respect to the weights

$$ \nabla_w J = 0 + 2\Psi(\tau)w - 2R_{dd}(\tau)s(\tau) $$

$$ \hspace{1cm} \text{and setting the result equal to zero} $$

$$ 2\Psi(\tau)w_0 - 2R_{dd}(\tau)s(\tau) = 0 $$

$$ w_0 = R_{dd}(\tau)\Psi^{-1}(\tau)s(\tau). $$

Comparing (4.58) with (4.51), we observe that they are very similar. In fact, they are the same. To show it, this form can be further reduced by following the steps in [18, pp. 55]. First, to total correlation matrix is given by

$$ \Psi(\tau) = \Psi_u + \mu s(\tau)s^H(\tau). $$

Since $\Psi_u$ is nonsingular, the matrix inversion lemma leads to

$$ \Psi^{-1}(\tau) = \Psi_u^{-1} + \beta \Psi_u^{-1}s(\tau)s^H(\tau)\Psi_u^{-1}, $$

where

$$ \beta = \frac{\mu}{1 + \mu s^H(\tau)\Psi_u^{-1}s(\tau)}. $$

Multiplying both sides by $s(\tau)$ yields,

$$ \Psi^{-1}(\tau)s(\tau) = \Psi_u^{-1}s(\tau) + \beta \Psi_u^{-1}s(\tau)s^H(\tau)\Psi_u^{-1}s(\tau) $$

$$ = (1 + \beta s^H(\tau)\Psi_u^{-1}s(\tau))\Psi_u^{-1}s(\tau). $$
So $\Psi^{-1}s(\tau)$ is just a scalar multiple of $\Psi^{-1}_u s(\tau)$, and Equation 4.58 is equivalent to the optimal $C/N_0$ weights

$$w_0(\tau) = \Psi^{-1}s(\tau). \quad (4.64)$$

Therefore, maximizing $C/N_0$ minimizes the mean-square error at a particular $\tau$ on the cross-correlation function. Since the error corresponds to noise on the cross-correlation estimate, it makes sense that maximizing the post-correlation SINR at that delay would minimize that error.

### 4.4 Example Results

Simulations were performed to study the optimal algorithm proposed in this chapter. The example results are divided into two subsections. In this first subsection, the behavior of the optimal adaptive filter algorithms that lead to maximum SINR and maximum $C/N_0$ will be compared. It will be shown that an algorithm that leads to maximum SINR will not necessarily lead to maximum $C/N_0$. In the second subsection, simulations will compare SINR and $C/N_0$ performance of three common STAP algorithms in the presence of interference. It will be shown that SINR can be a deceiving metric when used to compare their relative performance. Furthermore, it will be shown that some algorithms in use today come very close to the upper performance bound.

The antenna array used in the simulations is a uniform linear array of 7 isotropic elements with half-wavelength spacing as depicted in Fig. 4.1. Both the system bandwidth and the STAP filter length will be varied. The front-end thermal noise is white, zero-mean Gaussian with a fixed dB/Hz. The front-end filters are assumed flat and equalized for all elements. The sampling rate was chosen to match the
system bandwidth. The SOI is a binary phase-shift keyed (BPSK) signal with 10MHz chipping rate which will give it a sinc-squared spectrum which corresponds to a GPS P-code signal. It has a power of -30 dB SNR at 24 MHz system bandwidth. It is incident on the array at 50 deg. offset from broadside. The integration time is 10ms. Note that the conclusions of this chapter will apply generally to any antenna array.

4.4.1 Comparison of the Optimal SINR and Optimal C/N₀ Algorithms

This section will illustrate the different behaviors of the optimum SINR (Max SINR) and optimum C/N₀ (Max C/N) algorithms. In this case, no interference is present, and the adaptive array is merely reacting to the thermal noise and SOI.

First, the effects of increasing the STAP filter length are observed. In this case, the system bandwidth was chosen to be 24MHz. Fig. 4.2 (left) shows pre-correlation SINR performance vs STAP filter length for the two algorithms. In this case, the Max SINR algorithm defines the performance bound and exhibits increasing SINR vs. filter length. This is contrary to the Max C/N algorithm, which appears independent of filter length. If one were to only observe this figure, it would be easy to conclude that

![Uniform Linear Array](image)

*Figure 4.1: The uniform linear array of 7 isotropic elements used for simulations in this study.*
the Max C/N algorithm is some 4 dB worse than the Max SINR algorithm. However, if instead we observe the post-correlation SINR performance, as shown in Fig. 4.2 (right), a very different trend emerges. In this case, the Max SINR performance actually decreases as the filter length is increased. This behavior is very counter-intuitive, and we would expect that, in the case of no interference, performance should be independent of STAP filter length, much like the Max C/N algorithm exhibits. To find the source of this stark difference between these two algorithms, we will observe their adaptive antenna response. Fig. 4.3 (left) shows the magnitude of the adaptive antenna response in the SOI direction using the Max SINR algorithm. The adaptive antenna response becomes increasingly narrow as the number of taps increase. This is due to the power spectral densities of the noise and SOI. The power spectrum of the noise is flat while the SOI power spectrum has a sinc-squared shape. As a result, to maximize pre-correlation SINR is to implement a filter that emphasizes the peak of the SOI spectrum. The narrowing of the response using the Max SINR algorithm causes a steady increase in SINR; however, the narrowing of the response is detrimental to post-correlation SINR performance, which steadily decreases with increasing taps. Fig. 4.3 (right) shows adaptive antenna responses using the Max C/N algorithm. All of the responses are the same and are completely flat. It is this behavior which results in such different performance when observing either pre-correlation SINR or C/N₀ performance. In this scenario with isotropic elements and white thermal noise, to maximize C/N is to not form any temporal response. This will maximize the processing gain of the coherent integration performed by the receiver. In this sense, only a single tap is needed for optimal performance in this case. Yet, this fact is not obvious from observing pre-correlation SINR.
Now the STAP filter length will remain fixed, and the effects of increasing the system bandwidth will be considered. In this case, the STAP filter length is 7-taps; however, the sampling rate varies according to the system bandwidth. Also, as the system bandwidth increases, the total thermal noise power increases since it has a constant dB/Hz. Fig. 4.4 (left) shows the plot of pre-correlation SINR versus system bandwidth. Although the noise power increases by 3 dB as the system bandwidth doubles, the Max SINR algorithm shows only a 0.5 dB decrease in performance. Clearly, it is able to suppress the excess noise. The Max C/N algorithm, on the other hand, exhibits the 3 dB performance drop. Furthermore, it reaches a -6 dB performance difference when compared to the Max SINR performance. Once again, however, the post-correlation SINR performance, as shown in Fig. 4.4 (right), shows a very different result. In terms of post-correlation SINR, the Max C/N algorithm has constant performance and is actually 1-2 dB better than the Max SINR algorithm. The reason for this is that the higher sampling rate and the constant integration time results in more samples being integrated. This cancels the effect of the increasing noise power. Also, since the Max SINR algorithm is able to filter the additional noise, this results in increasing performance with increasing system bandwidth. Fig. 4.5 (left) shows the magnitude of the adaptive antenna response in the SOI direction for the Max SINR algorithm. From the figure, it is clear that the Max SINR algorithm forms a narrow frequency response around the carrier frequency. As the bandwidth increases, the response is stretched, but its overall shape remains the same. Fig. 4.5 (right) shows the same case using the Max C/N algorithm. Again, it has a flat response for all system bandwidths.
These results demonstrate that there is a significant difference between optimizing STAP filters for pre-correlation SINR performance versus optimizing them for receiver C/N₀.

4.4.2 Comparison of SINR and C/N₀ Performance of Common Adaptive Filter Algorithms

This section presents simulation results comparing pre-correlation and post-correlation SINR of common adaptive filter algorithms in interference scenarios. Two different interference scenarios will be considered. The first interference scenario has a single strong, wideband interferer. The second interference scenario is much harsher and contains a wideband interferer and six narrow band interferers. In all cases the interference is represented by stationary Gaussian noise with a given power spectral density and incident from a given direction. By comparing the performance of these algorithms to the maximum C/N₀ bound, it will be shown that many weighting algorithms come very close to achieving optimal performance.

We start with the performance of various weighting algorithms in the presence of a single interference signal. In this case, the angle-of-incidence of the SOI was swept from θ = −90° to 90°. The definition of θ is depicted in Fig. 4.1. The single interfering signal is zero mean Gaussian with a flat, 20 MHz bandwidth and a power 40dB above the noise floor. It is incident from θ = 60°. The STAP system has 7 taps and the system bandwidth is 24 MHz. Since the interference bandwidth is of comparable size to the system bandwidth, this is a wideband interference scenario. The angle of the interferer was chosen so that one can observe the angular size of the performance degradation formed around the interferer. Fig. 4.6 shows the pre-correlation SINR performance of different STAP algorithms. As expected, the Max
Figure 4.2: Pre-Correlation SINR (left) and post-correlation SINR (right) vs. STAP filter length using the Max SINR and Max C/N algorithms.

Figure 4.3: Adaptive antenna system response magnitude in SOI direction for different STAP filter lengths using the Max SINR algorithm (left) and maximum C/N₀ algorithm (right). Note that, in the absence of interference, all of the responses on the right are the same regardless of filter length.
Figure 4.4: Pre-Correlation SINR (left) and post-correlation SINR (right) vs. System Bandwidth using the Max SINR and Max C/N algorithms.

Figure 4.5: Adaptive antenna system response magnitude in SOI direction for different system bandwidths using the Max SINR algorithm (left) and the maximum C/N₀ algorithm (right). Note that, in the absence of interference, all of the responses on the right are the same regardless of system bandwidth.
SINR method achieves the best performance. The MMSE method is consistently 0.5 dB below the optimum performance. It is interesting to note that the Max C/N and simple beam steering methods appear to perform poorly, being approximately 4 dB below optimum performance. As expected, the simple power minimization method performs the worst since it does not provide beam forming. We also note that its drop in SINR has a significantly wider angular region around the interferer than the other algorithms.

Fig. 4.7 shows the post-correlation SINR performance. Using this performance metric, the relative performances of the different algorithms have been rearranged. Now, the Max C/N weights define the optimum performance, which is achieved using the simple beam steering method. Again, MMSE is within 0.5 dB of optimum performance. However, the Max SINR algorithm performs significantly worse than the others. Although the simple power minimization algorithm still performs the worst, we see that on average it is only 5 dB below optimal performance. This contradicts its pre-correlation SINR performance in which it was measured to be 10 dB worse. From these results, it is clear that there is a significant difference between pre-correlation and post-correlation SINR performance.

To understand why there is such a discrepancy between the pre-correlation and post-correlation SINR, we will first consider the processing gain. In this case, the processing gain is defined as the ratio of the post-correlation SINR to the pre-correlation SINR. Fig. 4.8 shows the processing gain for different STAP algorithms. It is clear that the processing gain for the Max SINR and MMSE algorithms are worse than the other methods. The reason for this is evident in Fig. 4.9. The figure shows the normalized power spectral density of the noise at the output of the adaptive
antenna. The plots for simple beam steering and power inversion overlap. It is clear that the noise at the output is not necessarily white. The MMSE method produces an approximately matched noise spectrum. The Max SINR method produces a very narrow noise spectrum. Although these matched and narrowed spectra improve the pre-correlation SINR performance, they are detrimental to the processing gain. This trade-off is what accounts for the discrepancy between pre-correlation and post-correlation SINR performance.

The next set of figures show the performance in the presence of 7 interferers. The strong wideband interferer from the previous scenario is still present, but 6 narrowband interferers have been added. The narrowband interferers are zero mean Gaussian with a flat, 2 MHz bandwidth and a power 40 dB above the noise floor. They are centered within the system bandwidth, have incident angles distributed uniformly in azimuth and elevation angles within the range $75^\circ < \theta < 90^\circ$. The STAP system has 7 taps and the system bandwidth is 24 MHz. This scenario has as many interferers as antenna elements, and the STAP system is stressed both spatially and temporally. This scenario was chosen to determine whether the conclusions drawn from the results in the previous paragraph are more generally applicable.

Fig. 4.10 shows the pre-correlation SINR performance, and Fig. 4.11 shows the post-correlation SINR. Again, the angle-of-incidence of the SOI was swept from $\theta = -90^\circ$ to $90^\circ$. Comparing these results to Figs. 4.6 and 4.7 reveal some noticeable differences. Namely, the Max SINR and Max C/N algorithms have much more comparable performance in this case. However, we also notice that, just as in the single interferer case, the relative performance of the simple beam steering and
simple power minimization algorithms are different based on which performance metric is used. The simple beam steering algorithm has SINR performance 3 dB below optimum, yet it still effectively achieves optimum C/N performance.
Figure 4.6: Pre-correlation SINR performance of different STAP algorithms in the presence of an interfering signal.

Figure 4.7: Receiver post-correlation SINR performance of different STAP algorithms in the presence of an interfering signal. DC, MMSE and Max C/N methods have near identical performance.
Figure 4.8: Processing gain of different STAP algorithms.

Figure 4.9: Normalized power spectral density of the noise at the output of the adaptive antenna for different STAP algorithms when the SOI is incident from $(\theta, \phi) = (0, 0)$. 

76
Figure 4.10: Pre-correlation SINR performance of different STAP algorithms in the presence of 7 interfering signals. MMSE, Max SINR, and Max C/N methods have very similar performance.

Figure 4.11: Receiver post-correlation SINR performance of different STAP algorithms in the presence of 7 interfering signals. DC and Max C/N methods have very similar performance.
4.5 Summary & Conclusions

This chapter developed an adaptive filter algorithm that maximizes $C/N_0$. It achieves the optimal interference suppression for a GNSS receiver with an adaptive antenna. Furthermore, this chapter demonstrated that there are significant differences between pre-correlation SINR and receiver $C/N_0$ performance for different adaptive algorithms. It was shown that adaptive filter algorithms which optimize SINR behave very differently than those which optimize $C/N_0$. As a result, the optimum SINR algorithm exhibited poor $C/N_0$ performance in some cases. It is clear that the relative performance of the different algorithms changed noticeably depending on whether SINR or $C/N_0$ was used as the performance metric.
CHAPTER 5

Novel Adaptive Filtering Algorithm for GNSS Adaptive Antennas

As demonstrated in Chapter 3, adaptive antenna arrays have the potential to introduce distortion into received GNSS signals, especially during interference suppression. This distortion causes bias errors in the code delay and carrier phase measurements made by the GNSS receiver. For precision GNSS applications, these antenna-induced biases are a significant source of error. Although Chapter 4 developed an algorithm to optimally suppress interference, it does not prevent antenna-induced errors. This chapter derives a novel adaptive filtering algorithm for GNSS antenna arrays which is specifically designed to produce zero antenna-induced biases while simultaneously maximizing $C/N_0$. This is achieved in scenarios both with and without interference, with any length STAP filter, and irregardless of the antenna array used. These qualities represent the optimal behavior for an adaptive antenna in GNSS applications. Most importantly, the proposed algorithm has implementation requirements comparable to algorithms in use today. Section 5.1 begins by reviewing antenna-induced biases and alternative approaches that have been proposed. Section 5.2 derives mathematical conditions that the adaptive filter weights must satisfy in order to produce zero bias, while Section 5.3 uses these conditions to extend the optimal algorithm of
Chapter 4. Section 5.4 provides some detail on how the modified algorithm would be implemented, and Section 5.5 shows example simulation results. These results demonstrate that our new algorithm achieves C/N\textsubscript{0} performance that approaches the theoretical upper bound while guaranteeing zero bias errors.

5.1 Conditions for Zero Antenna-Induced Bias

In precision GNSS navigation applications, antenna-induced bias errors contribute a large portion of the total error in the ranging signal time-of-arrival (TOA) estimate [23]. As discussed in Chapter 3, this distortion manifests itself as code delay biases and carrier phase biases in the receiver measurements. Since antenna-induced biases are direction-dependent, they ultimately lead to errors in the final navigation solution.

This section defines the mathematical conditions that the adaptive filter \( \mathbf{w} \) must satisfy in order to prevent antenna-induced biases in the carrier phase and code delay measurements made by a GNSS receiver. It will be shown that these zero-bias conditions can be written as a pair of linear equations. Note that a convention adopted by this paper is that when the correlation vector \( \mathbf{s}(\tau) \) from Chapter 3 or its derivative is evaluated at \( \tau = 0 \), those will be written as \( \mathbf{s} \) and \( \dot{\mathbf{s}} \); i.e.,

\[
\mathbf{s} = \mathbf{s}(0) \quad (5.1)
\]

\[
\dot{\mathbf{s}} = \dot{\mathbf{s}}(0) = \frac{d}{d\tau}\mathbf{s}(\tau)\bigg|_{\tau=0}, \quad (5.2)
\]

instead of explicitly writing “(0)”. This will simplify the notation considerably. The repeated appearance of \( \mathbf{s} \) is indicative of its significance, and it is commonly referred to as the reference correlation vector.
If the carrier phase bias is zero, then

$$\psi_0 = 0 \quad (5.3)$$

$$\angle R_{yd}(0) = 0 \quad (5.4)$$

$$\angle s^H w = 0 , \quad (5.5)$$

and the conditions that the filter weights $w$ must satisfy for zero carrier phase bias are

$$\text{Im}\{s^H w\} = 0 \quad (5.6)$$

$$\text{Re}\{s^H w\} > 0 . \quad (5.7)$$

To find the condition for zero code delay bias, we begin with the magnitude squared of the cross-correlation function

$$P_d(\tau) = |R_{yd}(\tau)|^2 \quad (5.8)$$

$$= s^H(\tau)ww^H s(\tau) , \quad (5.9)$$

and its derivative

$$\frac{d}{d\tau} P_d(\tau) = s^H(\tau)ww^H s(\tau) + s^H(\tau)ww^H \dot{s}(\tau) , \quad (5.10)$$

$$= 2 \text{Re}\left\{\dot{s}^H(\tau)ww^H s(\tau)\right\} . \quad (5.11)$$

In order for the code delay bias to be zero, the weights are chosen so that they shift the peak of the cross-correlation function to $\tau = 0$. That is,

$$\frac{d}{d\tau} P_d(\tau) \bigg|_{\tau=0} = 0 . \quad (5.12)$$

Combining (5.11) and (5.12) yields a condition for zero code delay bias in terms of the filter weights,

$$\text{Re}\left\{\dot{s}^H w w^H s\right\} = 0 . \quad (5.13)$$
If the zero phase bias conditions in (5.6) and (5.7) are satisfied, then (5.13) can be simplified to

\[
\text{Re} \left\{ s^Hww^Hs \right\} = 0 \quad (5.14)
\]

\[
\text{Re} \left\{ s^Hw \right\} (w^Hs) = 0 \quad (5.15)
\]

\[
\text{Re}\{s^Hw\} = 0, \quad (5.16)
\]

and all three conditions can be written as

\[
\text{Im}\{s^Hw\} = 0 \quad (5.17)
\]

\[
\text{Re}\{s^Hw\} > 0 \quad (5.18)
\]

\[
\text{Re}\{s^Hw\} = 0. \quad (5.19)
\]

These are the conditions that the adaptive filter must jointly satisfy in order to achieve both zero carrier phase bias and zero code delay bias. These three conditions can be reduced to two conditions

\[
s^Hw = 1 \quad (5.20)
\]

\[
\dot{s}^Hw = \alpha i \quad \alpha \in \mathbb{R}, \quad (5.21)
\]

where \( i \) is the unit imaginary and \( \alpha \) is a real number. Since the weights \( w \) can be scaled by an arbitrary real value without affecting the bias or \( C/N_0 \), we are able to normalize the weights to \( s^Hw = 1 \) in (5.20) without limiting our approach. Finally, (5.20) and (5.21) can be written as

\[
C^Hw = f \quad (5.22)
\]

where

\[
C = [ s \ \dot{s} ], \quad f = [ 1 \ \alpha i ]^T. \quad (5.23)
\]
Any real value of $\alpha$ is acceptable. Ultimately, our goal will be to find the adaptive filter $w$ which satisfies the zero-bias condition (5.22) while maximizing $C/N_0$. Our choice of $\alpha$ will be a degree of freedom in this optimization process.

Note that the derivative condition in (5.12) is technically not enough to guarantee that the peak is at $\tau = 0$. A zero derivative could very well correspond to a minimum, or higher peaks could be elsewhere on the cross-correlation function. However, our proposed adaptive filter algorithm derived in the proceeding sections will have filter weights that satisfy the zero derivative condition while simultaneously maximizing $C/N_0$. In this way, it is effectively ensured that this delay will correspond to the peak. Although the adaptive filter might also affect the sidelobe level of the cross-correlation function, this algorithm will not address this aspect.

### 5.2 Adaptive Filtering with Bias Constraints

This section derives a novel adaptive filtering algorithm which optimizes interference suppression (maximizes $C/N_0$) while satisfying the conditions that guarantee no antenna-induced biases are introduced into the GNSS receiver measurements.

Let us first find a filter $w_0$ that maximizes $C/N_0$. Recall from Chapter 4 that the $C/N_0$ can be written as

$$C/N_0 = \frac{1}{T} \frac{w^H \Psi_d(\tau_0) w}{w^H \Psi_u w}$$

(5.24)

where the correlation matrices of the desired and undesired correlation vector components were defined in (4.30) and (4.33) as

$$\Psi_d(\tau) = s(\tau) s^H(\tau)$$

(5.25)

$$\Psi_u = E\left\{ \hat{s}_u \hat{s}^H_u \right\} .$$

(5.26)
Note that the weights $\mathbf{w}$ in (5.24) can be scaled arbitrarily without affecting the $C/N_0$. Consequently, the numerator in (5.24) can be constrained to be 1, while the denominator is minimized

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}} \mathbf{w}^H \Psi_u \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^H \Psi_d(0) \mathbf{w} = 1 \quad (5.27)$$

This is the same approach more commonly used to maximize SINR. Note that we have chosen to optimize $C/N_0$ for the case when the incident and reference signals are synchronized ($\tau = 0$). Combining (5.25) with (5.27) yields

$$\mathbf{w}_0 = \arg\min_{\mathbf{w}} \mathbf{w}^H \Psi_u \mathbf{w} \quad \text{s.t.} \quad \mathbf{s}^H \mathbf{w} = 1 \quad . \quad (5.28)$$

Observe that this condition is identical to the first condition in (5.22). Now, it should be obvious that if we want to maximize $C/N_0$ subject to the zero-bias conditions in (5.22), then we need only to add one additional condition,

$$\{\mathbf{w}_0, \alpha_0\} = \arg\min_{\mathbf{w}, \alpha} \mathbf{w}^H \Psi_u \mathbf{w} \quad \text{s.t.} \quad \mathbf{C}^H \mathbf{w} = \mathbf{f} \quad (5.29)$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{s} & \mathbf{s} \end{bmatrix}, \quad \mathbf{f} = [1 \, \mathbf{i} \alpha]^T, \quad \alpha \in \mathbb{R} \quad . \quad (5.30)$$

which creates an optimization problem over both $\mathbf{w}$ and $\alpha$.

**Part I. Optimizing $\mathbf{w}$**

The partial optimization involving only the weights parameter $\mathbf{w}$ has the form of the well known linearly constrained minimum variance (LCMV) optimum adaptive filter. In this context, $\mathbf{C}$ is known as the constraint matrix and $\mathbf{f}$ is known as the gain vector. That standard solution from [19, Chp. 6] is repeated below.

Using the method of Lagrange multipliers, we will minimize the cost function $J$,

$$J \triangleq \mathbf{w}^H \Psi_u \mathbf{w} + (\mathbf{w}^H \mathbf{C} - \mathbf{f}^H) \lambda + \lambda^H (\mathbf{C}^H \mathbf{w} - \mathbf{f}) \quad (5.31)$$
where \( \lambda \) is a 2x1 vector since there are two constraints. The complex gradient of \( J \) taken with respect to \( w \) is

\[
\nabla_w J = \Psi_u w + C\lambda,
\]

and setting it equal to zero yields

\[
0 = \Psi_u w + C\lambda \tag{5.33}
\]

\[
w = -\Psi_u^{-1}C\lambda. \tag{5.34}
\]

Substituting (5.34) into the condition in (5.29) allows us to solve for \( \lambda \),

\[
C^Hw = f \tag{5.35}
\]

\[
-C^H\Psi_u^{-1}C\lambda = f \tag{5.36}
\]

\[
\lambda = -(C^H\Psi_u^{-1}C)^{-1}f, \tag{5.37}
\]

and substituting (5.37) into (5.34) produces the optimal weights \( w_0 \),

\[
w_0 = \Psi_u^{-1}C(C^H\Psi_u^{-1}C)^{-1}f \tag{5.38}
\]

which is the well-known solution to a LCMV adaptive filter. However, the gain vector \( f \) still contains an \( \alpha \) term that must be optimized.

**Part II. Optimizing \( \alpha \)**

To complete the optimization, we must now minimize over the gain vector parameter \( \alpha \). As explained in the previous section, any real value will prevent the adaptive antenna from introducing bias errors. Each value of \( \alpha \) corresponds to a unique gain vector \( f \) for which the filter (5.38) will maximize \( C/N_0 \).
First, the matrix inverse in (5.38) is analytically evaluated,

\[
(C^H \Psi^{-1} u C)^{-1} = \frac{1}{\gamma} \begin{bmatrix}
(s^H \Psi^{-1} u \hat{s}) & (-s^H \Psi^{-1} u \hat{s}) \\
(-s^H \Psi^{-1} u \hat{s}) & (s^H \Psi^{-1} u \hat{s})
\end{bmatrix}
\]

(5.39)

\[
= \begin{bmatrix}
\beta_{11} & \beta_{12} \\
\beta_{21} & \beta_{22}
\end{bmatrix},
\]

(5.40)

where \( \beta_{ij} \) correspond to elements of the matrix inverse, and we note that \( \beta_{11} \) and \( \beta_{22} \) are real. \( \gamma \) is the determinant,

\[
\gamma = ||C^H \Psi^{-1} u C||,
\]

(5.41)

however, it is inconsequential since it will drop out of our solution at the end. Next, we introduce the weight vectors \( w_0 \) and \( \dot{w}_0 \),

\[
w_0 = \Psi^{-1} s
\]

(5.42)

\[
\dot{w}_0 = \Psi^{-1} \dot{s}.
\]

(5.43)

Now, the filter weights (5.38) can be written in the form

\[
w = \Psi^{-1} u C (C^H \Psi^{-1} C)^{-1} f
\]

(5.44)

\[
= \begin{bmatrix} w_0 & \dot{w}_0 \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{12} \\
\beta_{21} & \beta_{22} \end{bmatrix} f
\]

(5.45)

\[
= \begin{bmatrix} w_0 & \dot{w}_0 \end{bmatrix} \begin{bmatrix} \beta_{11} + i\alpha\beta_{12} \\
\beta_{21} + i\alpha\beta_{22} \end{bmatrix}
\]

(5.46)

\[
= w_0 (\beta_{11} + i\alpha\beta_{12}) + \dot{w}_0 (\beta_{21} + i\alpha\beta_{22}).
\]

(5.47)

Next, \( \dot{w}_0 \) is decomposed into two orthogonal components

\[
\dot{w}_0 = \gamma w_0 + \dot{w}_\perp,
\]

(5.48)

and we can write (5.47) as

\[
w = w_0 (\beta_{11} + i\alpha\beta_{12} + \gamma\beta_{21} + \gamma i\alpha\beta_{22}) +
\]

(5.49)

\[
\dot{w}_\perp (\beta_{21} + i\alpha\beta_{22})
\]

(5.50)

\[
= c_1 w_0 + c_2 \dot{w}_\perp
\]

(5.51)
for two complex constants $c_1$ and $c_2$.

The weights $w_0$ in (5.42) correspond to the maximum C/N$_0$ algorithm. It was shown in Section 4.3 that C/N$_0$ can be understood as a mean-square error function which forms a quadratic performance surface as a function of the filter weights. $w_0$ represents a minimum of this surface. Therefore, to optimize performance is to minimize the deviation of $w$ from this optimum filter $w_0$ in (5.51). In other words, $\alpha$ should be chosen to minimize $c_2$. Choosing

$$\alpha = -\frac{\beta_{21}}{i\beta_{22}},$$

(5.52)

would cause $c_2$ to be zero. However, since $\alpha$ must be real, we can, at most, minimize the imaginary component of $c_2$ by choosing $\alpha$ to be

$$\alpha = \text{Re} \left\{ -\frac{\beta_{21}}{i\beta_{22}} \right\} = \text{Re} \left\{ -\frac{s^H\Psi^{-1}_u s}{s^H\Psi^{-1}_u s} \right\}.$$  

(5.54)

This yields the optimal solution $\alpha_0$,

$$\alpha_0 = \frac{\text{Im} \left\{ s^H\Psi^{-1}_u s \right\}}{s^H\Psi^{-1}_u s},$$

(5.55)

where, in going from (5.54) to (5.55), we have made use of the fact that $s^H\Psi^{-1}_u s$ is real since $\Psi_u$ is positive definite.

**Proposed Algorithm**

Equations (5.38), and (5.55) solve (5.29) and collectively represent our proposed adaptive filtering algorithm. There are two important things to note here. First, the zero-bias conditions are specifically derived to prevent errors in the GNSS receiver measurements. That is, the proposed adaptive algorithm does not attempt to prevent
general distortion to the received signal. Rather, it specifically focuses on antenna-induced biases. This is a significant departure from other approaches which attempt to minimize distortion across the received signal bandwidth by introducing a large number of constraints [50, 63]. The ability to prevent bias errors using only two constraints could be considered the primary contribution of this study, and it is the reason our approach is able to simultaneously prevent bias and achieve near optimal C/N$_0$.

Secondly, it should be emphasized that the proposed adaptive filter algorithm does not simply prevent bias errors caused by the filter itself, but rather the biases caused by the entire adaptive antenna system, which includes the physical antenna (with platform effects), the front-end, and the adaptive filters. If the antenna or the front-end electronics distort the received GNSS signals, then these effects are captured by the reference correlation vector and the adaptive filter will compensate for them. It is in this way that all antenna-induced biases are mitigated by our proposed algorithm. Other adaptive filtering algorithms have been proposed which focus only on preventing biases caused by the filter [31], and they represent only a partial solution.

5.3 Implementation

There are a variety of techniques available for the implementation of an LCMV filter of the form (5.38), and they are well documented [64–68]. However, we note that in conventional LCMV filtering, the gain vector $f$ is constant. Therefore, our proposed filter is unconventional in that the gain vector in (5.30) and (5.55) is adaptive. However, it is our opinion that this will not cause any significant barrier to
implementation since the values of f are both concise and formulated using terms found elsewhere in the algorithm. A detailed analysis of numerical stability and computational complexity of such a modification is beyond the scope of this paper.

The correlation matrix \( \Psi \) defined in (4.34) appears very different than the one in (3.26); however, it is straightforward to construct. From (4.23), we see that \( \hat{s}_u \) is formed by correlating the received signal snapshots with the reference signal. Since we have assumed that the interference and noise are uncorrelated with the reference signal, this is equivalent to filtering the noise and interference with a filter formed from the spectrum of the reference signal. This is clear based on (4.38) and (4.39), where the correlation matrix has changed to include the spectrum \( G_d(f) \) of the SOI.

Mathematically, we can construct \( \Psi \) using \( N \) filtered snapshots

\[
\hat{\Psi} \approx \sum_{n=1}^{N} \tilde{x}[n]\tilde{x}^H[n]
\]

(5.56)

where \( \tilde{x} \) are the filtered received signal snapshots. For example, if the SOI is the GPS P-code signal, than the filter should be sinc shaped. Since the SOI is very weak in GNSS applications, \( \hat{\Psi} \) will effectively contain only noise and interference components.

As part of our mathematical derivation, we assumed that the weights were in a steady state; however, this assumption is not required for the proposed algorithm to prevent bias errors. In a real implementation, finite sample effects in the correlation matrix will cause weight jitter and finite convergence time. Non-stationary interference could also prevent the weights from entering a steady state. Although these effects may degrade C/N\textsubscript{0} performance, the constraints in (5.22) will still be satisfied and the weights will not introduce bias errors into the GNSS receiver measurements.

Note that both the constraint matrix \( C \) and gain vector \( f \) depend on the reference correlation vector and its derivative. As mentioned earlier, these vectors can be
pre-calculated for each incident signal angle and stored in the adaptive antenna electronics. It is implied that control logic coordinates with the GNSS receiver in order to provide the adaptive filter with run-time information regarding the GNSS signal spectra and incident signal angles. Using attitude information provided by a sensor such as an inertial measurement unit (IMU), the adaptive processor would then load the corresponding reference correlation vectors from storage. These requirements are not extraordinary since the beamforming algorithms in use today in modern GNSS antenna arrays have comparable requirements [20, 21, 25, 26, 69]. The most difficult requirement is that the in-situ antenna array manifold must be known in order to pre-calculate the reference correlation vectors. In order to prevent antenna-induced biases to great precision, the antenna manifold must be known with corresponding precision.

5.4 Example Results

Simulations were performed to compare the performance of the proposed adaptive algorithm to commonly implemented existing STAP algorithms. Results show that the new algorithm successfully prevents antenna-induced bias under all interference scenarios and yields optimal interference suppression.

The antenna array that will be used is a 6-element circular array of patches mounted on an infinite ground plane shown in Fig. 3.3. A patch array was chosen since they are commonly used for GNSS applications; however, our proposed algorithm does not rely on any specific antenna array in order to work. The antenna pattern was simulated at the GPS L1 band (1.576 GHz) using the computational electromagnetics program FEKO [61]. The front-end thermal noise is assumed white,
zero-mean Gaussian. The front-end filters are assumed flat and equalized for all elements so that they are effectively neglected. The system bandwidth and the corresponding sampling rate were chosen to be 24MHz, and the STAP filter has 7 taps (unless otherwise noted). The SOI is a BPSK signal with 10.23 MHz chipping rate which will give it a sinc-squared power spectrum. The SOI has an incident signal power of -30 dB SNR at 24 MHz system bandwidth, and its angle-of-arrival will vary. Receiver performance metrics are evaluated using the analytic equations provided in previous sections, and the STAP filter weights will be analytically computed assuming the weights have converged to a steady state in a stationary signal scenario. In our coordinate system, $\phi$ is the azimuthal angle and $\theta$ is the elevation angle, where $\theta = 0$ corresponds to boresight.

The interference bandwidths, relative powers, and incidence angles are provided in Table 3.1. Wideband interferers occupy the entire system bandwidth while partial-band interferers have a narrower bandwidth and are offset from the L1 carrier frequency. All interferers are strong with a 50 dB interference-to-noise ratio (INR) and are incident from angles near the horizon. Three different scenarios will be simulated. The first has no interference, the second has 3 partial-band interferers, and the last contains 3 wideband and 3 partial-band interferers. Note that the optimal behavior of our proposed algorithm does not depend on any specific interference scenario or properties. The choice of offset partial-band interferers merely serves to emphasize the degradation of certain algorithms in use today.

Five different adaptive filtering algorithms will be studied, and they are listed in Table 5.1. The first three algorithms were introduced earlier in Chapter 3 and 4. The first is known as power inversion and is commonly implemented due to its simplicity.
This method only suppresses signals whose power is above the noise floor and does not provide beamforming. The second algorithm produces the weights which minimize the mean-square error between the adaptive antenna output and a reference signal. Figures will refer to this method as ‘MMSE’. Although this algorithm tends to yield good bias performance, there is nothing mathematically guaranteeing that the SOI will not be distorted. We will see that this algorithm will have significant bias errors in certain interference environments. The next algorithm yields optimal C/N₀, is labeled ‘Optimal’ in the figures, and was defined in Chapter 4. We shall see that, even though it yields optimal C/N₀, it does not prevent biases from being introduced into the receiver’s code delay measurements.

The next algorithm is the one proposed by this paper in (5.38). It maximizes C/N₀ subject to the zero-bias constraints and is given by

\[ \mathbf{w} = \Psi_u^{-1} \mathbf{C}(\mathbf{C}^H \Psi_u^{-1} \mathbf{C})^{-1} \mathbf{f} \]  \hspace{1cm} (5.57)

<table>
<thead>
<tr>
<th>Algorithm Label</th>
<th>Filter Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Inversion (PI)</td>
<td>( \mathbf{w} = \mu \Phi^{-1} \mathbf{u} )</td>
</tr>
<tr>
<td>MMSE</td>
<td>( \mathbf{w} = \mu \Phi^{-1} \mathbf{s} )</td>
</tr>
<tr>
<td>Optimal C/N₀</td>
<td>( \mathbf{w} = \mu \Psi_u^{-1} \mathbf{s} )</td>
</tr>
<tr>
<td>New Method (( \Psi ))</td>
<td>( \mathbf{w} = \Psi_u^{-1} \mathbf{C}(\mathbf{C}^H \Psi_u^{-1} \mathbf{C})^{-1} \mathbf{f} )</td>
</tr>
<tr>
<td>New Method (( \Phi ))</td>
<td>( \mathbf{w} = \Phi^{-1} \mathbf{C}(\mathbf{C}^H \Phi^{-1} \mathbf{C})^{-1} \mathbf{f} )</td>
</tr>
</tbody>
</table>
The previous section discussed how $\Psi_u$ can be estimated in a real implementation; however, we will also consider an approximation in which this matrix is replaced with the traditional correlation matrix $\Phi$

$$w = \Phi^{-1}C(C^H\Phi^{-1}C)^{-1}f$$

(5.58)

This reduces the implementation complexity, and the proceeding simulation results will quantify the $C/N_0$ trade-off of using this approximation. The algorithm in (5.57) will be labeled ‘New Method ($\Psi$)’ in the figures while the use of (5.58) will be labeled ‘New Method ($\Phi$)’.

In the first signal scenario, no interference is present. Fig. 5.1 shows the $C/N_0$, carrier phase bias, and code delay bias when the elevation angle $\theta$ of the SOI is varied from -90 to 90 degrees in the $\phi = 0$ plane. The power inversion method has suboptimal $C/N_0$ since it does not provide beam-forming. Furthermore, it has significant bias errors in both carrier phase and code delay. However, all of the other algorithms achieve zero carrier phase bias, zero code delay bias, and have near optimal $C/N_0$ performance. This serves as our baseline performance.

The second signal scenario has 3 strong, partial-band interferers. Fig. 5.2 shows the resulting $C/N_0$, carrier phase bias, and code delay bias vs. angle. Again, the power inversion method produces suboptimal $C/N_0$ performance. Its antenna-induced biases have also increased substantially. The MMSE and Optimal methods both have significant antenna-induced code delay biases. This demonstrates that optimizing the $C/N_0$ or minimizing the mean square error do not automatically prevent signal distortion. Only the new method proposed in this study achieves both optimal $C/N_0$ and zero bias error. Note that, in this case, our new method has similar performance whether $\Phi$ or $\Psi_u$ is used.
The third signal scenario has 3 wideband interferers in addition to the 3 partial-band interferers from the previous scenario. Fig. 5.3 shows the resulting $C/N_0$, carrier phase bias, and code delay bias vs. angle. These results are similar to the previous scenario; however, there is one important difference. Now, the only algorithm that achieves optimal $C/N_0$ is our new method using the $\Psi_u$ correlation matrix. When the $\Phi$ matrix is used in our new method, its performance is more than 1 dB-Hz lower. Although this is a small difference, we will see in later results that its suboptimal performance is dependent on the STAP filter length.

Next, we are interested in verifying the performance of these algorithms over more than just a single elevation cut and for many different interference configurations. To do this, Monte Carlo simulations were performed for multiple interference scenarios containing a specific number and type of interferers. In the first case, 3 partial-band interferers are present. In the second, 3 wideband interferers are present. In the third, 3 partial-band and 3 wideband are used in order to create a harsh scenario. The incident angles of these interferers were varied over low elevation angles. Over all scenarios, the performance measure is the percent of the upper hemispherical coverage which achieved a given $C/N_0$. The cumulative distribution of this performance is given in Fig. 5.4. For each SINR value, the corresponding percentage represents the amount of upper hemispherical area which achieves at least that SINR. The results verify our conclusions made from the previous figures, which were limited to a single elevation cut. Most importantly, these results show that the new algorithm (using $\Psi$) provides optimal $C/N_0$ performance over a wide range of interference scenarios. When $\Phi$ is used with the new algorithm, there is only a minor degradation in performance.
Next, we would like to understand the effects of STAP filter length on these results. Previously, we had only considered the case of 7-tap STAP. Monte Carlo simulations were performed for the scenario with 3 partial-band and 3 wideband interferers. The STAP filter length was varied from 3, 5, and 9 taps. Fig. 5.5 shows the resulting distribution of $C/N_0$. In all three cases, the new method using the $\Psi_u$ correlation matrix achieves optimal $C/N_0$ performance. Even though it is not shown, the new method also produces zero antenna-induced biases over all scenarios. Additionally, when the $\Phi$ matrix is used with our new method, the amount of sub-optimality in $C/N_0$ is greatly dependent on the STAP filter length. We conclude that, if a STAP filter length of 7 or greater is being used, than the $\Phi$ matrix can be used instead of $\Psi_u$, resulting in easier implementation with effectively no trade-off in performance.
Figure 5.1: $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle in the absence of interference.
Figure 5.2: $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle for an interference scenario containing 3 partial-band interferers.
Figure 5.3: $C/N_0$, carrier phase bias, and code delay bias vs. elevation angle for an interference scenario containing 3 partial-band and 3 wideband interferers.
Figure 5.4: Distribution of $C/N_0$ in terms of upper-hemispherical coverage for multiple scenarios with 3 partial-band interferers (top), scenarios with 3 wideband interferers (middle), and scenarios containing 3 partial-band interferers and 3 wideband interferers (bottom).
Figure 5.5: Distribution of $C/N_0$ in terms of upper-hemispherical coverage over multiple scenarios containing 3 partial-band and 3 wideband interferers with adaptive filter lengths of 3 taps (top), 5 taps (middle), and 9 taps (bottom).
5.5 Summary & Conclusions

This chapter derived a novel adaptive beamforming algorithm that is able to constrain the antenna-induced carrier phase and code delay biases introduced into GNSS receiver measurements. The motivation behind its design is to allow adaptive antenna arrays to be used in precision GNSS applications where the antenna-induced bias errors produced by other methods were prohibitively large. Simulations demonstrated that the proposed algorithm achieves zero antenna-induced biases while simultaneously achieving C/N$_0$ that approaches the theoretical upper bound.

While the proposed adaptive filtering algorithm is able to mathematically guarantee zero antenna-induce bias, it does so by making use of stored antenna manifold information. To perfectly prevent bias errors, this information must be known to great accuracy and precision. In many situations, obtaining antenna manifold information of the required quality is impractical. The last chapters of this dissertation will focus on acquiring this information; however, before this is addressed, the next chapter will discuss an alternative bias mitigation approach.
CHAPTER 6

Mitigating Adaptive Antenna-Induced Errors in GNSS Receivers

Much like the previous chapter, this chapter is also focused on preventing the adaptive antenna from introducing errors in the GNSS receiver measurements. Since system designers have different circumstances, it might be the case that implementing a special adaptive filter algorithm – as proposed in Chapter 5 – is not possible. To provide an alternative, this chapter develops a novel bias estimation and correction technique in which additional logic is added to the receiver in order to provide run time compensation for these antenna-induced biases. The technique is general in the sense that it can be applied to a wide variety of adaptive antenna and receiver implementations. It utilizes a computationally efficient bias estimation equation that incorporates stored antenna manifold data, the adaptive filter weights, and the receiver discriminator function. After estimating the bias, the receiver can compensate for the bias errors either in the navigation processor or directly in the tracking loops. Section 6.1 begins by reviewing GNSS receiver tracking loops. Next, Section 6.2 discusses antenna-induced errors in the context of the tracking loop, and Section 6.3 discusses how to estimate and correct for these errors at run time. It is important that estimation of antenna-induced errors be very computationally efficient, and a
thorough discussion is devoted to this in Section 6.4. Finally, Section 6.5 shows example simulation results which demonstrate high accuracy mitigation of adaptive antenna-induced biases.

## 6.1 GNSS Receiver Tracking Loop

This section provides a brief review of the tracking loop in a traditional GNSS receiver. The simple model provided in Fig. 6.1 is based on [1] and shows both the delay lock loop (DLL) and phase lock loop (PLL). Ultimately, the discussion in this section will provide a basis for defining antenna-induced code and carrier phase biases. Initially, however, antenna effects will be neglected and one can consider this the receiver’s ideal behavior.

![Figure 6.1: The general, simplified GNSS receiver tracking loop model used in this study showing the delay lock loop (DLL) and carrier phase lock loop (PLL).](image)

103
Recall from Chapter 2 that the received GNSS signal $y[n]$ is a complex digital signal in the form

$$ y[n] = \sqrt{C_d} \ d(nT - \tau_r) \ e^{-j\psi} $$

(6.1)

after reception, down-conversion, and analog-to-digital conversion. Here, $C_d$ is the power of the incident signal, $d(t)$ is a pseudorandom noise (PRN) spreading code, $T$ is some sufficient sampling period, $\tau_r$ is the propagation delay between the satellite and the receiver, and $\psi$ is the carrier phase shift over the same distance. We will not consider clock biases, Doppler effects, data bits, or other complications of the signal model since these aspects are independent of adaptive-antenna related effects. It is understood that the $d(t)$ has unit power and is purely real.

Upon entering tracking mode, the receiver has a reasonable estimate of the propagation delay $\tau_r$ and carrier phase $\psi$. Let those estimates be $\hat{\tau}_r$ and $\hat{\psi}$, respectively. As depicted in Fig. 6.1, these estimates are provided to the navigation processor for calculation of the navigation solution. Additionally, these estimates are used as feedback for the tracking loops. A numerically controlled oscillator uses the phase estimate to generate a carrier signal which controls carrier wipe-off. A code generator uses the code phase estimate to generate a replica of the transmitted PRN signal $r(t)$ which we will refer to as the reference signal and is given by

$$ r(t) = d(t - \hat{\tau}_r). $$

(6.2)

At any instant, the relative delay between the incident and reference signals and the relative phase between the incident signal and local oscillator are, respectively,

$$ \tau_0 = \tau_r - \hat{\tau}_r $$

(6.3)

$$ \psi_0 = \psi - \hat{\psi}. $$

(6.4)
To accomplish lock, the receiver estimates these relative quantities based on the cross-correlation of the received and reference signals,

\[
R_{yd}(\tau) = E\{y[n]r(nT - \tau)\} 
\]

\[
= \sqrt{C_d} E\{d(nT - \tau_r)d(nT - \hat{\tau}_r - \tau)e^{j\psi_0}\} 
\]

\[
= \sqrt{C_d} R_{dd}(\tau - \tau_0)e^{j\psi_0},
\]

where \(R_{dd}(\tau)\) is the auto-correlation of the spreading code. In this case, it is delayed by \(\tau_0\) and phase shifted \(\psi_0\). In a real receiver, the expectation in (6.5) would be a finite length integration and the resulting cross-correlation would have residual noise. However, since the focus of this paper is on bias errors, noise will be neglected.

Effectively, both the PLL and DLL utilize samples of the cross-correlation in (6.7) to estimate the delay and phase errors, \(\tau_0\) and \(\psi_0\). These samples are calculated by a correlator bank at a set of fixed delays relative to \(\tau_0\). This set of \(M\) delays is known as the correlator taps, which we denote here as \(\{c_1, \ldots, c_M\}\). Using these cross-correlation values at these delays, the DLL forms an estimate of the relative delay, \(\hat{\tau}_0\), using a discriminator function \(D\) where

\[
\hat{\tau}_0 = D\left(R_{yd}(\tau_0 - c_1), \ldots, R_{yd}(\tau_0 - c_M)\right). 
\]

(6.8)

Alternatively, this can be rewritten as a function of \(\tau_0\),

\[
\hat{\tau}_0 = S(\tau_0) 
\]

(6.9)

and is commonly known as the S-curve. The S-curve depends on the discriminator as well as the antenna response since each cross-correlation \(R_{yd}\) depends on the received signal \(y\) at the output of the antenna. The DLL ultimately provides a propagation
delay estimate by locking onto the zero of the S-curve at $\hat{\tau}_r$

$$S(\hat{\tau}_r) = 0.$$  \hspace{1cm} (6.10)

The discriminator for the PLL is typically much simpler, and this study will use

$$\hat{\psi}_0 = \angle R_{yd}(0).$$ \hspace{1cm} (6.11)

The tracking loops will continually adjust $\hat{\tau}_r$ and $\hat{\psi}$ until they have converged to zero update ($\hat{\tau}_0 = 0$ and $\hat{\psi}_0 = 0$). For this study, we will neglect the loop filters, since we are concerned with the steady-state biases of the receiver, on which the loop filters should have no effect.

This chapter will focus on the three GPS signal types. The first is the coarse acquisition code or C/A-code and is used in civilian GPS receivers. The second signal is the precision (encrypted) or P(Y)-code and is used in military GPS receivers. The third signal is the new military code or M-code. The respective power spectra and auto-correlation of these signals are shown in Fig. 6.2. These are the ideal cases and antenna effects are not yet included. Each signal type will be tracked using a different DLL discriminator. The motivation here is not to discuss which discriminator is better, but rather to provide a diverse set of tracking behaviors. The sampling rate throughout the adaptive processing and receiver processing will remain a constant 24 MHz.

C/A-code tracking will use the early-late-slope discriminator [70]. Its discriminator function is given by

$$D(E, L) = \frac{(E - L) + \delta/2(m_1 + m_2)}{m_1 - m_2},$$ \hspace{1cm} (6.12)

where $\delta = 2T_0$ corresponds to the spacing between the early $E = R_{yd}(\tau_0 - \delta/2)$ and late $L = R_{yd}(\tau_0 + \delta/2)$ samples, and $m_1$ and $m_2$ correspond to the slopes on the early
and late sides of the cross-correlation main lobe, respectively. For the P-code, the common early-minus-late discriminator [71] was used and is given by

\[ D(E, L) = \frac{E - L}{2\sqrt{C_d/T_c}}, \]  

(6.13)

where \( E = R_{yd}(\tau_0 - T_0) \) is the early sample, \( L = R_{yd}(\tau_0 + T_0) \) is the late samples, \( T_0 \) is the sampling period, and \( T_c \) is the chipping rate. For the M-code, the smooth multi-gate discriminator (MGD) [72,73] was used and is given by

\[ D(E_1, L_1, \ldots, E_N, L_N) = \sum_{n=1}^{N} \alpha_n (E_n - L_n) \frac{T_c}{\sum_{n=1}^{N} \alpha_n (E_n + L_n)}, \]  

(6.14)

where \( E_n \) and \( L_n \) correspond to the multiple early and late samples weighted by \( \alpha_n \). The early and late samples are each separated by the sample spacing \( T_0 \), and determination of the weighting coefficients is discussed elsewhere [72].

The S-curves for each signal type and discriminator are also shown in Fig. 6.2. The S-curves were evaluated by using each signal’s auto-correlation function to evaluate (6.9) with the corresponding discriminator for that signal type.
Figure 6.2: Three GPS signal types and their respective power spectra (top), autocorrelation functions (middle), and $S$-curves (bottom) for the particular discriminators.
6.2 Antenna-Induced Bias Errors in the Tracking Loop

To determine the effect of the adaptive antenna on measurements made by the GNSS receiver tracking loops, it is necessary to consider the effects of the adaptive antenna on the receiver cross-correlation. Without loss of generality, let us assume that the incident and reference signals are synchronized. That is, \( \tau_0 = 0 \) and \( \psi = 0 \) in (6.7). In this way, any code or carrier phase measured from the cross-correlation corresponds solely to antenna-induced biases. The responses of the antenna, front-end and STAP filters can be combined into a single direction-dependent adaptive antenna response. From Chapter 3, this response is

\[
H(f, \theta, \phi) = \sum_{k=1}^{K} W_k(f) F_k(f) A_k(f, \theta, \phi).
\]

(6.15)

In this case, the received signal in (6.1) is filtered by the antenna, and it follows that the cross-correlation is also filtered by the adaptive antenna response. The new cross-correlation in the GNSS receiver is

\[
R_{yd}(\tau) = \sqrt{C_d} \int H(f, \theta, \phi) G_d(f) e^{-j2\pi f(\tau-\tau_{ref})} df.
\]

(6.16)

where \( G_d \) is the normalized power spectral density of the SOI. Depending on the length of the adaptive filters and the algorithm used to determine their weights, the response (6.15) can potentially distort the cross-correlation function in a way that is dependent on the incident signal scenario. The discriminator acts on this new, distorted cross-correlation function. Consequently, the S-curve will also be distorted, and the tracking-loop will produce biased results.

Simulations were performed to demonstrate S-curve distortion and quantify the measurement biases introduced by an adaptive antenna. A 6-element, circular array
of patch antennas receiving on the GPS L1 band was used and is described in Chapter 3 and depicted in Fig. 3.3. The SOI is one of three types of GPS signals introduced in Section 6.1 and has a -30 dB signal-to-noise ratio (SNR) on an isotropic element. In the cases with interference, one wideband (24 MHz, flat spectrum) interferer is present with a 50 dB interference-to-noise ratio (INR). 7-tap STAP processing was performed. Two commonly implemented adaptive algorithms will be used in this paper, and they were already introduced in Chapter 3. The first is the simple power minimization method. The second algorithm is commonly referred to as simple beam steering. It performs both beamforming and null steering, so it should yield greater C/N₀ performance over the first algorithm.

First, we will demonstrate the S-curve distortion caused by the adaptive antenna. Fig. 6.3 shows S-curves for each of the C/A-code, P-code, and M-code tracking loops. Results are for the SOI incident at an angle 50° offset from broadside using the simple power minimization adaptive algorithm. Three different cases are shown: the ideal case without antenna effects, the case using the antenna without interference present, and the case using the antenna with interference present. Both in the presence and absence of interference, the antenna distorts the S-curve substantially. The zero-crossing of the S-curve, which corresponds to the code phase bias, is shifted noticeably away from the origin. Furthermore, we note that the biases introduced by the antenna are different depending upon whether or not interference is present.

Next, we will quantify the antenna-induced bias errors for varying SOI incidence angles. Fig. 6.4 shows the adaptive antenna-induced code delay and carrier phase biases vs. incident signal angle both in the absence and presence of interference. Again, only the simple power minimization algorithm is used. The azimuth of the
SOI is fixed at $\phi = 0^\circ$ while its elevation is varied from $\theta = -90^\circ$ to $90^\circ$, where $0^\circ$ corresponds to boresight of the array. One interferer is located in the cut at $(\theta, \phi) = (80^\circ, 0^\circ)$. Clearly, the biases change significantly versus angle-of-arrival, and they also change depending on the interference scenario. While biases are worst near the interferer, they can also vary far away from the interferer. It is evident that these biases must be mitigated if the adaptive antenna system is to be used in precision GNSS applications.
Figure 6.3: S-curves for the C/A-code (top), P-code (middle), and M-code (bottom) tracking loops showing the effects of the antenna in the presence and absence of interference using the simple null steering algorithm. Also shown is the ideal S-curve which neglects antenna effects.
Figure 6.4: Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle using the simple null steering algorithm.
6.3 Bias Estimation and Correction

Precise estimation of the antenna-induced carrier phase and code phase biases requires consideration of the antenna array, front-end antenna electronics, adaptive filters, and the receiver tracking loop implementation. This section describes a simple technique which accounts for all of these aspects. It will be shown that by considering the S-curve, simple equations present themselves regarding how to estimate the necessary DLL bias corrections. Additionally, a simple method for carrier phase bias estimation is also shown.

This chapter describes additional logic that can be added to a receiver in order to compensate for the antenna-induced code and carrier phase biases. The proposed system is depicted in Fig. 6.5. Bias estimation is accomplished using the in-situ

![Figure 6.5: The proposed bias estimation and correction mechanism for a GNSS receiver equipped with an interference suppressing adaptive antenna array.](image_url)
volumetric patterns of individual antenna elements which are stored locally in a compact form, as well as runtime access to the adaptive filter weights and attitude. The biases can be predicted for a wide variety of adaptive algorithms and tracking loop designs. More practically, it would be desirable to have a bias mitigation approach which applies generally and requires minimal modification to existing systems.

Initially, one might attempt to estimate the antenna-induced biases by analyzing the transfer function of the adaptive antenna in (6.15). For instance, one can isolate the group delay and carrier phase shift caused by the filter on a given signal. Such a technique would, in-fact, yield a good approximation of antenna-induced bias errors. This has been demonstrated by [23] where it was used to study the biases caused by many adaptive antennas operating in the presence of multiple interfering signals. However, while this is a good approximation of the actual code phase bias, there are some scenarios in which this approximation is not sufficient for precision navigation applications. The actual bias is based on a more complicated interaction between the antenna, the adaptive filter, and the receiver tracking loop. From the discussion in Section 6.1, it is understood that it is the S-curve which incorporates all of these components, and, based on this fact, this section develops a more precise bias correction method using the S-curve.

Recall that the code tracking loop locks onto the zero of the S-curve. In order to correct the code phase bias, the zero-crossing of the S-curve must be effectively moved so that it is bias-free (that is, so it occurs at $\tau_0 = 0$). There are two different ways to accomplish this. Fig 6.6 shows a closer view of an antenna distorted S-curve, and two distinct delays are denoted $\tau_1$ and $\tau_2$. In this case, to put the zero in the correct place, the S-curve can be shifted down $\tau_1$ or to the right $\tau_2$. In a different case, it
might be necessary to shift up or left. To accomplish these shifts, correction terms can be inserted into the tracking loop model as depicted in Fig. 6.7. A correction term can be applied in two places, and, while the figure shows both terms in the model simultaneously, only one of the corrections will need to be applied. The receiver designer may choose which is the more applicable and implementable.

To shift up and down, a $\tau_1$ term is added into the model. The effect of inserting an additional term here is to effectively create a new discriminator function and, thus, a new S-curve

$$S'(\tau_0) = S(\tau_0) - \tau_1.$$  \hspace{1cm} (6.17)

This new S-curve has a zero at the origin. We will identify this type of correction as a tracking loop bias correction, since the correction is part of the feedback loop.

Alternatively, we can shift left-right. In this case, we let the tracking loop lock onto the biased value $S(\tau_2) = 0$ such that the DLL estimates $\hat{\tau}_r = \tau_r + \tau_2$. This value is output to the navigation processor to be used as a pseudorange value. However, in the navigation processor, we will utilize a bias correction term so that the corrected

\[ Figure 6.6: \text{An example of an S-curve, distorted by an antenna, showing the two bias correction terms.} \]
propagation delay is

\[ \hat{\tau}' = \hat{\tau} - \tau_2 \]  

(6.18)

This method will be referred to as navigation processor bias correction. Thus, we have identified two subtly different places at which to perform bias correction. But before correction, we must estimate these correction terms. Note that the values of \( \tau_1 \) and \( \tau_2 \) can be quite different.

The tracking loop correction term, \( \tau_1 \), will be very easy to find and can be known perfectly. If one refers back to Fig. 6.6, it is clear that the tracking loop correction term is simply the \( \hat{\tau}_0 \)-axis crossing of the S-curve, \( S(0) \). That is,

\[ \tau_1 = S(0). \]  

(6.19)

Figure 6.7: Receiver code phase tracking loop model. The tracking loop bias correction term, \( \tau_1 \), and the navigation processor bias correction term, \( \tau_2 \), are both shown; however, only one type of correction would need to be applied in a specific implementation.
Unfortunately, solving for the navigation processor bias correction term, $\tau_2$, is more difficult. To find $\tau_2$ is to find $\tau$ such that $S(\tau) = 0$. Although all the information that is needed is known, to find this delay would require either a detailed analytical treatment based on the discriminator function or a brute-force search. Either option is not particularly appealing. For this reason, suitable approximations will be used.

In Fig. 6.6, the slope of the S-curve between its two axes crossings is very linear. In fact, simulations show that this fact is generally true. The slope can be approximated at two places on the S-curve by

$$m_1 = \frac{S(0) - S(-d\tau)}{d\tau}$$
$$m_2 = \frac{S(d\tau) - S(0)}{d\tau}$$

(6.20)

(6.21)

where $d\tau$ is a sufficiently large delay. For instance, this study uses a $d\tau$ corresponding to 1 meter. Next, a linear approximation is performed on the side of the S-curve that is likely to contain the $x$-crossing

$$m = \begin{cases} m_1, & \text{if } m_1 > 0 \\ m_2, & \text{if } m_1 \leq 0 \end{cases}$$

(6.22)

$$\hat{\tau}_2 = -\frac{S(0)}{m}.$$  

(6.23)

The proceeding section will simulate a realistic adaptive antenna and demonstrate that this simple linear approximation is sufficient to correct for the bias errors. If this approximation was not sufficient for a different adaptive antenna system (for instance, one with significantly greater biases), then it should be understood that the technique is easily extended to include more S-curve samples and higher-order approximations.

One matter overlooked in the bias estimates of (6.19) and (6.23) is how to evaluate the S-curve at specific delays. This involves application of the known discriminator
function to specific cross-correlation values. As an example, consider the simple, non-coherent early-minus-late discriminator used for the P-code in this study. In this case, the cross-correlation is evaluated at two fixed delays,

\[ S(0) = \frac{|R_{yd}(0 + T_0)| - |R_{yd}(0 - T_0)|}{2\sqrt{C_d/T_c}} \]  

where \( T_c \) is the PRN chipping rate. The cross-correlation is given by (6.16), and note that the \( \sqrt{C_d} \) term cancels out. Since (6.16) is costly to evaluate, the next section will describe a more efficient means to solve for it.

For the carrier phase bias estimation, it is necessary to evaluate only the prompt value of the cross-correlation

\[ \psi_1 = \angle R_{yd}(0) \]  

This equation should provide perfect estimation and correction of the carrier phase bias. Fig. 6.7 shows the carrier phase correction being applied in the navigation processor.

### 6.4 Implementation

Evaluation of the cross-correlation as given by (6.16) is the first step to estimating the adaptive antenna-induced biases. However, direct evaluation of (6.16) would be too cumbersome to be performed at runtime. For practical purposes, we will represent the cross-correlation in an alternative, vector form. This form will allow efficient storage of the antenna pattern in the receiver and faster estimation of the cross-correlation at fixed delays.

To begin, we will assume that the STAP filter weights are in the steady-state. That is, they are constant over the time interval corresponding to the correlation.
Recall from Chapter 4 that the estimated receiver cross-correlation can be simpliﬁed to

\[
\hat{R}_{yy}(\tau) = E\{y[n]d(nT + \tau - \tau_{ref})\} \\
= E\{x^T[n]w d(nT + \tau - \tau_{ref})\} \\
= E\{x^T[n]d(nT + \tau - \tau_{ref})\}w \\
= s^H(\tau)w
\]

where \(s(\tau)\) is known as the correlation vector and is deﬁned by

\[
s(\tau) = E\{x_d^*[n]x(nT + \tau - \tau_{ref})\}.
\]

Here, \(x_d[n]\) represents only the SOI component of \(x[n]\) since the noise and interference is assumed to be uncorrelated with the reference signal. The \(KLx1\) correlation vector is composed of \(K\) \(Lx1\) correlation vectors corresponding to each antenna element,

\[
s(\tau) = \begin{bmatrix}
s_1(\tau) \\
\vdots \\
s_K(\tau)
\end{bmatrix},
\]

and the \(l\)th value of each vector can be evaluated analytically as

\[
[s_k(\tau)]_l = \int G_d(f)F_k^*(f)A_k^*(f, \theta, \phi)e^{-j2\pi(f(lT + \tau - \tau_{ref}))}df,
\]

Evaluation of (6.32) requires the power spectrum of the SOI, the front-end response, and antenna response of each element in the array. These quantities are constant and available before runtime. Therefore, the correlation vector for a particular relative delay, \(\tau\), can be pre-processed and stored in a look-up table. This can be done for all angles-of-arrival and signal spectra. On the other hand, the STAP ﬁlter weights \(w\) change based on the incident signal scenario, and cannot be pre-processed. However,
we see that the vector form of the cross-correlation given by (6.29) perfectly separates these two components.

The bias estimation and correction techniques in the preceding section require evaluation of the cross-correlation for particular delays. Using (6.29), this evaluation will be a simple dot product. It is only necessary to store the correlation vector for the delays corresponding to the correlator taps used by the discriminator. For instance, the discriminator in (6.24) would require it to be stored for two delays. Since all discriminators are functions of the cross-correlation values, $S(0)$ can be found for any discriminator. The computational requirements are quite reasonable. Evaluation of (6.29) requires a single dot product, and (6.19) need only be evaluated when the steady-state value of the weights changes.

### 6.5 Example Results

This section will demonstrate the mitigation of antenna-induced biases using the technique described in the previous section. Simulations are performed using the antenna array in Chapter 3 and the two adaptive algorithms given in the previous section. The simulated receiver tracks GPS C/A-code, P-code and M-code in the presence of wideband interference. The code delay and carrier phase biases of the adaptive antenna system will be measured both before and after mitigation. Simulation parameters in this section are identical to those described in Section 6.3. The incident elevation angle of the SOI was varied and the code phase and carrier phase biases were measured. A single strong, wideband interferer is present at $\theta = 80^\circ$, and it is expected that the bias errors will be worst in the vicinity of the interferer.
Fig. 6.8 shows the results using the simple power minimization algorithm. From top to bottom, each set of sub-figures correspond to the C/A-code, P-code, and M-code signals. The original code delay and carrier phase biases are shown in each figure, as well as the residual biases after mitigation. Both code delay bias correction methods exhibit near perfect bias correction. The only case in which one bias mitigation method fails is when the SOI is very close to the interferer; however, in this case, it is likely that the $C/N_0$ very close to the interferer will not be able to support signal acquisition. Also note that carrier phase bias has been completely mitigated.

Fig. 6.9 shows the exact same scenario but using the single directional constraint algorithm. Using this algorithm, the original code delay and carrier phase biases are much smaller. In fact, this algorithm introduces close to zero carrier phase bias, even near the interferer. Again, both code delay bias correction methods display near perfect bias correction. Although this section provided results for only a few cases, it should be noted that its performance is similar for scenarios with multiple interferers or interferers with varying powers and bandwidths. The near perfect performance of our bias mitigation technique is simply a result of the mathematical exactness of the approach.
Figure 6.8: Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle. The adaptive algorithm is the simple power minimization method.
Figure 6.9: Code delay (left) and carrier phase (right) biases for the C/A-code (top), P-code (middle), and M-code (bottom) versus signal elevation angle. The adaptive algorithm is the single directional constraint method.
6.6 Summary & Conclusions

A novel technique was presented to mitigate the bias errors caused by adaptive antennas in GNSS receiver systems. This technique, which involves the addition of logic in the receiver, is general and applicable to a variety of adaptive antennas and receiver implementations. It provides efficient storage and runtime estimation of the antenna-induced biases and requires minimal modification to existing systems. Simulations demonstrate that bias mitigation succeeds in environments with interference and provides sufficient accuracy and precision for GNSS systems.

The bias mitigation approaches of Chapters 5 and 6 represent two techniques that allow use of interference suppression in precision GNSS applications. Both also assume that very accurate antenna manifold information is available to them. The next two chapters will discuss new approaches for obtaining this calibration information.
CHAPTER 7

Self-Calibration Algorithm for GNSS Antennas

GNSS antennas must be calibrated with high accuracy in order to be used in precision applications. Unfortunately, obtaining sufficiently accurate calibration measurements is a task that is commonly tedious, inconvenient or impractical. In this chapter, we discuss the feasibility of extending a GNSS receiver to perform self-calibration: simultaneous estimation of the navigation parameters and antenna calibration information. This self-calibration can be performed on-the-fly (i.e. during navigation) instead of being restricted to special calibration environments and procedures. Since the antenna is calibrated in-situ, platform effects will be incorporated automatically. As the platform moves, the estimate of the antenna-induced biases will steadily converge as the antenna is exposed to sufficient diversity in the angles of arriving satellite signals. Emphasis has been placed on incorporating the self-calibration logic into an add-on module that processes standard GNSS receiver outputs. In this chapter, we will focus on calibration of a single-element antenna. Section 7.1 begins by justifying the need for a new calibration approach. Section 7.2 introduces the proposed self-calibration system, which can be considered an incremental extension of existing GNSS field calibration techniques. Sections 7.3 and 7.4 define the antenna model and
the differential GNSS measurement model. Section 7.5 isolates an observable component of the antenna-induced biases that is independent of user motion, and, based on this, Section 7.6 formulates a self-calibration algorithm. Section 7.7 describes an efficient, iterative implementation of the algorithm, and Section 7.8 shows results from simulations that demonstrate successful calibration of a single-element GNSS antenna.

7.1 Introduction

Accurate calibration of single-element GNSS antennas are currently performed in special antenna measurement chambers or using carefully controlled, outdoor field calibration procedures [33, 48, 74]. Both options are costly, time-consuming, and require significant resources. Moreover, there are still a number of potential error sources that are difficult to overcome. For example, even if an antenna can be precisely measured in a chamber, the effective antenna response will change once the antenna is mounted on platforms such as aircraft or ships [75, 76]. While outdoor field calibrations can be performed with the antenna mounted on the platform, large platforms pose a significant technical burden. Furthermore, manufacturing tolerances can also lead to significant differences between antennas of the same model. Other error sources include antenna aging and temperature. In the end, one is left with nagging uncertainty as to the quality of antenna calibration data. To be confident in the calibration, a system designer might conclude that the specific antenna on the specific platform must be calibrated in each case—a highly impractical conclusion for either chamber measurements or field calibrations.
It would be ideal if the GNSS receiver could simply use its standard measurements to continually update and improve stored antenna information on-the-fly. Since the receiver itself would jointly estimate the nuisance parameters (the antenna model) and desired parameters (navigation information), the system is said to be performing self-calibration. The concept of antenna self-calibration is common in literature, and there are a number of properties of GNSS that make it an ideal candidate for self-calibration: each satellite signal is almost completely known, each signal arrives at a precisely known angle (provided the attitude of the platform is known), there are a large number of satellites visible at any time, and their signals are all precisely synchronized. Unfortunately, these advantages are overshadowed by some significant drawbacks. First, GNSS signals are very weak and must be heavily processed before noise is sufficiently suppressed. One should not re-implement this processing to perform calibration. Also, it should not be assumed that one has access to processes deep within the GNSS receiver. Second, although the incident signals are known by the receiver, many important military signals are encrypted and access to these signals by any proposed receiver extension will be difficult. One should assume that only standard GNSS receiver outputs are available. Third and most importantly, the particular antennas we are ultimately interested in (complicated arrays mounted in non-ideal locations on complex platforms) require a large, detailed antenna model for highly accurate calibration. All too often, the self-calibration approaches proposed in literature use trivial antenna models such as direction-independent gain/phase shifts, phase center offsets, and mutual coupling matrices [77–81]. These models are simply not applicable for all antennas that are used in high-precision GNSS applications.
All of these drawbacks hinder the creation of a practical self-calibration implementation, and - to the author's knowledge - there are no reported uses of GNSS antenna self-calibration systems in practice and very limited treatment in literature [82, 83]. It is a very difficult problem, no doubt, but one whose benefits are clear and whose consequences are great.

This chapter develops a novel antenna self-calibration system for precision GNSS receivers. The proposed system serves as a proof-of-concept that the above hurdles can be overcome, that a practical and efficient implementation is possible, and that self-calibration - at the very least - is conceptually feasible for precision GNSS receivers. The next section provides an overview of the proposed system, and the following sections describe the mathematical and implementation details of the algorithm. To simplify our initial discussion, only calibration of single-element antennas will be performed in this chapter. Discussion of adaptive antenna array calibration will wait until Chapter 8.

7.2 Overview of the Proposed System

The proposed antenna calibration logic is located in a self-contained module as depicted in Fig. 7.1. By design, the antenna calibration module is very loosely-integrated with existing, unmodified GNSS receivers. The few inputs it accepts and outputs it provides are already standard in GNSS receiver designs and operate at very low rates (~10Hz). Great effort has been made to ensure that the raw received signals do not need to be directly processed by the antenna calibration logic. This will drastically reduce the computational complexity as compared to other techniques that have been proposed. Standard differential measurements from the GNSS receiver
(pseudorange, carrier phase, and C/N₀) are streamed to the calibration logic which iteratively updates locally stored antenna calibration data. This calibration occurs continuously, even during navigation while the system is in motion and its relative position not known. It should be emphasized that calibration of the antenna does not require accurate position/time estimation (for if it did, this approach would trivially reduce to existing field calibration procedures). We assume that there exists some source of external control that determines when the system should be calibrating and when it should not be. This mechanism would depend heavily on the specific application, and this study will not explore this aspect.

![Diagram of proposed self-calibration logic](image)

**Figure 7.1:** Integration of the proposed self-calibration logic with a standard receiver system.
The existing receiver system that is to be extended must meet some minimal requirements. During calibration, the receiver must operate in differential mode whereby the GNSS measurements from a nearby reference station are used to cancel common mode errors such as ionospheric effects, tropospheric effects, and satellite clock biases. The baseline between the user and reference receivers must be short enough so that these errors are sufficiently canceled. The baseline length will depend heavily on the current propagation (ionospheric and tropospheric) conditions. For calibration accuracies of a few centimeters, the maximum baseline would typically be on the order of several kilometers [1]. In this case, the differential measurements would then solely depend on the differential user position/clock, the antenna-induced effects, as well as noise and multipath. Standard differential GNSS measurements are to be provided to the self-calibration module and serve as the fundamental measurement from which the antenna information is extracted. Although an outdoor environment may seem like an inappropriate location for a high-precision antenna calibration, many authors claim very precise results for traditional field calibrations [47, 49]. We appeal to these previous results as justification that uncertainty in outdoor environments can be sufficiently mitigated.

Note that, for this study, we are interested in an absolute calibration of the antenna. To accomplish this, we will assume that the reference station antenna has already been precisely calibrated. If it were not, then techniques developed for absolute field calibration could potentially be adapted for this system [43, 48, 49]. Our implementation could be greatly simplified if the reference antenna were located on the platform; however, we have chosen to avoid this assumption since calibration of the reference antenna on the platform could bring its own uncertainties.
The system is also required to have an inertial measurement unit (IMU). IMU’s are commonplace in modern GNSS receiver systems, and, in our case, it serves two purposes. First, the coupled IMU/GNSS receiver provides an attitude estimate to the calibration module so that the relative angles to the satellites are precisely known. Second, standard inertial measurements are sent to the antenna calibration module which uses the inertial sensor (specifically, the center of the IMU body frame) as an unambiguous reference point. Ultimately, the antenna calibration data will be made relative to this point using standard “lever arm” estimation methods. These two requirements are very modest, so it is not necessary that the IMU be particularly advanced or tightly-coupled to the receiver.

The antenna calibration data is stored in the receiver and accessed by the antenna calibration logic. It is iteratively updated so that GNSS measurements are used immediately and discarded rather than being buffered or stored. The antenna model used will be discussed in the following section. It is a high-resolution model that involves a significant number of coefficients and allows for a very accurate and precise representation of the actual antenna effects even when the antenna is mounted on a platform where scattering can cause rapid angular variations. In this study, we presume that the module provides antenna-induced bias corrections to the receiver. This is straightforward, so our discussion will only be focused on the estimation of the antenna calibration information.

The environment must be moderately favorable for the calibration. The platform, antenna, and front-end must be stable over a large number of measurements, while any environmental effects (i.e. multipath from nearby objects, etc) should vary quickly as to average out over the duration of the calibration. The result will simply be
more accurate if these conditions break down. Control signals from elsewhere in the
receiver determine whether calibration should be paused if interference is present or
if the platform is in a location not conducive to calibration. In airborne applications,
non-platform multipath should be manageable; however, if calibration is performed on
the ground, non-platform multipath will be a dominant source of error, particularly
when receiving signals at lower elevations. A detailed discussion of multipath effects
on the proposed self-calibration technique is beyond the scope of this work.

7.3 Antenna Calibration Data

As discussed in Chapter 2, the effect of a single-element GNSS antenna on a
specific signal is completely characterized by three continuous functions of incident
angle \((\theta, \phi)\):

\[
\begin{align*}
a_\rho(\theta, \phi) & : \text{code delay bias} \\
a_\phi(\theta, \phi) & : \text{carrier phase bias} \\
a_g(\theta, \phi) & : \text{gain/loss in C}/N_0,
\end{align*}
\]

the first two of which are relative to some known antenna reference point. Note that
the carrier phase bias \(a_\phi\) is always a fraction of a cycle and never contains whole
cycles. The complete characterization of the antenna would require a set of functions
for each GNSS signal type on each frequency band. Acquisition of these functions
would constitute a complete, absolute calibration of a GNSS antenna. For an adaptive
antenna array, significantly more information is required about the entire antenna
array manifold. It is a consequence of this that calibration of GNSS adaptive arrays
must be handled differently and is discussed in Chapter 8. Note that an unfortunate
conflict between conventions leads us to use \(\phi\) both as an azimuthal angle and to
denote carrier phase measurements. Context should make it clear as to which is intended.

Since (7.1)-(7.3) are continuous functions, we must approximate them on some finite basis \{\Omega_j(\theta, \phi)\} for \(j = 1..J\), where

\[
a_\rho(\theta, \phi) = \sum_{j=1}^{J} \alpha_{\rho,j} \Omega_j(\theta, \phi) \tag{7.4}
\]

\[
a_\phi(\theta, \phi) = \sum_{j=1}^{J} \alpha_{\phi,j} \Omega_j(\theta, \phi) \tag{7.5}
\]

\[
a_g(\theta, \phi) = \sum_{j=1}^{J} \alpha_{g,j} \Omega_j(\theta, \phi) \tag{7.6}
\]

If there are \(K\) visible satellites and the angle of the \(k\)th satellite relative to the user receiver is \((\theta_k, \phi_k)\), then three \((K \times 1)\) vectors can be constructed

\[
\begin{bmatrix}
    a_\rho(\theta_1, \phi_1) \\
    \vdots \\
    a_\rho(\theta_K, \phi_K)
\end{bmatrix},
\begin{bmatrix}
    a_\phi(\theta_1, \phi_1) \\
    \vdots \\
    a_\phi(\theta_K, \phi_K)
\end{bmatrix},
\begin{bmatrix}
    a_g(\theta_1, \phi_1) \\
    \vdots \\
    a_g(\theta_K, \phi_K)
\end{bmatrix} \tag{7.7}
\]

and it follows that they can be written as

\[
a_\rho = M v_\rho 
\]

\[
a_\phi = M v_\phi 
\]

\[
a_g = M v_g 
\]

where each \((J \times 1)\) vector \(v\) contains the coefficients,

\[
v_\rho = \begin{bmatrix}
    \alpha_{\rho,1} \\
    \vdots \\
    \alpha_{\rho,J}
\end{bmatrix},
\begin{bmatrix}
    \alpha_{\phi,1} \\
    \vdots \\
    \alpha_{\phi,J}
\end{bmatrix},
\begin{bmatrix}
    \alpha_{g,1} \\
    \vdots \\
    \alpha_{g,J}
\end{bmatrix}
\]

and \(M\) is a \((K \times J)\) matrix where entry \((k, j)\) is given by

\[
[M]_{kj} = \Omega_j(\theta_k, \phi_k). \tag{7.12}
\]
Expressions (7.8)-(7.10) will be used in our GNSS measurement model discussed in the next section. The set of three vectors \( \{v_\rho, v_\phi, v_g\} \) represent the complete antenna model (that is, the complete calibration information) for a single GNSS signal type. It is implied that this information is needed for each GNSS signal type on each frequency band.

In this work, we choose a simple regional basis set which divides the upper-hemisphere into \( J \) independent angular regions, where the \( j \)th angular region will be denoted \( S_j \). The angular basis function set \( \{\Omega_j\} \) corresponding to the set of regions \( \{S_j\} \) is

\[
\Omega_j(\theta, \phi) = \begin{cases} 
1 & \text{if } (\theta, \phi) \in S_j, \\
0 & \text{else} 
\end{cases}, \quad \text{for } j = 1 \ldots J, \tag{7.13}
\]

and \( M \) has a very simple form which maps the angular regions in \( v \) to the specific angles of \( a \).

\[
[M]_{kj} = \begin{cases} 
1 & \text{if } (\theta_k, \phi_k) \in S_j, \\
0 & \text{else} 
\end{cases}. \tag{7.14}
\]

In our case, the upper-hemisphere is divided into regions which span 1 degree in both \( \theta \) and \( \phi \). This results in \( 360 \times 91 = 32760 \) regions with non-uniform areas. This a very high resolution model; and this choice of basis will demonstrate the feasibility of our calibration technique to operate efficiently with a very large number of coefficients. Although more advanced basis functions (such as spherical harmonics) could be used, they require significantly more processing and would not efficiently represent the rapid angular variations caused by platform scattering. Our choice of finite basis functions result in a sparse and efficient \( M \) that is more suited for real-time implementation. If desired, interpolation can be performed in order to fill in small regions that lack measurements.
Currently, the GNSS community uses a variety of different antenna models for calibration purposes (as discussed in Section 2.4), and all can be understood as approximations of (7.1)-(7.3). Compared to these previous approaches, the antenna model used in this study obtains a much higher resolution representation at the cost of significantly greater number of coefficients. Unfortunately, the number of coefficients in the estimation is inversely related to the ability to reduce noise. Therefore, we will see that a very large number of measurements will be required to reduce the noise to practical levels. Presumably, alternative choices of basis functions could be found which are better able to control the trade-off between accuracy, noise, and computational efficiency; however, we will not explore this aspect here.

7.4 Measurement Model

This section derives a linear model for the differential GNSS measurements. It is from these measurements that the antenna model coefficients defined in the previous section will be estimated. Note that, in order to maintain the conventional GNSS notation, we are forced to reuse variable names that defined other quantities in our adaptive antenna model. To prevent confusion, the math in this section should be considered completely independent of the variables defined in earlier chapters.

7.4.1 GNSS Geometric Model

Consider a 3-D coordinate system shown in Fig. (7.2). For our discussion, the choice of origin is completely arbitrary. We will define two general points in space, \( i \)
and \( j \). Each point is represented by a position vector \( \mathbf{x}_i \) containing its x-y-z coordinates

\[
\mathbf{x}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}
\]  
(7.15)

The relative position vector which points from position \( j \) to position \( i \) is a differential position vector \( \mathbf{x}_{ij} \),

\[
\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j.
\]  
(7.16)

We will assume there are \( K \) visible satellites, and the position vector of the \( k \)th satellite is denoted \( \mathbf{x}^{(k)} \). This set of satellite positions are referred to as the satellite scenario. The range from point \( i \) to the \( k \)th satellite is denoted \( r_{i}^{(k)} \), and ranges to the \( K \) visible satellites are collected to form a range vector \( \mathbf{r}_i \), where

\[
r_{i}^{(k)} = \| \mathbf{x}^{(k)} - \mathbf{x}_i \|, \quad \mathbf{r}_i = \begin{bmatrix} r_{i}^{(1)} \\ \vdots \\ r_{i}^{(K)} \end{bmatrix}.
\]  
(7.17)

In this work, it is implied that all vector norms used in equations are the Euclidean norm. The differential range vector \( r_{ij}^{(k)} \) represents the range difference from the \( k \)th satellite to two different points \( i \) and \( j \), where

\[
r_{ij}^{(k)} = r_{i}^{(k)} - r_{j}^{(k)}, \quad \mathbf{r}_{ij} = \begin{bmatrix} r_{ij}^{(1)} \\ \vdots \\ r_{ij}^{(K)} \end{bmatrix}.
\]  
(7.18)

Now consider the case where each point has not only a position in space \( \mathbf{x}_i \), but also a clock which is delayed in time relative to some common clock. If \( t_i \) is the delay of the clock at point \( i \), then we can define a new position/time vector \( \mathbf{u}_i \)

\[
\mathbf{u}_i = \begin{bmatrix} \mathbf{x}_i \\ t_i \end{bmatrix}
\]  
(7.19)

and differential position/time vector \( \mathbf{u}_{ij} \)

\[
\mathbf{u}_{ij} = \begin{bmatrix} \mathbf{x}_i - \mathbf{x}_j \\ t_i - t_j \end{bmatrix}.
\]  
(7.20)
If the $k$th satellite has some delay $t^{(k)}$, then the relative delay between it and point $i$ is denoted

$$t_i^{(k)} = t^{(k)} - t_i.$$  \hfill (7.21)

The differential delay between two relative delays from different points is independent of the satellite clock,

$$t_{ij}^{(k)} = t_i^{(k)} - t_j^{(k)} = (t^{(k)} - t_i) - (t^{(k)} - t_j) \hfill (7.22)$$

$$= t_j - t_i \hfill (7.23)$$

$$= -t_{ij}. \hfill (7.24)$$

The pseudorange to the $k$th satellite $\rho_i^{(k)}$ is a distance that incorporates both range and delay,

$$\rho_i^{(k)} = r_i^{(k)} + t_i^{(k)}, \quad \mathbf{p}_i = \begin{bmatrix} \rho_i^{(1)} \\ \vdots \\ \rho_i^{(K)} \end{bmatrix} \hfill (7.26)$$
where $\mathbf{p}_i$ is a pseudorange vector which collects the pseudoranges from all visible satellites. In (7.26) and the rest of this study, it is implied that $t$ has been converted to a unit of distance using the speed of light. Additionally, a differential pseudorange $\rho_{ij}^{(k)}$ and differential pseudorange vector $\mathbf{p}_{ij}$ can be defined in a similar manner as (7.18)

\[
\rho_{ij}^{(k)} = \rho_i^{(k)} - \rho_j^{(k)}, \quad \mathbf{p}_{ij} = \begin{bmatrix} \rho_{ij}^{(1)} \\ \vdots \\ \rho_{ij}^{(K)} \end{bmatrix}.
\]  

(7.27)

If the antenna introduces a bias error (7.1) into the pseudorange measurement, then we can include it in (7.26) as

\[
\rho_i^{(k)} = r_i^{(k)} + t_i^{(k)} + a_{\rho}(\theta_k, \phi_k)
\]  

(7.28)

$i$ represents an infinitesimal point in space/time, and, after including the effect of a real antenna, we can now interpret $i$ as the antenna reference point. This connects our geometric model with a physical, identifiable point external to the antenna. We will also write (7.28) as

\[
\rho_i^{(k)} = r_i^{(k)} + t_i^{(k)} + a_{\rho,i}^{(k)}.
\]  

(7.29)

### 7.4.2 Linearization of Measurements

The unit vector from point $i$ to the $k$th satellite is denoted $\mathbf{1}_i^{(k)}$ and is given by

\[
\mathbf{1}_i^{(k)} = \frac{\mathbf{x}^{(k)} - \mathbf{x}_i}{r_i^{(k)}}.
\]  

(7.30)

The differential range (7.18) is commonly approximated as

\[
r_{ij}^{(k)} \approx -\mathbf{1}_i^{(k)} \cdot \mathbf{x}_{ij} \approx -\mathbf{1}_j^{(k)} \cdot \mathbf{x}_{ij}
\]  

(7.31)

for cases in which the baseline $\|\mathbf{x}_{ij}\|$ is much smaller than the satellite distances $r_i^{(k)}$ or $r_j^{(k)}$. These range relationships are depicted in Fig. (7.3).
Since the approximation in (7.31) is so important, we will quickly review how it is derived. Recall that the one-dimensional Taylor series expansion of a real function $f(x)$ about a point $x_0$ is

$$f(x) = f(x_0) + (x - x_0)f'(x - x_0) + \frac{1}{2!}(x - x_0)^2 f''(x_0) + \ldots$$  \hspace{1cm} (7.32)$$

If we define the finite difference between $x$ and $x_0$ as

$$\Delta x = x - x_0$$  \hspace{1cm} (7.33)$$

then the Taylor series in (7.32) can be written as

$$f(x_0 + \Delta x) = f(x_0) + \Delta x f'(x_0) + \frac{1}{2!}(\Delta x)^2 f''(x_0) + \ldots$$  \hspace{1cm} (7.34)$$

The vector Taylor series expansion of a scalar function of vector variable follows similarly as

$$f(x_0 + \Delta x) = f(x_0) + (\Delta x \cdot \nabla f(x)) f(x_0) + \frac{1}{2!}(\Delta x \cdot \nabla f(x))^2 f''(x_0) + \ldots$$  \hspace{1cm} (7.35)$$
Which gives the first order approximation of the difference

\[
\left. f(x_0 + \Delta x) - f(x_0) \approx (\Delta x \cdot \nabla x') f(x') \right|_{x' = x_0}
\]  

(7.36)

In our case, the function \( f(x) \) we are interested in is the Euclidean norm

\[
f(x) = \|x\|,
\]

(7.37)

and its derivative is

\[
\left. \nabla x' f(x') \right|_{x' = x} = \frac{x}{\|x\|}
\]

(7.38)

We can now apply these relationships to derive (7.31),

\[
\begin{align*}
\gamma_{ij}^{(k)} &= r_i^{(k)} - r_j^{(k)} \\
&= \|x^{(k)} - x_i\| - \|x^{(k)} - x_j\| \\
&= \|x^{(k)} - (x_j + x_{ij})\| - \|x^{(k)} - x_j\| \\
&= \|x^{(k)} - x_j - x_{ij}\| - \|x^{(k)} - x_j\|.
\end{align*}
\]

(7.39) \hspace{1cm} (7.40) \hspace{1cm} (7.41) \hspace{1cm} (7.42)

Based on the change of variables

\[
\begin{align*}
x_0 &= x^{(k)} - x_j \\
\Delta x &= -x_{ij}.
\end{align*}
\]

(7.43) \hspace{1cm} (7.44)

we can use (7.36) to approximate (7.42) as

\[
\begin{align*}
\gamma_{ij}^{(k)} &\approx -x_{ij} \cdot \frac{x^{(k)} - x_j}{\|x^{(k)} - x_j\|} \\
&= -x_{ij} \cdot 1_j^{(k)}.
\end{align*}
\]

(7.45) \hspace{1cm} (7.46)

Since a satellite is approximately 25,000 km. away, the approximation in (7.46) is accurate to centimeters for baselines of several kilometers.
Using (7.30), the unit vectors to all visible satellites can be combined into a matrix $G_i$, 

g_i = \begin{bmatrix} (-1)^{(1)}_i T & 1 \\ \vdots \\ (-1)^{(K)}_i T & 1 \end{bmatrix}, \quad (K \times 4) \tag{7.47}

where $G_i$ characterizes the satellite scenario’s geometry – the angles to the visible satellites – relative to point $i$. It is often referred to as the geometry matrix. This allows the differential pseudorange $p_{ij}$ to be approximated as linear equation in terms of the differential position/time $u_{ij}$, 

\[ p_{ij} = r_{ij} + t_{ij} \]  
\[ \approx G_i \begin{bmatrix} x_{ij} \\ t_{ij} \end{bmatrix} \tag{7.49} \]
\[ = G_i u_{ij}. \tag{7.50} \]

The approximation in (7.49) is insensitive to small changes in the location of the geometry matrix. For example, $G_i$ could be chosen to be centered around point $j$, 

\[ p_{ij} \approx G_j u_{ij} \tag{7.51} \]

or around the midpoint $m$ between $i$ and $j$, 

\[ p_{ij} \approx G_m u_{ij}. \tag{7.52} \]

In general, this allows for some flexibility with the subscript on $G$ in our equations.

Pseudorange measurements from (7.50) can be extended to include antenna-induced pseudorange biases $a_{\rho}$ from (7.7). Since points $i$ and $j$ both have antennas, we must now explicitly use an additional subscript to denote $a_{\rho,i}$ and $a_{\rho,j}$. The differential measurement would then be 

\[ p_{ij} \approx G_i u_{ij} + a_{\rho,ij}. \tag{7.53} \]
However, we will assume that only point $i$ has antenna biases and write it as

$$p_{ij} \approx G_i u_{ij} + a_{\rho,i}.$$  \hspace{1cm} (7.54)

Substituting (7.8) into (7.54) yields

$$p_{ij} \approx G_i u_{ij} + M v_{\rho,i},$$  \hspace{1cm} (7.55)

and we now have a linear expression relating the unknown differential user position $u_{ij}$ and antenna model coefficients $v_{\rho,i}$ to the GNSS receiver measurements $p_{ij}$. The measurement matrices $G_i$ and $M$ are determined by (7.47) and (7.14) based on the angle to each satellite, and this work will assume they are known exactly. A measurement model in the form of (7.55) will be the basis of our approach; however, we must first discuss how it arises in a more realistic differential GNSS receiver system model.

### 7.4.3 Differential GNSS Measurement Model

Building upon the results of the previous subsections, we will now consider a differential GNSS system. In this case, a reference station broadcasts time-tagged GNSS measurements to a nearby user receiver which differences them with its own measurements. This reformulates the GNSS equations into one of finding the relative (differential) position/time and has the benefit of canceling many common sources of error (in particular, propagation effects on the signals as they travel from the satellites). We have two points $u$ and $r$, where $u$ is the user receiver and $r$ is the reference station receiver. It follows that the mathematics from the previous sections (which applied to arbitrary points $i$ and $j$) will also apply to this case. We will assume reference station antenna is already calibrated and its effects have already been removed. Our model is based on [1] and [84].
Pseudorange Measurements

The pseudorange measurement for the $k$th satellite for user receiver $\rho_u^{(k)}$ and the reference receiver $\rho_r^{(k)}$ are modeled as

$$\rho_u^{(k)} = r_u^{(k)} + t_u^{(k)} + I_{\rho,u}^{(k)} + T_{\rho,u}^{(k)} + \epsilon_{\rho,u}^{(k)} + a_{\rho,u}^{(k)}$$ (7.56)

$$\rho_r^{(k)} = r_r^{(k)} + t_r^{(k)} + I_{\rho,r}^{(k)} + T_{\rho,r}^{(k)} + \epsilon_{\rho,r}^{(k)}$$ (7.57)

where the errors sources are the ionospheric delay $I_{\rho}^{(k)}$, the tropospheric delay $T_{\rho}^{(k)}$, and noise/modeling errors $\epsilon_{\rho}^{(k)}$. In this case, only the user receiver has antenna-induced biases $a_{\rho,u}^{(k)} = a_{\rho}(\theta_k, \phi_k)$. The single difference $\rho_{ur}^{(k)}$ is formed between the user and reference pseudorange measurements to produce

$$\rho_{ur}^{(k)} = \rho_u^{(k)} - \rho_r^{(k)}$$ (7.58)

$$= r_{ur}^{(k)} + t_{ur} + I_{\rho,ur}^{(k)} + T_{\rho,ur}^{(k)} + \epsilon_{\rho,ur}^{(k)} + a_{\rho,u}^{(k)}$$ (7.59)

$$= r_{ur}^{(k)} + t_{ur} + \epsilon_{\rho,ur}^{(k)} + a_{\rho,u}^{(k)}$$ (7.60)

which eliminates satellite clock biases and (for short baselines) also eliminates the ionospheric and tropospheric errors.

Let us assume that the user receiver has calculated an estimate of the differential position/time $u_\zeta r$. Here, we use the symbol $\zeta$ to denote the estimated user receiver point while $u$ is still used to denote the true user receiver point. Then, the differential pseudorange corresponding to this position/time,

$$\rho_{\zeta r}^{(k)} = r_{\zeta r}^{(k)} + t_{\zeta r}$$ (7.61)

can be subtracted from (7.60)

$$\rho_{u\zeta}^{(k)} = \rho_{ur}^{(k)} - \rho_{\zeta r}^{(k)}$$ (7.62)

$$= r_{u\zeta}^{(k)} + t_{u\zeta} + \epsilon_{\rho,ur}^{(k)} + a_{\rho,u}^{(k)}$$ (7.63)
in order to produce a shortened baseline. This will improve the quality of our linearization so that our antenna coefficients can be estimated more accurately. These measurements are collected into a $K \times 1$ vector $\mathbf{p}_{u\zeta}$ for all visible satellites

$$
\mathbf{p}_{u\zeta} = \begin{bmatrix} 
\rho_{u\zeta}^{(1)} \\
\vdots \\
\rho_{u\zeta}^{(K)}
\end{bmatrix}, \quad 
\epsilon_{\rho,ur} = \begin{bmatrix} 
\epsilon^{(1)}_{\rho,ur} \\
\vdots \\
\epsilon^{(K)}_{\rho,ur}
\end{bmatrix}, \quad (7.64)
$$

and this allows us to approximate the measurement using a simple linear expression

$$\hat{\mathbf{p}}_{u\zeta} \approx \mathbf{G}_{\zeta} \mathbf{u}_{u\zeta} + a_{\rho,u} + \epsilon_{\rho,ur}, \quad (7.65)$$

which has the same form as (7.54).

**Carrier Phase Measurements**

The carrier phase measurements (in units of cycles) for the user receiver $\phi^{(k)}_u$ and the reference receiver $\phi^{(k)}_r$ can be modeled

$$
\phi^{(k)}_u = \lambda^{-1} [r^{(k)}_u - I^{(k)}_{\phi,u} + T^{(k)}_{\phi,u}] + f \cdot [t_u - t^{(k)}] + N^{(k)}_u + (\phi^{(k)}_0 - \phi_u) + \epsilon^{(k)}_{\phi,u} + a^{(k)}_{\phi,u}
$$

$$
\phi^{(k)}_r = \lambda^{-1} [r^{(k)}_r - I^{(k)}_{\phi,r} + T^{(k)}_{\phi,r}] + f \cdot [t_r - t^{(k)}] + N^{(k)}_r + (\phi^{(k)}_0 - \phi_r) + \epsilon^{(k)}_{\phi,r}
$$

where $\lambda$ is the wavelength at the carrier frequency $f$. $N$ is an integer which accounts for the fact that the phase $\phi^{(k)}$ can only be measured in fractions of a cycle. Again, the errors sources are the ionospheric phase shift $I^{(k)}_{\phi}$, the tropospheric phase shift $T^{(k)}_{\phi}$, and noise/modeling errors $\epsilon^{(k)}_{\phi}$, $\phi^{(k)}_0$, $\phi_{u0}$, and $\phi_{r0}$ are the initial fractional phases of the satellite, user receiver and reference receiver, respectively. Only the user receiver has antenna-induced biases $a^{(k)}_{\phi,u} = a_{\phi}(\theta_k, \phi_k)$. The single difference $\phi^{(k)}_{ur}$ is given by

$$
\phi^{(k)}_{ur} = \phi^{(k)}_u - \phi^{(k)}_r
$$

$$
= \lambda^{-1} [r^{(k)}_{ur} + I^{(k)}_{\phi,ur} + T^{(k)}_{\phi,ur}] + f \cdot [t_{ur} - t^{(k)}] + N^{(k)*}_{ur} + \epsilon^{(k)}_{\phi,ur} + a^{(k)}_{\phi,u}
$$

$$
= \lambda^{-1} r^{(k)}_{ur} + f \cdot t_{ur} + N^{(k)*}_{ur} + \epsilon^{(k)}_{\phi,ur} + a^{(k)}_{\phi,u}, \quad (7.66)
$$
which eliminates satellite clock biases, ionospheric and tropospheric errors. \( N_{ur}^{(k)*} \) is defined as

\[
N_{ur}^{(k)*} = N_{ur}^{(k)} + (\phi_{uo} - \phi_{ro}) ,
\]

(7.67)

which is a non-integer ambiguity due to the fractional phase of the receivers at the start of tracking. The double difference carrier phase is the difference between two single difference measurements from different satellites

\[
\phi^{(kj)}_{ur} = \phi^{(k)}_{ur} - \phi^{(j)}_{ur} = \lambda^{-1} r^{(kj)}_{ur} + f \cdot t_{ur} + N^{(kj)}_{ur} + \epsilon^{(kj)}_{\phi,ur} + a^{(kj)}_{\phi,u},
\]

(7.68)

in which \( N^{(kj)}_{ur} \) is now an integer ambiguity.

In this work, we will use single difference carrier phase measurements; however, it would be straightforward to reformulate it in terms of double difference ( or perhaps other ) carrier phase measurements depending. These measurements are collected into a \( K \times 1 \) vectors for all visible satellites

\[
\phi_{u\zeta} = \begin{bmatrix} \phi^{(1)}_{u\zeta} \\ \vdots \\ \phi^{(K)}_{u\zeta} \end{bmatrix} , \quad \epsilon_{\phi,ur} = \begin{bmatrix} \epsilon^{(1)}_{\phi,ur} \\ \vdots \\ \epsilon^{(K)}_{\phi,ur} \end{bmatrix} , \quad N^{*}_{u\zeta} = \begin{bmatrix} N^{(1)*}_{u\zeta} \\ \vdots \\ N^{(K)*}_{u\zeta} \end{bmatrix} ,
\]

(7.69)

which allows us to write a simple linear expression

\[
\phi_{u\zeta} \approx G_{\zeta} u_{u\zeta} + N^{*}_{u\zeta} + a_{\phi,u} + \epsilon_{\phi,ur} .
\]

(7.70)

This has the same form as (7.54) except for the issue of the the ambiguity vector \( N^{*}_{u\zeta} \).
Amplitude Measurements

The antenna will also affect the received power of an incident satellite signal. Technically, this can be considered an antenna-induced amplitude bias on C/N\(_0\) measurements, but it is often ignored since it does not cause errors in the position estimation. Nonetheless, since we will make use of the amplitude later in Chapter 8, amplitude measurements will be discussed here.

A GNSS receiver estimates the C/N\(_0\) of the each signal being tracked. Since the effective noise N\(_0\) of the receiver can be found, one can form an estimate of the received signal strength \(g^{(k)}_u\) of the \(k\)th satellite signal. We will model this measurement as the sum

\[
g^{(k)}_u = P^{(k)}_c + a^{(k)}_{g,u} + \epsilon^{(k)}_{g,u} \quad (7.71)
\]

where \(P^{(k)}_c\) is the true incident signal power, \(a^{(k)}_{g,u}\) is the gain/loss due to the antenna, and \(\epsilon^{(k)}_{g,u}\) is some noise in the estimation. Since the reference station antenna has been calibrated, we will assume it is able to directly estimate the incident signal power

\[
g^{(k)}_r = P^{(k)}_c + \epsilon^{(k)}_{g,r} \quad (7.72)
\]

The single difference \(g^{(k)}_{ur}\) is then

\[
g^{(k)}_{ur} = g^{(k)}_u - g^{(k)}_r \quad (7.73)
\]

\[
= a^{(k)}_{g,u} + \epsilon^{(k)}_{g,ur} \quad (7.74)
\]

and these measurements are collected into a \(K\times1\) vectors for all visible satellites

\[
g_{ur} = \begin{bmatrix} g^{(1)}_{ur} \\ \vdots \\ g^{(K)}_{ur} \end{bmatrix}, \quad \epsilon_{g,ur} = \begin{bmatrix} \epsilon^{(1)}_{g,ur} \\ \vdots \\ \epsilon^{(K)}_{g,ur} \end{bmatrix}. \quad (7.75)
\]
to form the simple expression

\[ g_{ur} = a_{g,u} + \epsilon_{g,ur} \]  

(7.76)

In this case, the effect of the antenna is directly observable except for minor additional noise. For this reason, this chapter focuses mainly on the issues surrounding pseudorange and carrier phase measurements.

**Measurement Sequence**

The differential pseudorange, carrier phase, and amplitude measurements are performed at each time instant \( n \) at a fixed rate. Thus, the measurements (7.65), (7.70) and (7.76) should be written as discrete time signals:

\[
p_{u\zeta}(n) \approx G_{\zeta}(n)u_{\zeta}(n) + a_{\rho,u}(n) + \epsilon_{\rho,ur}(n)
\]

(7.77)

\[
\phi_{u\zeta}(n) \approx G_{\zeta}(n)u_{\zeta}(n) + N_{u\zeta}^*(n) + a_{\phi,u}(n) + \epsilon_{\phi,ur}(n)
\]

(7.78)

\[
g_{ur}(n) = a_{g,u}(n) + \epsilon_{g,ur}(n).
\]

(7.79)

or, explicitly showing the antenna model coefficients,

\[
p_{u\zeta}(n) \approx G_{\zeta}(n)u_{\zeta}(n) + M(n)v_{\rho,u} + \epsilon_{\rho,ur}(n)
\]

(7.80)

\[
\phi_{u\zeta}(n) \approx G_{\zeta}(n)u_{\zeta}(n) + N_{u\zeta}^*(n) + M(n)v_{\phi,u} + \epsilon_{\phi,ur}(n)
\]

(7.81)

\[
g_{ur}(n) = M(n)v_{g,u} + \epsilon_{g,ur}(n).
\]

(7.82)

These inputs are assumed to have started at time \( n = 0 \). The satellite information and the attitude estimate can be used to find \( G_{\zeta}(n) \) and \( M(n) \) based on (7.47) and (7.14), respectively. The unknowns are the differential user position/time \( u_{\zeta}(n) \), the carrier phase integer ambiguity \( N_{u\zeta}(n) \), and the set of coefficients representing our
antenna \{v_{\rho,u}, v_{\phi,u}, v_{g,u}\}. The measurement noise is modeled as zero-mean Gaussian with some covariance matrix \(\Sigma\),

\[
\epsilon_{\rho}(n) \sim \mathcal{N}(0, \Sigma_{\rho}) \\
\epsilon_{\phi}(n) \sim \mathcal{N}(0, \Sigma_{\phi}) \\
\epsilon_{g}(n) \sim \mathcal{N}(0, \Sigma_{g})
\]

and is uncorrelated in time. It is implied that the covariance can be found based on the C/N\(_0\) of each tracked signal, and we will simply assume it is known. Although non-platform multipath will also be an important source of error in many calibration environments, it will not be dealt with in this study.

### 7.5 Isolating the Antenna-Induced Pseudorange Biases

This section will address the problem of estimating the antenna-induced pseudorange biases while the user receiver is navigating. The key to this will be the fact that there are commonly 8-12 satellite measurements made simultaneously while the user position/time represent only 4 unknowns. This over-determined system of equations reveals a component of the antenna-induced bias that is observable and independent of the position/clock of the user receiver. This section describes the decomposition of the antenna-induced biases pseudorange biases into observable and ambiguous components. The observable component is collected over multiple measurements in order to form a system of equations that can ultimately be used to solved for the antenna model coefficients.
7.5.1 Antenna-Induced Bias Residual

Starting with the biased differential pseudorange measurements in (7.54), let us try to understand how $a_{\rho,i}$ biases the position/time estimate of the GNSS receiver. We begin by defining $P_{G_i}$ as the projector onto the column space of $G_i$,

$$P_{G_i} = G_i(G_i^T G_i)^{-1}G_i^T. \quad (K \times K) \quad (7.86)$$

This allows the antenna bias vector $a_{\rho,i}$ in (7.54) to be decomposed into two orthogonal components,

$$a_{\rho,i} = \underbrace{P_{G_i} a_{\rho,i}}_{1} + \underbrace{(I - P_{G_i}) a_{\rho,i}}_{2}. \quad (7.87)$$

This type of decomposition using a projector $P_{G_i}$ is often used in GNSS receivers to perform integrity monitoring [85]. Since the first component in (7.87) is in the column space of $G_i$, there exists some vector $\delta u$ such that

$$G_i \delta u = P_{G_i} a_{\rho,i}. \quad (7.88)$$

It is clear from its relation to $G_i$ that $\delta u$ represents a differential position vector. The second component in (7.87) is known as a residual and will simply be denoted $q_i$,

$$q_{\rho,i} = (I - P_{G_i}) a_{\rho,i}. \quad (7.89)$$

Substituting (7.88) and (7.89) back into (7.54) yields

$$p_{ij} \approx G_i(u_{ij} + \delta u) + q_{\rho,i} \quad (7.90)$$

So far, $\delta u$ just represents a shift, and its meaning is not tied to any specific points, so it has been written without subscripts. However, the biases $a_{\rho,i}$ from which $\delta u$
follows in (7.88) are relative to the reference point \(i\). Therefore, we choose to anchor \(\delta u\) at point \(i\) and define it as pointing to some new point \(\xi\). That is,

\[
u_{\xi i} = \delta u
\]  

(7.91)
as shown in Fig. 7.4.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig7.4.png}
\caption{Antenna-induced bias where \(j\) is the reference point location, \(i\) is the initial position/time estimate, and \(\xi\) is the biased position solution.}
\end{figure}

Thus, the antenna-induced pseudorange biases \(a_{\rho,i}\) can be decomposed into two parts \((u_{\xi i}, q_{\rho,i})\)

\[
a_{\rho,i} = G_i u_{\xi i} + q_{\rho,i},
\]  

(7.92)

where

\[
u_{\xi i} = (G_i^T G_i)^{-1} G_i^T a_{\rho,i}
\]  

(7.93)

\[
q_{\rho,i} = (I - P_{G_i}) a_{\rho,i},
\]  

(7.94)

which represent the antenna-induced position/time bias and residual bias, respectively. Since, \(u_{\xi i}\) is a differential position relative to the antenna reference point \(i\),
we can consider it a phase center offset and $\xi$ as the phase center. Note that this decomposition is dependent on the specific satellite scenario and will vary in time even though the biases $a_{\rho,i}$ are static. Furthermore, the decomposition in (7.92) is insensitive to small changes in the location to with which $G$ is respect to (that is, the subscript on $G$).

To understand how antenna-induced biases affect the estimated position/time, (7.91) can be substituted back into (7.90), and we make use of the fact that the sum of any two differential positions which share a common point simplify to

$$u_{ij} + u_{\xi i} = (u_i - u_j) + (u_{\xi} - u_i)$$  

(7.95)

$$= u_{\xi} - u_j$$  

(7.96)

$$= u_{\xi j}$$  

(7.97)

Consequently, (7.90) becomes

$$p_{ij} \approx G_i u_{\xi j} + q_{\rho,i}.$$  

(7.98)

Note how the differential pseudoranges and differential positions in (7.98) correspond to different sets of points. Effectively, the position/time solution will be biased by the antenna.

### 7.5.2 Solving for the Antenna-Induced Pseudorange Bias

Based on our discussion above, the residual bias $q_{\rho,i}$ does not contribute to the position/time error. Another way to interpret this is that the residual is *observable* while the offset term $u_{\xi j}$ is *ambiguous* with the user’s actual position/time. Therefore, when both the user location and the antenna-induced biases are unknown, then the
only reliably observable part of the antenna-induced biases is the residual component. This fact will form the foundation of our on-the-fly antenna calibration approach.

In order to isolate the antenna-induced biases in the receiver’s pseudorange measurements, a projection matrix $Q$ is defined as

$$Q(n) = I - P_{G_{\zeta}}(n). \quad (7.99)$$

The pseudorange measurements $p_{u\zeta}(n)$ are processed to produce a new measurement $q_{\rho,u}(n)$

$$q_{\rho,u}(n) = Q(n) \cdot p_{u\zeta}(n). \quad (7.100)$$

Based on (7.77), we can model these processed measurements as

$$q_{\rho,u}(n) = Q(n) \cdot a_{\rho,u} + Q(n) \cdot \epsilon_{{\rho,ur}}(n) \quad (7.101)$$

$$= Q(n) \cdot a_{\rho,u} + \epsilon'_{{\rho,ur}}(n) \quad (7.102)$$

or

$$q_{\rho,u}(n) = Q(n)M(n) \cdot v_{\rho,u} + \epsilon'_{{\rho,ur}}(n). \quad (7.103)$$

which is only dependent on the antenna. Having created a linear measurement model with Gaussian noise, it is now straightforward to solve for the antenna model coefficients. The simplest way is to collect multiple measurements and form a system of equations

$$\begin{bmatrix} q_{\rho,u}(1) \\ q_{\rho,u}(2) \\ \vdots \end{bmatrix} = \begin{bmatrix} Q(1) \\ Q(2) \\ \vdots \end{bmatrix} \begin{bmatrix} M(1) \\ M(2) \\ \vdots \end{bmatrix} v_{\rho,u} + \begin{bmatrix} \epsilon'_{\rho,ur}(1) \\ \epsilon'_{\rho,ur}(2) \\ \vdots \end{bmatrix} \quad (7.104)$$

and solve for $v_{\rho,u}$. Efficient and iterative solutions to this set of equations will be discussed later in this chapter.
7.5.3 The Unknown Phase Center Offset

Even if the system of equations in (7.104) is solved using an ideal set of measurements, there will still be undetermined components in the antenna model coefficient vector. This fact is not immediately obvious, so this section will demonstrate that these undetermined components correspond to an unknown phase center offset.

Recall that the code delay biases \( a_{\rho,i}(\theta,\phi) \) in (7.1) are relative to some phase reference point \( i \). It is a simple manner to shift the location of this phase reference point (in position and time). To do this, all that is needed is to add a factor \( a_{\rho,ij}(\theta,\phi) \) which will shift from phase reference point \( i \) to point \( j \),

\[
a_{\rho,j}(\theta,\phi) = a_{\rho,i}(\theta,\phi) - a_{\rho,ij}(\theta,\phi) \\
\]

(7.105)

The shift itself is represented by the differential position/time vector \( u_{ij} \), and the value of \( a_{\rho,ij}(\theta,\phi) \) is simply

\[
a_{\rho,ij}(\theta,\phi) = \frac{1}{c} \left[ -\mathbf{1}_i(\theta,\phi) \right] \cdot u_{ij} \\
\]

(7.106)

where \( c \) is the speed of light and \( \cdot \) performs a dot product. This equation represents the direction-dependent delay associated with the shift of the phase center, and \( \mathbf{1}_i(\theta,\phi) \) is the unit vector in the \((\theta,\phi)\) direction. For \( K \) visible satellites, the antenna-induced biases of this shift will be written \( a_{\rho}(u_{ij}) \) and are given by

\[
a_{\rho}(u_{ij}) = \begin{bmatrix} a_{\rho}(\theta_1,\phi_1) \\ \vdots \\ a_{\rho}(\theta_K,\phi_K) \end{bmatrix} = \frac{1}{c} G_i u_{ij} . \\
\]

(7.107)

It follows from (7.8) that they can be written in terms of basis coefficients \( v_{\rho}(u_{ij}) \),

\[
a_{\rho}(u_{ij}) = M v_{\rho}(u_{ij}). \\
\]

(7.108)
(7.107) and (7.108) can be combined and written as

\[
\frac{1}{c} G_i(n) u_{ij} = M(n) v_\rho(u_{ij}) ,
\]

(7.109)

where we have included the dependence on measurement time \( n \). From the form of (7.109), it follows that multiplication by \( Q(n) \) yields

\[
Q(n) M(n) v_\rho(u_{ij}) = 0 .
\]

(7.110)

Therefore, we can add a component \( v_\rho(u_{ij}) \) to the antenna model in (7.104) without affecting the solution,

\[
\begin{bmatrix}
q_{\rho,u}(1) \\
q_{\rho,u}(2) \\
\vdots
\end{bmatrix} = \begin{bmatrix}
Q(1) & & \\
& Q(2) & \\
& & \ddots
\end{bmatrix}
\begin{bmatrix}
M(1) \\
M(2) \\
\vdots
\end{bmatrix}
\begin{bmatrix}
v_{\rho,i} \\
v_\rho(u_{ij}) + v_\rho(u_{iu})
\end{bmatrix} + \begin{bmatrix}
\epsilon'_{\rho,ur}(1) \\
\epsilon'_{\rho,ur}(2) \\
\vdots
\end{bmatrix}
\]

This component, corresponding to a phase center offset in position/time, is completely unobservable by our self-calibration approach, which solves this system of equations. Essentially, when we solve for the antenna model coefficients \( v_{\rho,i} \), we will have solved for the complete antenna calibration information except that we will not know to which phase reference point \( i \) this data corresponds.

From simulations results later in this chapter, it becomes clear that this is the only unobservable component given a sufficient number of measurements. Using (7.109), one can see that this component has only four dimensions even though \( v \) has significantly more degrees of freedom. To resolve this issue, we first use the estimated antenna-induced pseudorange bias to correct the GNSS receiver measurements. As a result, the remaining phase center offset becomes completely observable when GNSS receiver measurements are combined with inertial measurements. This will be the key to measuring the remaining unknown component and ultimately achieving a complete calibration.

155
7.6 Isolating the Antenna-Induced Carrier Phase Biases

Thus far, we have only dealt with estimation of antenna-induced pseudorange biases. Conceptually, everything discussed in the previous section also applies to antenna-induced carrier phase biases as well. For example, in the equation for the differential carrier phase measurements,

\[ \phi_{u\zeta}(n) \approx G_{\zeta}(n)u_{u\zeta}(n) + N_{u\zeta}^*(n) + a_{\phi,u}(n) + \epsilon_{\phi,ur}(n), \] (7.111)

the antenna-induced carrier phase biases can be decomposed into components \( u_{\xi u}(n) \) and \( q_{\phi,u}(n) \),

\[ \phi_{u\zeta}(n) \approx G_{\zeta}(n)u_{\xi}(n) + N_{u\zeta}^*(n) + q_{\phi,u}(n) + \epsilon_{\phi,ur}(n). \] (7.112)

Furthermore, the measurements can be processed by the projection matrix \( Q(n) \),

\[ q_{\phi,u}(n) = Q(n)\phi_{u\zeta}(n) \] (7.113)
\[ = Q(n)N_{u\zeta}^*(n) + Q(n)M(n)v_{\phi,u} + \epsilon'_{\phi,ur}(n). \] (7.114)

The form of (7.116) is very similar to (7.103); however, we are forced to go through extra steps in order to overcome the additional ambiguity vector. If we assume that carrier lock is maintained once a satellite signal is acquired, then the integer cycles can be tracked overtime. This removes the dependence of \( N_{u\zeta}^* \) on time \( n \) except that a new initial ambiguity is created whenever a new satellite is acquired during the course of calibration. Under such a situation, the measurement equation could be given the following form

\[ q_{\phi,u}(n) = Q(n)\phi_{u\zeta}(n) \] (7.115)
\[ = Q(n)\begin{bmatrix} M(n) & \mathbf{I} \end{bmatrix} \begin{bmatrix} v_{\phi,u}^* \\ N_{u\zeta}^* \end{bmatrix} + \epsilon'_{\phi,ur}(n). \] (7.116)
Unfortunately, limitations in our current simulation capabilities prevent accurate simulation of carrier phase acquisition, tracking, and integer ambiguities. As a consequence, an simulation of measurements based on (7.116) cannot be provided. Rather, the measurement model used for simulation results in this chapter will be based on the simplified model

\[ q_{\phi,u}(n) = Q(n)M(n) v_{\phi,u} + \epsilon'_{\phi,ur}(n), \quad (7.117) \]

with the understanding that further work is necessary to test and verify methods for antenna-induced carrier phase bias estimation.

7.7 Antenna Calibration Algorithm

Having defined our antenna model in Section 7.3 and our measurement model in Section 7.4 and 7.5, we are now in a position to formally state our self-calibration algorithm in well-defined terms. While the specific implementation will be discussed in the next section, the algorithm is fundamentally composed of the following four steps:

**Step 1: Preprocess Measurements to Isolate Antenna-Induced Effects**

Standard differential measurements are streamed to the self-calibration module. An estimated differential position is also sent to the module. Using the procedure described in Section 7.4.2, this differential position is used to shorten the baseline and improve the linearization of measurement model. Next, the procedures described in the Sections 7.5 and 7.6 is used to preprocess the differential pseudorange and carrier phase measurements in order to produce a measurement sequence that is dependent solely on the antenna-induced biases. This measurement model from (7.103) and
\[ q_{\rho,u}(n) = Q(n)M(n) v_{\rho,u} + \epsilon_{\rho,ur}(n) \quad (7.118) \]
\[ q_{\phi,u}(n) = Q(n)M(n) v_{\phi,u} + \epsilon_{\phi,ur}(n) . \quad (7.119) \]

**Step 2: Solve for Antenna Calibration Data**

Having created a linear measurement model with Gaussian noise, it is now straightforward to solve for the antenna model coefficients. The simplest way is to collect multiple measurements and form the system of equations
\[
\begin{bmatrix}
q(1) \\
q(2) \\
\vdots
\end{bmatrix} =
\begin{bmatrix}
Q(1) & \cdots & M(1) \\
Q(2) & \cdots & M(2) \\
\vdots & \ddots & \vdots
\end{bmatrix}
\begin{bmatrix}
v(1) \\
v(2) \\
\vdots
\end{bmatrix}
\]

which has a well known form of
\[ b = Ax. \quad (7.121) \]

For computer simulations or post-processing, one can solve the above system directly using the following optimization problem
\[
x_0 = \arg\min_x \|Ax - b\|^2 + \|\gamma x\|^2. \quad (7.122)\]

Since \( A \) will be underdetermined in many cases, a damping factor \( \gamma \) will be used to ensure that undetermined component (in particular, the phase center offset described in Section 7.5.3) is minimized since it is likely to be small. \( A \) is sparse and large, so it can be iteratively solved using an efficient conjugate gradient method. If the noise varies on each measurement, we could include weighting in the minimization. This provides for a simple way to solve for the antenna biases, but is insufficient for real-time systems. Section 7.7 will provide an iterative implementation of this minimization.
Step 3: Estimate User Position/Time Based on Corrected Antenna Calibration Data

As discussed in Section 7.5.3, our estimated antenna calibration data has an unknown phase center offset (in position and time). This makes sense since a phase center offset or clock bias is completely ambiguous with notions of position and time of the user receiver. Unfortunately, each GNSS signal type will have independent antenna calibration information and each will have a unique phase center offset and clock bias.

In order to resolve this issue, the antenna-calibration data found in Step 2 should be used to create a sequence of corrected differential measurements. Based on these corrected measurements, the differential position/time of the user should be found using each signal type in isolation. That is, only measurements from that signal are used in the solution. As a result, each position/time solution will be biased by a unique phase center offset. In order to resolve this phase center offset, we must utilize measurements from the IMU.

Step 4: Solve for Remaining Phase Center Offset of Antenna Data

A phase center offset is completely observable in a GNSS/INS receiver. Using a loosely-coupled Kalman filter, the phase center offset relative to the IMU body coordinate system (i.e. the “lever arm”) is easily and accurately found so long as the platform undergoes a minor change in attitude. This estimate will allow us to remove this ambiguity from our calibration and essentially make the antenna reference point of our calibration data correspond to the IMU. Since the location of the IMU is known, our antenna calibration becomes complete and absolute with respect to position.
At this point, all that would remain is an unknown clock bias for each signal type’s antenna calibration data. If a differential clock estimate could be formed by each signal type simultaneously, then it would be possible to remove relative clock bias between them. However, a single, antenna-induced clock bias still would remain for the system as a whole. Since the clock bias is caused by the antenna alone, it should be sufficiently small for most needs. However, this limited study will not go into more detail regarding this clock aspect.

7.8 Implementation

In the antenna calibration algorithm described above, Step 2 performs a significant amount of processing in order to estimate the antenna model coefficients. This section will develop an implementation based on recursive least-squares (RLS) in order to demonstrate that it is possible to perform this antenna calibration in a computationally efficient manner suitable for real-time processing. This simple, proof-of-concept implementation has a number of notable aspects. It is iterative so each measurement is used to update the antenna manifold and then discarded, reducing the storage requirements associated with post-processing. There are well-documented approaches for efficient implementation. If needed, an exponential forget factor could be utilized to provide a slow tracking capability in order to handle slow variation in the antenna model.

To begin, (7.103) and (7.117) will be written in a common form

\[ q(n) = C(n)v + \epsilon(n) \]  

where the measurement matrix \( C(n) \) is known

\[ C(n) = (I - P_{G_c}(n))M(n). \]
The coefficients $v$ are chosen at time $n$ to minimize the sum of weighted error since initialization ($n = 0$),

$$E(n) = \sum_{j=0}^{n} \lambda^{n-j} \|q(j) - C(j)v\|^2$$  \hspace{1cm} (7.125)

The coefficient vector obtained by minimizing (7.125) is denoted $v(n)$ and provides the optimum least square error at time $n$:

$$v(n) = \arg\min_v E(n)$$ \hspace{1cm} (7.126)

For our purposes, this study will use the conventional RLS implementation. Many techniques are available for improved numerical accuracy or computational efficiency, but are more difficult to explain. First, the system is initialized using

$$v(0) = 0$$ \hspace{1cm} (7.127)

$$P(0) = \delta^{-1}I$$ \hspace{1cm} (7.128)

where $\delta$ is a small value and $P$ is the covariance matrix of the estimation error. Although more meaningful initializations are possible. If we had some Gaussian uncertainty model, we could use the mean as the initialization and covariance matrix. At each iteration, we calculate the following values

$$G(n) = P(n-1)C^H(n)$$

$$G(n) = \tilde{G}(n)\left( C(n)\tilde{G}(n) + \lambda I + \Sigma(n) \right)^{-1}$$ \hspace{1cm} (7.129)

$$e(n) = q(n) - C(n)v(n-1)$$ \hspace{1cm} (7.130)

$$v(n) = v(n-1) + G(n)e(n)$$ \hspace{1cm} (7.131)

$$P(n) = \lambda^{-1}\left( P(n-1) - G(n)\tilde{G}(n) \right)$$ \hspace{1cm} (7.132)
where (7.129) is the gain calculation, (7.130) is the innovation sequence, (7.131) is the coefficient updating, and (7.132) is the correlation matrix updating.

As it is, the conventional RLS algorithm outlined above is not computationally feasible to implement with our abnormally large coefficient vector. To improve its efficiency, we must take advantage of the fact that the matrix $M$ is sparse and performs a simple indexing operation. The following steps can be performed without directly implementing the matrix operations:

\[
P'(n-1) = M(n)P(n-1)M^H(n) \tag{7.133}
\]

\[
P''(n-1) = P(n-1)M^H(n) \tag{7.134}
\]

\[
v'(n-1) = M(n)v(n-1) \tag{7.135}
\]

Based on these values, the rest of the operations can be reformulated and made considerably more efficient,

\[
C(n) = (I - PG_\zeta(n)) \tag{7.136}
\]

\[
A(n) = \left( C(n)P'(n-1)C(n) + \lambda I + \Sigma(n) \right)^{-1} \tag{7.137}
\]

\[
\bar{G}(n) = P''(n-1)C(n) \tag{7.138}
\]

\[
G(n) = \bar{G}(n)A(n) \tag{7.139}
\]

\[
e(n) = q(n) - C(n)v'(n-1) \tag{7.140}
\]

\[
v(n) = v(n-1) + G(n)e(n) \tag{7.141}
\]

\[
P(n) = \lambda^{-1}\left( P(n-1) - G(n)\bar{G}(n) \right) \tag{7.142}
\]

The matrix inverted in (7.137) is $K(n)xK(n)$, where $K(n)$ is typically 8-12 so it is fairly efficient. However, the last operation (the covariance matrix update) still requires $O(J^2)$. To overcome this, we are forced to take a sub-optimal approach and
assume the matrix $P(n)$ has a certain structure. The simplest structure would be to assume it is diagonal. In this case (7.133), (7.134) and (7.142) no longer depend on the number of coefficients but rather the number of visible satellites. Most likely, this will result in a trade-off of increased noise and suboptimal convergence, but we will not explore these details. The next section shows simulation results using this algorithm, and these results indicate that this approximation does not prevent the algorithm from operating. On a high-end desktop PC, the algorithm is able to process simulated measurements at significantly above real-time speeds. Based on this, we believe it would be computationally practical to implement the actual antenna calibration module using this algorithm.

### 7.9 Example Results

Simulations were performed to demonstrate that single element antenna can be calibrated using the proposed method during navigation. The antenna is a single RHCP crossed-slot mounted on a simple aircraft model shown in Fig. 7.5. The antenna pattern was simulated at the GPS L1 band (1.575 GHz) using the computational electromagnetics program NEWAIR [86]. The proposed calibration algorithm does not rely on any specific antenna or platform in order to work. In our coordinate system, $\phi$ is the azimuthal angle and $\theta$ is the elevation angle, where $\theta = 0$ corresponds to boresight. Only a GPS P-code signal will be used. A full software receiver was not utilized for these simulations. Rather, the measurement model derived earlier was directly used. Fig. 7.6 shows the code delay and carrier phase biases caused by the antenna over the entire upper hemisphere. The amplitude of a received signal relative to an isotropic antenna is also shown. The figure depicts the antenna on two
separate platforms. The first case corresponds to the antenna mounted on an infinite ground plane, and the second corresponds to the antenna on the aircraft. It is clear that scattering from aircraft structures causes the otherwise smooth pattern to vary rapidly with angle.

Motion of the aircraft was simulated using a simple, smoothed random walk for both position and orientation as plotted in Fig. 7.7. While this motion is not particularly realistic, it is sufficient for this study, and the large variation in orientation allows for reception of the satellite signals from many different angles. The GPS constellation of 31 satellites was simulated using custom satellite prediction software. The locations of the satellites were combined with the user motion in order to accurately determine the relative angle to satellites from the receiver. For this study, receiver measurements will be taken at a rate of 1 Hz. Fig. 7.8 show sky plots of the visible satellites accumulated over different lengths of time. At the end of our calibration period of 4 hours, the receiver has been exposed to signals from practically every angle. Only a few small gaps still remain, particularly at high-elevations. Fig. 7.9 provides information regarding the total number of measurements and the

*Figure 7.5: Simulated crossed-slot antenna on aircraft model.*
amount of angular coverage. After 4 hours, over 90% coverage is achieved with most regions having multiple measurements.

Pseudorange and carrier phase measurements were generated based on the models presented earlier. The calibration algorithm was implemented using RLS assuming a static antenna model (λ = 1). To begin, the results of code delay calibration will be studied in the absence of noise after measurements have been received for 4 hours. Fig. 7.10 contains a number of results. First, Fig. 7.10 (a) shows the original code delay bias of the antenna. After 4 hours, measurements have been received for most angular regions except for a small hole at zenith. The regions that do not have measurements are depicted in Fig. 7.10 (b). Our proposed calibration algorithm was used to iteratively process these measurements. Fig. 7.10 (c) shows the estimated code delay bias before phase center correction is applied. The difference between this estimate and the original code delay bias is shown in Fig. 7.10 (d). Next, we estimated the phase center offset directly, and used it to correct the estimated code delay bias. The new estimated bias and its residual error is shown in Fig. 7.10 (e) and (f), respectively. It is clear that almost all of the code delay bias has been estimated. However, we note that there are a number of tiny angular regions where the bias has not been sufficiently estimated. We expect most of these regions to contain zero or very few measurements. In order to provide a more detailed view, Fig. 7.11 repeats the exact same sequence of subfigures for a single elevation cut corresponding to (φ = 45°).

Next, the carrier phase calibration was studied in the absence of noise after measurements have been received for 4 hours. Fig. 7.12 depicts the same sequence of
subfigures that was shown for the code delay bias estimation. We see that the proposed calibration algorithm can also be applied to the carrier phase estimation with similar results. Fig. 7.13 shows a single elevation cut corresponding to \( \phi = 75^\circ \). Clearly, the algorithm is able to capture the precise variations with respect to angle.

The convergence of the code and carrier phase bias estimates over time was also studied. Fig. 7.14 shows the mean absolute error of each estimate over the entire upper hemisphere. Three cases are shown. In the first case, no noise is present, and we see both code delay and carrier phase estimates linearly improve with the number of measurements. In the second case, labeled ‘Noise 1’, noise has been added to the measurements with the standard power found in precision receivers \( \sigma_p = 0.5 \text{ m} \) and \( \sigma_\phi = 0.025 \text{ cycles} \) [1]. Additionally, the measurements for this case are provided at 10 Hz. The amount of error in the antenna model estimate has increased and is no longer linear; however, it is expected that, given additional time, this noise would decrease sufficiently. In the third case, labeled ’Noise 2’, the same noise is added but measurements are provided at a rate of 1 Hz. Thus, there are fewer measurements and more noise in the final estimate. In this case, the error is quite significant, and the calibration algorithm actually increases the error for a portion of the time. The reason for this is that the original calibration data was initialized to zero and provides a closer estimate than a limited number of noisy measurements.

There are a number of options for overcoming the effects of noise. Basis functions that occupy a larger angular region could be used, but at the cost of not being able to represent rapid variations. The approximations made in the derivation of our RLS algorithm could be improved at the cost of increased computational complexity. Or, more measurements could be taken by increasing the measurement rate or increasing
the calibration time. Additionally, some initial uncertainty about the antenna cali-
bration data could be used to help prevent noisy measurements from increasing the
antenna-induced bias error.
Figure 7.6: Code delay bias (top), carrier phase bias (middle) and amplitude (bottom) for the antenna on a groundplane (left) and on an aircraft (right).
Figure 7.7: True position (top) and orientation (bottom) of the simulated user receiver.
Figure 7.8: Sky plots of the GPS satellite angle-of-arrivals – accounting for user position and orientation – from left to right and top to bottom: 5 minutes, 15 minutes, 30 minutes, 1 hour, 2 hours, and 4 hours.
Figure 7.9: The total number of measurements versus time (top), the percent upper hemisphere coverage of those measurements (bottom).
Figure 7.10: Code delay calibration after 4 hours. The original bias (a), the angular regions that don’t have measurements (b), the estimated phase (c) and residual error (d), and the estimated phase with phase center offset correction (e) and residual error with phase center offset correction (f).
Figure 7.11: Code delay calibration after 4 hours for an elevation cut ($\phi = 45^\circ$). The original bias (top left), the angular regions that don’t have measurements (top right), the estimated phase and residual error (middle), and the estimated phase and residual error with phase center offset correction (bottom).
Figure 7.12: Carrier phase calibration after 4 hours. The original bias (a), the angular regions that don’t have measurements (b), the estimated phase (c) and residual error (d), and the estimated phase with phase center offset correction (e) and residual error with phase center offset correction (f).
Figure 7.13: Carrier phase calibration after 4 hours for an elevation cut ($\phi = 75^\circ$). The original bias (top left), the angular regions that don't have measurements (top right), the estimated phase and residual error (middle), and the estimated phase and residual error with phase center offset correction (bottom).
Figure 7.14: Mean absolute error after calibration for the code delay (top) and carrier phase (bottom) for three cases: no noise, standard noise levels at 10 Hz (Noise 1) measurement rate, and standard noise levels at 1 Hz (Noise 2).
7.10 Summary & Conclusions

Fundamentally, the goal of this chapter was to demonstrate the use of precision GNSS measurements to continually update and improve the quality of stored antenna calibration data in real-time as the receiver navigates. Toward this goal, this chapter has delivered a proof-of-concept self-calibration algorithm suitable for precision GNSS receivers. There are some notable achievements. First, the proposed algorithm is able to calibrate the antenna without requiring accurate position/time estimation. Therefore, it is unique as compared to traditional field calibration procedures in which the receiver position and clock are highly restricted and controlled. The algorithm is encapsulated into a simple add-on module that uses standard differential measurements rather than requiring direct processing of received signals. The algorithm iteratively updates the antenna calibration data rather than accumulating measurements for post-processing. Finally, the algorithm uses a high-resolution antenna model suitable for antennas on complex platforms, yet it can still be efficiently implemented for real-time processing. The next chapter will discuss how to extend this algorithm so that it is applicable not only to single-element antennas but also adaptive antenna arrays.
CHAPTER 8

Self-Calibration Algorithm for GNSS Antenna Arrays

The previous chapter developed a self-calibration algorithm for single-element GNSS antennas. However, if an antenna to be calibrated is an array equipped with space-time adaptive processing, then this calibration is not sufficient. This is because calibration of an adaptive array requires information about the entire antenna array manifold. This chapter provides a short study of how a recently proposed GNSS adaptive array calibration approach [32] can be extending to perform self-calibration. Standard differential GNSS receiver measurements will be processed as the weights of the adaptive filter are gradually adjusted. These measurements will be combined in order to estimate the antenna array manifold. As with the previous chapter, this calibration can be performed during navigation while the position and clock of the receiver is unknown. Since the antenna is calibrated in-situ, mutual coupling between antenna elements and platform effects will be incorporated automatically. Section 8.1 begins by describing the motivation behind development of a new approach. Then, Section 8.2 provides an overview of the extended self-calibration algorithm. Section 8.3 describes the antenna array manifold model, and Section 8.4 and 8.5 describe how traditional GNSS receiver measurements can be used to estimate it. Section 8.6 shows results obtained from simulations for an adaptive antenna array mounted on
an aircraft. During interference suppression, the original antenna-induced biases for this system are shown to be several meters in code delay and hundreds of degrees in carrier phase. After application of the proposed self-calibration approach, these biases are greatly reduced given sufficient measurements.

8.1 Introduction

Thus far, Chapters 4, 5 and 6 have developed novel techniques and algorithms for GNSS adaptive antennas that optimally suppress interference without introducing errors into the received satellite signals. These features define the ideal GNSS adaptive antenna; however, these techniques have been developed under the assumption that the receiver has perfect knowledge of the reference correlation vector \( s(\tau, \theta, \phi) \). As discussed earlier in Chapter 4, \( s \) is a \((KN \times 1)\) vector where \( K \) is the number of antenna elements in the array and \( N \) is the number of taps in each adaptive filter. The entry of the vector corresponding to the \( k \)th element and the \( n \)th tap is given by

\[
[s_k(\tau, \theta, \phi)]_n = \int G_d(f) A_k^*(\theta, \phi, f) F_k^*(f) e^{j2\pi f (nT_0 + \tau - \tau_{ref})} df
\]  

(8.1)

which depends on the normalized signal spectrum \( G_d(f) \), the front-end electronics \( F_k(f) \), and the response of the \( k \)th antenna element \( A_k(\theta, \phi, f) \). Therefore, previous chapters have indirectly assumed that the antenna array manifold is known perfectly. Unfortunately, there will often be a “mismatch” between the actual and used manifolds. Thus, while the algorithms we have proposed are very practical to implement, it is impractical to assume that perfect antenna manifold knowledge is available. Precision GNSS receivers require a correspondingly accurate antenna calibration. Although it is commonly believed that small errors in the manifold will produce negligible errors in the GNSS solution, this belief is incorrect. Even small
errors can contribute significantly during interference suppression in certain scenarios. Consequently, we require that antenna manifold be known very accurately in order for our antenna-induced bias mitigation approaches to be successful.

Recently, a new field calibration approach has been proposed for adaptive antenna arrays [32]. This proposed approach is an extension of traditional field calibration techniques for single-element GNSS antennas and requires that the user position and clock are precisely known. However, it is able to estimate the entire adaptive antenna array manifold using only standard GNSS receiver measurements. This is accomplished by varying the weights of the adaptive filters and performing a standard calibration for each choice of weights. The results of these calibrations are then combined in order to estimate the antenna array manifold. Essentially, this chapter will utilize this approach to extend our self-calibration algorithm and make it applicable to adaptive antenna arrays.

8.2 Overview of Proposed System

Fig. 8.1 depicts the proposed extension to the self-calibration logic that was introduced in the previous chapter. Once again, the extension is located in a self-contained module and is very loosely-integrated with existing STAP-based GNSS receivers. It should be emphasized that the received signals are not directly processed by any of the antenna calibration logic. Rather, its inputs and outputs are already standard in existing GNSS receiver designs and operate at very low rates. This allows the module to be practical and efficient to implement. During the absence of interference, the weights of the adaptive filter are controlled by the array calibration module. The weights are varied among a fixed set of possible weights whose values
will be discussed later in this chapter. It is implied that, if interference appears, then the STAP processor would pause calibration and resume its standard interference suppression (which can be implemented using any method).

For each set of filter weights, a self-calibration is performed according to the description in Chapter 7. The standard differential measurements from the GNSS receiver (pseudorange, carrier phase, and C/N₀) are streamed to the calibration logic which iteratively updates locally stored antenna calibration data. Thus, there is a separate set of calibration data for each set of weights. This calibration occurs continuously, even during navigation while the system is in motion and its relative position not known. An independent process in the array calibration logic combines

![Diagram](image)

*Figure 8.1: An extension of the self-calibration logic proposed in Chapter 7 in order to make it applicable to adaptive antenna arrays.*
each separate set of calibration data in order to estimate the antenna array manifold, which is stored independently in the receiver. This processing occurs for a single angular region at a time, so its computational complexity can be distributed evenly over the duration of the calibration.

Using methods describe in Chapter 6, the manifold data can be used to provide antenna-induced bias corrections to the GNSS receiver. Alternatively, biases could be mitigated using the special adaptive filtering algorithm described in Chapter 5. Since this has already been discussed, this chapter will only focus on the manifold estimation aspect. We must emphasize that calibration of the antenna does not require accurate position/time estimation. Once again, we will assume that there exists some source of external control that determines when the system should be calibrating and when it should not be.

8.3 Antenna Array Manifold Model

The antenna array manifold is the response of each antenna element over the GNSS frequencies over all incident signal angles. The response $A_k(f, \theta, \phi)$ of the $k$th antenna element for some particular frequency band in the array will be approximated in frequency on some finite basis $\{\Gamma_l(f)\}$ for $l = 1..L$ and in angle on some finite basis $\{\Omega_j(\theta, \phi)\}$ for $j = 1..J$ as

$$A_k(f, \theta, \phi) = \sum_{j=1}^{J} \sum_{l=1}^{L} \alpha_{j,k,l} \Gamma_l(f) \Omega_j(\theta, \phi).$$

The response of each element is understood to be relative to some known, common antenna reference point. The complete characterization of the antenna would require a set of functions for each GNSS each frequency band. Acquisition of these functions
would constitute a complete, absolute calibration of a GNSS adaptive antenna array. This is significantly more information than the single element GNSS antenna in Chapter 7 required.

GNSS signals occupy relatively narrow bandwidths, and the antenna response tends to vary slowly and smoothly versus frequency. Consequently, it can be accurately approximated using only a few basis functions, which we will chose to be polynomials \( \{ \Gamma_l(f) \} \) for \( l = 1 \ldots L \) where

\[
\Gamma_l(f) = \frac{(f - f_0)^{(l-1)}}{(l - 1)!}.
\]  

(8.3)

The number of basis functions \( L \) will depend on the signals occupying the particular band.

We will use the angular basis defined in Section 7.3, so each \( j \) corresponds to a finite angular region. The set of coefficients for a particular angular region \( j \) can be stacked into vectors

\[
a_{j,k} = \begin{bmatrix} \alpha_{j,k,1} \\ \vdots \\ \alpha_{j,k,L} \end{bmatrix}, \quad a_j = \begin{bmatrix} a_{j,1} \\ a_{j,2} \\ \vdots \\ a_{j,K} \end{bmatrix},
\]  

(8.4)

where \( a_{j,k} \) and \( a_j \) represent the response for the \( k \)th element and the entire array, respectively. The set of vectors \( \{ a_j \} \) for \( j = 1 \ldots J \) represent the complete antenna array manifold model (that is, the complete calibration information) for a single GNSS frequency band.

### 8.4 Collecting Measurements

In order to estimate the antenna manifold, we must first understand how it affects conventional GNSS receiver measurements. As discussed in Chapter 3, an expression
for the cross-correlation performed by the tracking loops in a GNSS receiver is given by

\[ R_{yd}(\tau, \theta, \phi) = \int G_d(f) \sum_{k=1}^{K} W_k(f) F_k(f) A_k(f, \theta, \phi) e^{j2\pi f(\tau - \tau_{ref})} \, df \]  

(8.5)

and depends on the frequency response \( W_k(f) \) for the \( k \)th adaptive filter

\[ W_k(f) = \sum_{n=1}^{N} w_{kn} e^{-j2\pi fT_0(n-1)}. \]  

(8.6)

where \( N \) is the adaptive filter length. The code phase, carrier phase, and amplitude biases caused by an adaptive antenna array follow from this expression as three continuous functions of incident angle \((\theta, \phi)\) as described in Section 7.3:

\[ a_\rho(\theta, \phi) : \arg\max_{\tau} |R_{yd}(\tau, \theta, \phi)| \]  

(8.7)

\[ a_\phi(\theta, \phi) : \angle R_{yd}(\tau_0, \theta, \phi) \]  

(8.8)

\[ a_g(\theta, \phi) : |R_{yd}(\tau_0, \theta, \phi)|, \]  

(8.9)

the first two of which are relative to the antenna reference point. These biases can be combined to yield

\[ R_{yd}(a_\rho(\theta, \phi), \theta, \phi) = a_g(\theta, \phi)e^{ja_\phi(\theta, \phi)} \]  

(8.10)

If we consider the front-end response as part of the antenna response, then (8.5) can be combined with (8.10) to produce

\[ a_g(\theta, \phi)e^{ja_\phi(\theta, \phi)} = \int G_d(f) \sum_{k=1}^{K} W_k(f) A_k(f, \theta, \phi) e^{j2\pi f(a_\rho(\theta, \phi) - \tau_{ref})} \, df \]  

(8.11)

In (8.11), the normalized signal spectrum \( G_d(f) \) is known, the adaptive filter responses \( W_k(f) \) are controllable, and \( a_\rho(\theta, \phi), a_\phi(\theta, \phi), \) and \( a_g(\theta, \phi) \) are measurable using our self-calibration approach. Therefore, it is clear that it should be possible to solve for the antenna manifold \( A_k(f, \theta, \phi) \) for \( k = 1, \ldots, K \).
Our procedure for estimating the antenna manifold is as follows. First, we define a set of adaptive filter weights, where \( \mathbf{w}(m) \) denotes the \( m \)th set of weights for \( m = 1..M \). For each weight vector, we measure (via self-calibration) the three direction-dependent antenna-induced bias functions: \( a_{\rho}(\theta, \phi) \), \( a_{\phi}(\theta, \phi) \), and \( a_{g}(\theta, \phi) \). This calibration data from each set of weights is then combined using the procedure described in the next section in order to estimate the complete antenna array manifold.

In our case, we will use the set of filter weights described in [32]. For \( K \) antenna elements and \( N \) taps, \( 2KN \) weight vectors are chosen according to the equation

\[
\mathbf{W} = \begin{bmatrix} \mathbf{I} \\ c_1 \mathbf{I} \end{bmatrix}^T + \begin{bmatrix} 0 & \ldots & 0 & c_2 & 0 & \ldots & 0 \\ \vdots \\ 0 & \ldots & 0 & c_2 & 0 & \ldots & 0 \end{bmatrix}
\]

where \( 2KN - K \) of the weight vectors are unique and the others discarded. The weight vectors correspond to the columns of \( \mathbf{W} \) and \( c_1 \) and \( c_2 \) are complex (not purely real or imaginary) numbers with \( c_1 \neq c_2 \). This is just one choice for a set of weights. In fact, there are potentially many different choices that would render \( A_k(f, \theta, \phi) \) observable. However, since this study is only focused on demonstrating a proof-of-concept system, this set of weights will be sufficient.

### 8.5 Solving for the Antenna Array Manifold

This section will take the accumulated antenna calibration data for each set of weights and combine it to estimate for the antenna array manifold. This derivation is based on work presented in [32]. For the sake of clarity, we will formulate it as a continuous function of \((\theta, \phi)\), and, at the end, convert it into coefficients on our angular basis function set.
To begin, the correlation vector \( \mathbf{s}(\tau, \theta, \phi) \) from (8.1) can be decomposed into \( K \) subvectors that correspond to each antenna element,

\[
\mathbf{s}(\tau, \theta, \phi) = \begin{bmatrix} s_1(\tau, \theta, \phi) \\ \vdots \\ s_K(\tau, \theta, \phi) \end{bmatrix} \tag{8.13}
\]

The components of each \( N \times 1 \) subvector \( s_k \) are dependent on the \( k \)th antenna response, and (8.2) can be incorporated into it to yield

\[
[s_k(\tau, \theta, \phi)]_n = \int A_k^*(f, \theta, \phi) G_d(f) e^{j2\pi f(nT_0 + \tau - \tau_{ref})} \, df
\tag{8.14}
\]

\[
= \sum_{l=1}^{L} \alpha_{k,l}(\theta, \phi) \int \Gamma_l(f) G_d(f) e^{j2\pi f(nT_0 + \tau - \tau_{ref})} \, df
\tag{8.15}
\]

\[
= \mathbf{m}_n^T(\tau) \mathbf{a}_k^*(\theta, \phi),
\tag{8.16}
\]

where we have introduced the \( L \times 1 \) vector \( \mathbf{m}_n \)

\[
\mathbf{m}_n(\tau) = \begin{bmatrix} \int \Gamma_1(f) G_d(f) e^{j2\pi f(nT_0 + \tau - \tau_{ref})} \, df \\ \vdots \\ \int \Gamma_L(f) G_d(f) e^{j2\pi f(nT_0 + \tau - \tau_{ref})} \, df \end{bmatrix}
\tag{8.17}
\]

and an \( L \times 1 \) vector \( \mathbf{a}_k \) which contains the antenna response coefficients

\[
\mathbf{a}_k(\theta, \phi) = \begin{bmatrix} \alpha_{k,1}(\theta, \phi) \\ \vdots \\ \alpha_{k,L}(\theta, \phi) \end{bmatrix}
\tag{8.18}
\]

It follows from (8.16) that \( s_k \) can be represented as

\[
\mathbf{s}_k(\tau, \theta, \phi) = \begin{bmatrix} \mathbf{m}_1^T(\tau) \\ \vdots \\ \mathbf{m}_N^T(\tau) \end{bmatrix} \mathbf{a}_k^*(\theta, \phi) = \mathbf{M}(\tau) \mathbf{a}_k^*(\theta, \phi)
\tag{8.19}
\]

and, subsequently, allows the correlation vector to be written as

\[
\mathbf{s}(\tau, \theta, \phi) = \begin{bmatrix} s_1(\tau, \theta, \phi) \\ s_2(\tau, \theta, \phi) \\ \vdots \\ s_K(\tau, \theta, \phi) \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\tau) & 0 & \ldots & 0 \\ 0 & \mathbf{M}(\tau) & \vdots \\ \vdots & \vdots & \ddots \\ 0 & \ldots & 0 & \mathbf{M}(\tau) \end{bmatrix} \begin{bmatrix} \mathbf{a}_1(\theta, \phi) \\ \mathbf{a}_2(\theta, \phi) \\ \vdots \\ \mathbf{a}_K(\theta, \phi) \end{bmatrix}^*
\tag{8.20}
\]

\[
= \mathbf{Q}(\tau) \mathbf{a}^*(\theta, \phi),
\tag{8.21}
\]

186
where we have implicitly defined new matrices $M$ and $Q$ that are $N \times L$ and $K \times K_L$, respectively. Finally, the cross-correlation function can be related directly to the coefficients of the antenna response using,

$$ R_{yd}(\tau, \theta, \phi) = w^T s^*(\tau, \theta, \phi) $$

$$ = w^T Q^*(\tau) a(\theta, \phi). $$

Using $M$ different adaptive filter weights, the antenna calibration data for each choice of weights is determined and used to determine the correlation function measurement $R_{yd}^{(m)}$ using (8.10). These measurements and weights are arranged in vectors

$$ r(\theta, \phi) = \begin{bmatrix} R_{yd}^{(1)}(\tau_1, \theta, \phi) \\ \vdots \\ R_{yd}^{(M)}(\tau_M, \theta, \phi) \end{bmatrix}, V = \begin{bmatrix} w^T(1) Q^*(\tau_1) \\ \vdots \\ w^T(M) Q^*(\tau_M) \end{bmatrix} $$

where $\tau_m = a_p(\theta, \phi)$. This yields the measurement equation

$$ r(\theta, \phi) = V \ a(\theta, \phi) $$

where $V$ is an $M \times K_L$ matrix. (8.25) provides a simple linear equation which relates the unknown, desired antenna pattern coefficients $a$ to the known GPS measurements $r$. Furthermore, once the antenna coefficients have been found, they can be used to determine the reference correlation vector using

$$ s(0, \theta, \phi) = Q^*(0) a(\theta, \phi). $$

Measurements from multiple signal types are transmitted in the same bandwidth (i.e. GPS L1 signals C/A code, P-code, and M-code). Therefore, it is advantageous to solve for the antenna manifold coefficients rather than the reference correlation vector since they are independent of the SOI spectrum while coefficients of the reference correlation vector are not.
Once our angular basis functions have been incorporated into (8.25), the measurement equation simply becomes

\[ r_j = V a_j \quad (8.27) \]

where \( a_j \) was defined in (8.4) and \( r_j \) corresponds to measurements occurring within the specific angular region \( j \). One can solve the above system directly by minimizing the error \( \| V a_j - r_j \| \). \( V \) will be underdetermined (depending on the number of basis coefficients), so we will use a damping factor \( \gamma \) to reduce spurious estimates and solve the optimization problem

\[ a_j = \arg\min_a \| V a - r_j \|^2 + \| \gamma a \|^2. \quad (8.28) \]

For efficient implementation of this minimization, we refer to [32].

### 8.6 Example Results

Simulations were performed to demonstrate the feasibility of calibrating an adaptive antenna array using the proposed method during navigation. The simulation parameters are almost identical to the ones used in Chapter 7. However, instead of a single element antenna, the antenna is a 7-element array of RHCP crossed-slots. The array is mounted on a simple aircraft model shown in Fig. 8.2. A 7-tap STAP processor was used with a system bandwidth of 24 MHz. Again, the antenna pattern was simulated at the GPS L1 band (1.575 GHz) using the computational electromagnetics program NEWAIR [86]. A GPS P-Code signal is received using set of 91 different adaptive filter weight vectors. The antenna manifold is represented in frequency using 5 polynomial basis functions and in angle using the 1 deg. regions described in
Chapter 7. In order to simplify this particular analysis, the receiver measurements will be made in the absence of noise.

Two different calibrations methods were used. In the first method (Method 1), a self-calibration was performed for each set of weights using 4 hours of receiver motion and a 1 Hz measurement rate, as was done in Chapter 7. All of the individual sets of calibration data were obtained simultaneously for all adaptive filter weights and then combined to estimate the antenna array manifold. In a real implementation, the weights would need to vary in time; however, we will not simulate this aspect in this study. For the second method (Method 2), we will consider an ideal situation where a measurement is made for every angle. Since this method does not rely on platform motion in order to provide an adequate number of measurements, it is more suitable for observing the quality of the antenna manifold estimation method in isolation. However, it is also less realistic.

To begin, we will demonstrate that the antenna manifold can be estimated from standard receiver measurements, as explained in Section 8.4. Fig. 8.3 compares the

![Simulated 7-element circular array of crossed-slot antennas on an aircraft.](image)
true antenna manifold to the estimated manifold for a single direction \((\theta, \phi) = (60, 0)\). In this case, we are using Method 2 to perform the calibration. The magnitude and phase for each antenna element is shown. It is clear that the estimated antenna manifold accurately represents the true manifold. Fig. 8.4 shows similar results for a different angle, \((\theta, \phi) = (40, 100)\). This confirms that manifold is completely observable.

Fig. 8.5 shows the residual code delay bias and carrier phase bias before and after calibration using Method 1. The antenna array is operating in the absence of interference using the MMSE algorithm introduced in Chapter 3. The figure shows an elevation cut in the \(\phi = 30^\circ\) plane. Before calibration, the MMSE algorithm only has access to the antenna manifold data while the antenna was mounted on a ground plane. The resulting antenna manifold mismatch causes the observed bias errors. In particular, large oscillations are observed at lower elevations due to scattering from wings and stabilizers. After calibration, it is clear that the biases have been significantly reduced in areas where there were measurements. It is worth noting that the antenna calibration starts off with zero initial knowledge about the antenna array. As a result, it has poor performance in areas without measurements. Had the algorithm been initialized with the ground plane manifold data, we would see a sharp reduction in error at higher elevations.

Fig. 8.6 shows the same situation except that Method 2 has been used to calibrate the array. Although this method is idealistic since it uses measurements for every angle, we can consider it as the limit of the Method 1 as the number of measurements and the calibration duration becomes very large. It is clear that the code delay and carrier phase biases have been precisely estimated using this approach.
To provide additional information, Fig. 8.7 and Fig. 8.8 show the results for an elevation cut in the $\phi = 50^\circ$ plane. Similarly, Fig. 8.9 and Fig. 8.10 show the results for the $\phi = 120^\circ$ plane. All of the results are the same as before. Array calibration using Method 1 results in significantly reduced biases in regions with measurements. Array calibration using Method 2 completely mitigates the biases. Presumably, if noise were present, then the quality of the calibration would be degraded, and a considerable number of additional measurements would be needed to overcome it. However, the results presented here serve as a proof-of-concept that the approach is able to estimate the array manifold using calibration data obtained from the self-calibration approach introduced in Chapter 7.
Figure 8.3: Original antenna manifold in the \((\theta, \phi) = (60, 0)\) direction (left) and estimated antenna manifold (right).

Figure 8.4: Original antenna manifold in the \((\theta, \phi) = (40, 100)\) direction (left) and estimated antenna manifold (right).
Figure 8.5: Elevation cut in the $\phi = 30^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1).
Figure 8.6: Elevation cut in the $\phi = 30^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2).
Figure 8.7: Elevation cut in the $\phi = 50^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1).
Figure 8.8: Elevation cut in the $\phi = 50^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2).
Figure 8.9: Elevation cut in the $\phi = 120^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements for the moving platform (Method 1).
Figure 8.10: Elevation cut in the $\phi = 120^\circ$ plane of code delay bias (top) and carrier phase bias (bottom) using simulated measurements in an ideal case (Method 2).
8.7 Summary & Conclusions

This chapter provided a brief study about how an existing adaptive array calibration technique can be combined with our single-element self-calibration approach. It has demonstrated the feasibility of using standard GNSS receiver measurements to calibrate an adaptive antenna array during navigation. Based on [32], it was shown that the weights of an adaptive filter could be varied in such a way that it becomes possible to estimate the entire antenna array manifold from standard GNSS receiver outputs. Building upon the results of Chapter 7, we showed how this calibration could be performed without precise knowledge of the position/time of the user receiver. This proof-of-concept system provides evidence which supports the potential of self-calibration of GNSS adaptive antenna arrays.
CHAPTER 9

Summary & Future Work

Adaptive antenna arrays provide interference suppression capabilities for GNSS receivers. Current systems exhibit a number of limitations that prevent them from providing interference suppression and precision navigation simultaneously. For example, in the process of suppressing interference, adaptive antennas may inadvertently distort the GNSS signal and introduce bias errors into the receiver’s position and time estimates. Additionally, some systems produce sub-optimal interference suppression performance, which degrades the accuracy of the navigation solution. To overcome these limitations, a collection of novel adaptive antenna algorithms and techniques suitable for precision GNSS receivers were developed.

One major contribution of this research effort is the development of adaptive antennas that are optimal for GNSS applications. An adaptive filtering algorithm was developed that maximized receiver C/N₀ and achieves optimal suppression of interference. Additionally, two practical techniques were developed for mitigation of adaptive-antenna induced bias errors. Without such mitigation, the adaptive antennas introduce errors into the GNSS receiver measurements that are potentially too large for precision applications. The two proposed techniques take the form of a
special adaptive filter algorithm and additional receiver logic, both of which mathematically guarantee zero antenna-induced error even during interference suppression. The features of this optimal GNSS adaptive antenna were achieved with practical extensions to existing systems. They are all computationally efficient and straightforward to implement.

Another significant contribution of this research effort is the investigation into novel adaptive antenna array calibration algorithms. Since our proposed adaptive antenna algorithms require very accurate antenna manifold information, we wanted to develop a less time consuming calibration approach for obtaining this information. The goal was to use standard GNSS receiver measurements to reduce the uncertainty in the antenna calibration data and to perform this “on-the-fly” (during navigation). It is well-understood that this was a very ambitious goal; however, numerous achievements have been made. The proposed algorithm is able to calibrate the antenna without requiring accurate position/time estimation. The algorithm is encapsulated into a simple add-on module that uses standard differential measurements rather than requiring direct processing of received signals. The algorithm operates iteratively and in real time to update the antenna calibration data rather than accumulating measurements for post-processing. Finally, the algorithm uses a high-resolution antenna model suitable for antennas on complex platforms.

There are a few remaining issues that should be addressed by future work. One aspect that is lacking is experimental verification. Much of this work involved techniques that were only advantageous in the presence of interference, and experimental verification would pose obvious difficulties. However, experimental verification could
be warranted for verifying the practicality of our proposed adaptive antenna calibration approach. For the case of a single-element antenna, this could be done by collecting live data from a differential GNSS receiver and post-processing it in order to estimate antenna-induced biases. These biases could then be compared to chamber measurements. This would confirm that the calibration is of sufficient quality despite environmental effects and other sources of uncertainty. In the case of an antenna array, STAP hardware would be required; however, it would offer considerable credibility to the proposed approach.
BIBLIOGRAPHY


