

Single Antenna Interference Cancellation in Asynchronous GSM/GPRS Networks

by
Chung Chan

Submitted to the Department of Electrical Engineering and Computer
Science

in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

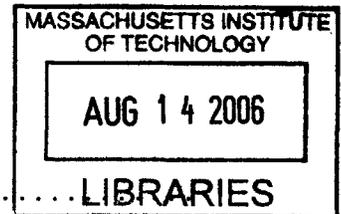
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

© Chung Chan, MMV. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis and to
grant others the right to do so.



Author
Department of Electrical Engineering and Computer Science
May 18, 2005

Certified by.....
Roland Rick
Principal Manager
VI-A Company Thesis Supervisor

Certified by.....
Lizhong Zheng
Assistant Professor
~~M.I.T. Thesis Supervisor~~

Accepted by
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Single Antenna Interference Cancellation in Asynchronous GSM/GPRS Networks

by

Chung Chan

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2005, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

In this project, we have proposed a decorrelator-based single antenna interference cancellation algorithm for the asynchronous GSM/GPRS network. The algorithm is tested according to the current SAIC/DARP performance requirement in the computer simulation, and is shown to give various gains in different test scenarios.

Thesis Supervisor: Roland Rick
Title: Principal Manager

Thesis Supervisor: Lizhong Zheng
Title: Assistant Professor

Acknowledgments

This project was conducted at Qualcomm from June to December 2004 while I was working in the GSM team of the Qualcomm CDMA Technology division. The thesis would not have been possible without the resources provided by Qualcomm and the VI-A Internship program at Massachusetts Institutes of Technology. I sincerely thank all the administrative staff, Professor Zahn, Gary Osumi, Kathy Sullivan and Danielle Guichard-Ashbrook.

I would like to thank my Qualcomm project manager Roland for his leadership and guidance. He made the task very clear for me by laying down the important achievement goals and helping us resolve numerous difficulties in the course of implementing the computer simulation.

I wish to acknowledge my co-worker, Raghu Challa, for his significant contribution on the Qualcomm SAIC solution. He came up with all the algorithms described in this thesis and implemented almost all the SAIC portion of the computer simulation. I am also amazed by his humble personality and his patience in teaching me the details of the algorithm.

I also like to thank Quei for spending significant time and effort on changing the computer simulation to adapt to the SAIC algorithm, Farrokh for patiently walking through the poorly written thesis draft with me everyday during my last week of work at Qualcomm, and Deirdre for sitting with me through the details of the computer simulation. I must also thank Deirdre for her well-documented alternative simulation that tremendously clarified the actual simulation program.

The layout and content of this thesis is a result of detailed review by my thesis supervisor, Professor Lizhong Zheng. His insistence on reducing the use of acronyms and including more descriptions on the mathematical equations has significantly made this thesis more readable.

Above all, I am most grateful for the support of my family in the course of writing the thesis. To them, I attribute every credit of my work.

Contents

1	Introduction	11
2	Background on GSM Transmission	15
2.1	GSM Normal Burst	15
2.2	Channel Coding for Traffic Channels	16
2.3	GMSK Modulation	18
2.4	Approximating GMSK with DBPSK	21
3	Background on GSM Signal Reception	23
3.1	Discrete-time Measurement Model without Interference	23
3.2	Multistage Joint Channel-Data Estimation	26
3.2.1	First-Stage LS Channel Estimation with Known Data	29
3.2.2	Second-Stage SO-MLSE of Data Burst with Known Channel	35
4	SAIC algorithm design	41
4.1	Discrete-time Measurement Model with Interference	41
4.2	Single Interferer Model	43
4.3	Proposed Decorrelator-based SAIC Algorithm	44
4.3.1	Matrix Representation of Decorrelation	44
4.3.2	Joint Decorrelator-channel-data Estimation	46
4.3.3	Least Squares Optimization of Decorrelator	49
5	Computer Simulation	55

A Background Theory	61
A.1 Channel Coding of Full Rate Traffic Channel	61
A.2 Amplitude Modulation Decomposition of GMSK Signal	63
A.3 Least Squares Estimation	66
A.4 Comparing CAZAC and LS Channel Estimates	68
A.5 Different Model Assumptions on the Single Interferer Model	70
Bibliography	73

List of Figures

1-1	Adjacent and co-channel interference at mobile station	12
2-1	Normal burst structure	16
2-2	GMSK modulation	19
2-3	Frequency pulse of the GMSK modulator	20
2-4	Logarithmic plot of the amplitude modulated pulses for GMSK	22
3-1	Probabilistic model of the GMSK signal	24
3-2	Probabilistic channel model without co-channel interference	24
3-3	Multistage joint channel-data estimation	29
3-4	Multistage joint channel-data estimation in conventional receiver	30
3-5	Structure of training sequence codes for GSM	31
3-6	Maximum correlation and constant amplitude zero auto-correlation channel estimation	35
3-7	Correspondence of path metric in $\check{H}\mathbf{s}$ space to codeword distance to \mathbf{s}^r	38
4-1	Overall SAIC receiver structure	44
4-2	Pseudocode for the proposed SAIC algorithm	48
5-1	Drop in C/I_1 at 1% frame error rate	58
5-2	Drop in Class 1b bit error rates at 1% frame error rate	59
5-3	Drop in Class 2 bit error rates at 1% frame error rate	60
A-1	Channel coding of full rate traffic channel	61

Chapter 1

Introduction

GSM/GPRS (Global System for Mobile Communications/General Packet Radio Service) is a popular cellular digital communication standard used for the mobile phones. In a GSM/GPRS network, physical areas are divided into cells with radius ranging from several hundred meters to 30km. As illustrated in Figure 1-1, each cell has a fixed transceiver, called the *base transceiver station*, responsible for transmitting and receiving voice and data packets to and from all the *mobile stations* (e.g. cell phones) within the cell. The transmission paths from the base transceiver station to all the mobile stations are called the *downlinks*, while the reverse transmission paths from each mobile station to the base station are called the *uplink*.

The base transceiver station can transmit simultaneously with any mobile station signals without interfering each other because the uplink and downlink are frequency division duplexed over two different frequency bands. For example, the 880-915MHz and 925-960MHz bands are allocated for downlink and uplink respectively in the GSM900 standard. These two frequency bands are each subdivided into consecutive 200kHz bands called the *physical channels*, each of which is allocated for the transmission of a narrow-band signal at the symbol rate $\frac{1}{T} = 270.83$ kilo-symbols/s.

To avoid interference within each cell, the uplink and downlink physical channels are shared among the mobile stations according to a *time division* scheme, whereby all channels are divided into 0.577ms time slots. Each mobile station synchronizes to the base transceiver station in order to transmit and receive its data in the designated

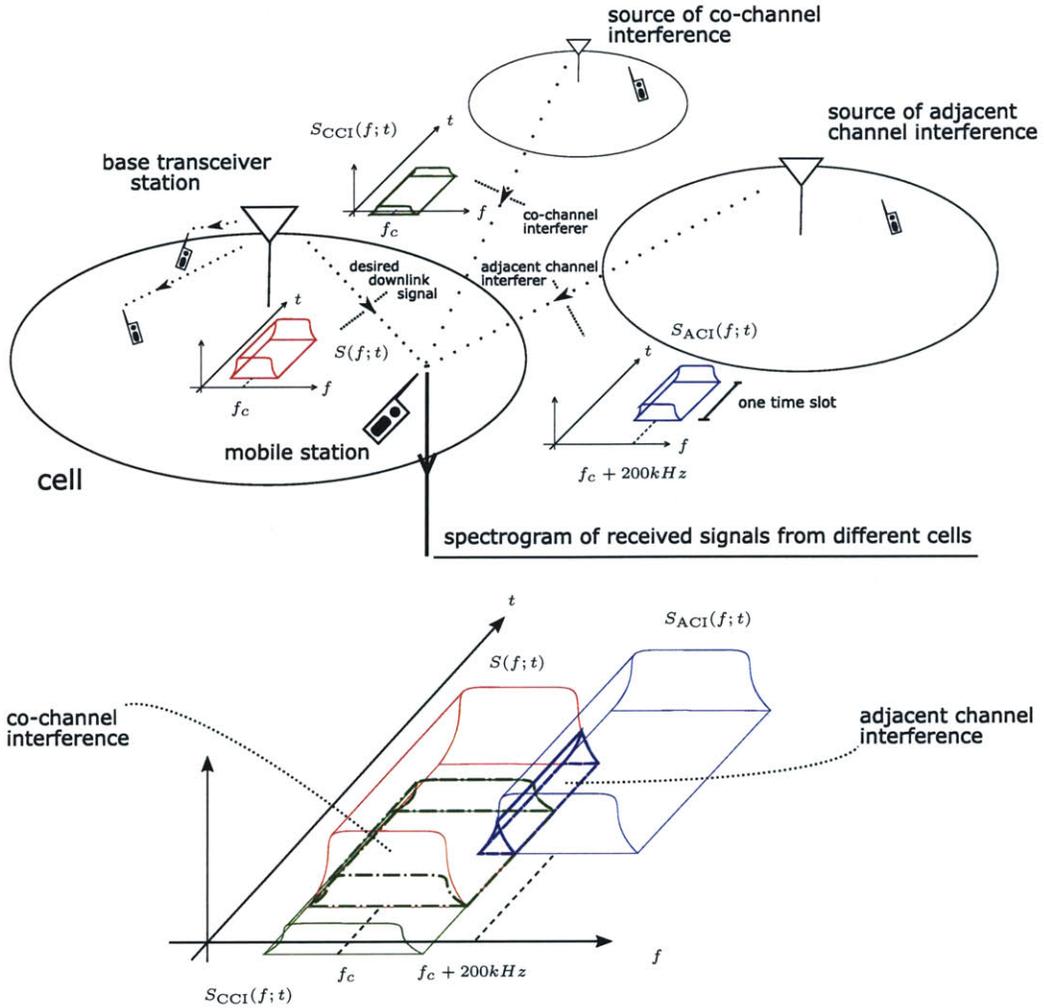


Figure 1-1: Adjacent and co-channel interference at mobile station

time slots over the downlink and uplink physical channels respectively.

In the asynchronous GSM/GPRS networks, the time division scheme is employed within each cell but not among different cells because the base transceiver stations are asynchronous. Thus, interference may occur if the same channel or overlapping adjacent channels are used simultaneously in near-by cells, causing co-channel interference and adjacent channel interference respectively. This is illustrated by the overlapping spectrograms of the downlink received signals in Figure 1-1.

To increase the downlink capacity, each base transceiver station has to transmit more frequently over a larger set of physical channels. As a consequence, there must be a decrease in the maximum distance (in the number cells) between two

base transceiver stations simultaneously transmitting over the same channel. The dominant co-channel interferer, thus, becomes stronger for every base station. To optimize the trade-off between the downlink capacity and interference level with just one receive antenna, the mobile station has to exploit the special property of the interferers in demodulation and decoding. Many such techniques have been developed and are commonly called the Single Antenna Interference Cancellation (SAIC). In this project, a decorrelator-based SAIC algorithm is implemented and tested in the computer simulation for the downlink receiver of the asynchronous GSM/GPRS networks under the set of test scenarios and performance requirements called the DARP (Downlink Advanced Receiver Performance). Since the algorithm reuses and extends on the key components in the conventional receiver without SAIC, the following two chapters (Chapter 2 and 3) will be devoted to the mathematical models involved in the GSM signal transmission and the conventional signal detection already developed at Qualcomm. In Chapter 4, the single co-channel interference model derived based on these models are used to analyze the SAIC algorithm. Finally, The DARP and the computer simulation results will be stated in 5.

Chapter 2

Background on GSM Transmission

This chapter and the following one will introduce the basics of GSM transmission and reception necessary in understanding the SAIC algorithm. The major part of the result presented here is based on the studies of the GSM simulation and system design document developed by many system engineers at Qualcomm. The other part describes the general GSM standards written by the European Telecommunication Standards Institutes. For conciseness, some implementational or mathematical details will be skipped. Interested readers may refer to the references or appendices for details.

2.1 GSM Normal Burst

In GSM, the base station communicates with different mobile stations in different time slots. The message intended for each mobile station is transmitted in one of the eight slots in every frame. Each frame, therefore, contains messages for at most eight mobile stations. In every slot, a structured data sequence is Gaussian Minimum Shift Keying (GMSK) modulated on the carrier. The structured data sequence is called a *burst*. Different *logical channels*, which are the communication paths defined by the content of the data rather than the physical transmission settings, may have different bursts structures and channel coding schema to adapt to their functions. However, we will only consider the *normal burst* structure[3] shown in Figure 2-1 as it is typically

used for the logical channels on which the SAIC algorithm is primarily designed for.

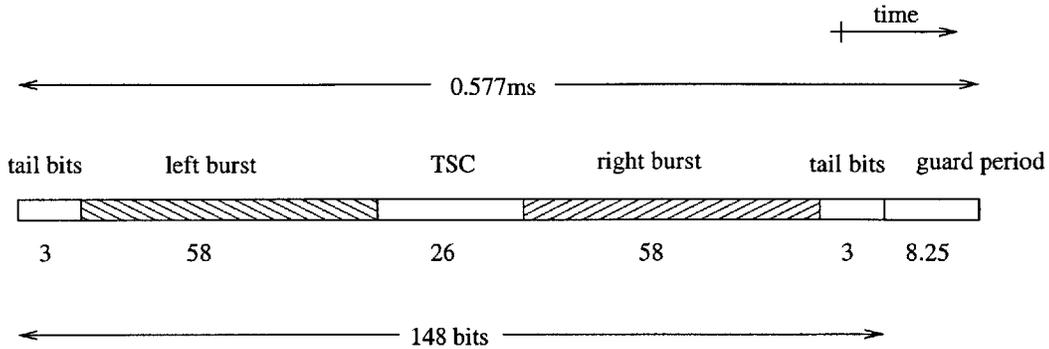


Figure 2-1: Normal burst structure

The normal burst consists of a known 26-bit *training sequence code* in the middle of the two unknown 58-bit data sequences, called the *left burst* and *right burst*. The two tails of the sequence consist of 3 bits of 0, called the *tail bits*. The guard period is not a data sequence but rather a time interval over which the transmitter must ramp down in power to avoid interfering the signal in the next time slot. It is also the interval for the transmitter to ramp up in power[4] to prepare transmission in the next time slot. If the base station need to transmit over adjacent time slots, then power ramp down is unnecessary.

The design of the burst structure allows a simple decoding scheme that will be outlined here and describe in greater detail later. After the receiver receive the signal in the normal burst structure, it can learn the channel from the observation corresponding to the training sequence code, and then decode the left and right burst assuming that the channel is approximately constant over one burst.

2.2 Channel Coding for Traffic Channels

While channel coding is not the main focus of the design of the proposed SAIC algorithm, it is important in understanding the performance metrics such as the *frame error rate*, *Class 1b residual bit error rate*, and *Class 2 residual bit error rate*. These metrics will be described through the *full-rate traffic* channel coding scheme[2], which is representative of other channel coding schema.

The full-rate traffic channel is a logical channel for encoded speech data. A fixed-length block encoder, called the vocoder, encodes a speech segment into three major classes of bits: Class 1a, 1b and 2 bits. These three classes differ by their importance in speech recovery: The Class 1a bits, which occupies about 20% of each source codeword, are the most important. They are the only bits that are protected by the cyclic redundancy code (CRC); The Class 1b bits are the next important bits. Together with the Class 1a bits, they are called the Class 1 bits, which occupies exactly 70% of the source codeword. Both the CRC-protected Class 1a bits and the Class 1b bits are convolutionally encoded; the least important class is the Class 2 bits, which occupies the remaining 30% of the source codeword. There is almost no channel coding over that class, except that they are interleaved over different bursts together with the Class 1 bits. In Appendix A.1, the channel code will be described in greater details for any interested readers.

Knowing how the encoded speech data is channel coded allows us to understand the performance metrics of the receiver. The frame error rate indicates how frequently an error occurs in the transmitted Class 1a bits. This is not an entirely correct statement because the frame error measured is indeed the error on the Class 1a bits *detected* by the three parity bits from the CRC. Thus, errors not detected by the CRC are not reflected in the frame error rate measurement. The purpose of detecting the frame error is such that the receiver can choose to throw away the entire frame with frame error because any errors on the Class 1a bits would render the speech unrecoverable. This is why the frame error is detected but not calculated by comparing the decoded Class 1a bits with the correct Class 1a bits, which are not known at the receiver. The residual bit errors, whether it is on Class 1b or Class 2 bits, indicates how frequency a bit error occurs in a frame that is detected to be correct by the CRC. This is because the frame detected with frame error is discarded in the receiver. It is, therefore, not very meaningful to measure the bit error rates over those discarded frames, which are never decoded. There is a separate bit error rate measurement on Class 1b and Class 2 bits because the two classes differ in their importance in speech recovery and thus have different levels of the maximum tolerable

error rate. Since they are not CRC-protected, the bit error cannot be detected at the receiver. Hence, bit error rate measurement is computed by comparing the decoded bits with the actual transmitted data, and it is not available to the receiver.

2.3 GMSK Modulation

Let $\{a_n\}_{n=n_i}^{n_f}$ be the data sequence assigned to one normal burst. Before it gets transmitted in one time slot, it is *differentially decoded* to another equal length sequence $\{d_n\}_{n=n_i}^{n_f}$ according to the following formula,

$$d_n = \begin{cases} a_n & \text{if } n = n_i \\ a_n \oplus a_{n-1} & \text{otherwise} \end{cases} \quad (2.1)$$

where \oplus denotes the mod-2 addition. It is interesting that this decoding step is performed even before the signal is modulated and transmitted. As will be clear in Section 3.1, it has the effect of undoing the differential coding in the Differential Binary Phase Shift Keying approximation to GMSK (see Section 2.4).

Each element of the differentially decoded sequence is mapped to ± 1 by the function $a \mapsto (-1)^a$. The resultant sequence is called the *Euclidean image* of the binary sequence $\{d_n\}_{n=n_i}^{n_f}$ and is denoted as $\{b_n\}_{n=n_i}^{n_f}$. Note that the original data sequence $\{a_n\}_{n=n_i}^{n_f}$ can be recovered from $\{b_n\}_{n=n_i}^{n_f}$. More precisely, let $\{s_n\}_{n=n_i}^{n_f}$ be the sequence such that

$$\begin{aligned} s_n &\triangleq \prod_{k=n_i}^n b_k = \prod_{k=n_i}^n (-1)^{d_k} = (-1)^{\overbrace{d_{n_i} \oplus \cdots \oplus d_n}^{\text{differential encoding}}} \\ &= (-1)^{a_n} \quad \text{by (2.1)} \end{aligned} \quad (2.2)$$

which is indeed the Euclidean image of the original data sequence $\{a_n\}_{n=n_i}^{n_f}$ and so $\{a_n\}_{n=n_i}^{n_f}$ can be recovered by the inverse map $s \mapsto \frac{1-s}{2}$.

The Euclidean image of the differentially encoded data sequence $\{d_n\}_{n=n_i}^{n_f}$ is GMSK modulated[8] on the carrier waveform as shown in Figure 2-2. Roughly speaking,

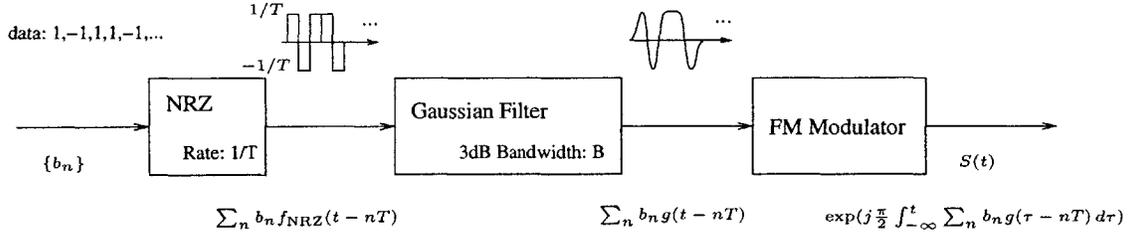


Figure 2-2: GMSK modulation

the sequence first amplitude modulates a train of rectangular pulses, which is then smoothed out by the Gaussian filter. Finally, the smoothed pulse train drives the frequency of a constant-envelope complex waveform to be transmitted over its specified frequency band. In the following, we will derive the mathematical expression of the continuous-time transmitted signal, from which a DT measurement model is obtained for the analysis of the SAIC algorithm.

Let $S(t)$ be the continuous-time transmitted signal, $f_G(t)$ be the unit-area Gaussian filter with 3 dB bandwidth B , and $f_{\text{NRZ}}(t) \triangleq \begin{cases} \frac{1}{T} & \frac{T}{2} \leq t < \frac{T}{2} \\ 0 & \text{otherwise} \end{cases}$ be the rectangular pulse. The expression for the Gaussian filter can be obtained by choosing the variance of the general Gaussian probability density function $\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{t^2}{2\sigma^2}}$ so that its Fourier transform $e^{-2\sigma^2(\pi f)^2}$ has a -3 dB power gain at $f = B$.

$$\text{Setting } e^{-2\sigma^2(\pi B)^2} = \frac{1}{\sqrt{2}} \implies \sigma = \frac{\sqrt{\ln 2}}{2\pi B}$$

$$\therefore f_G(t) \triangleq \sqrt{\frac{2\pi}{\ln 2}} B e^{-\frac{2}{\ln 2}(\pi B t)^2}$$

The system response of the first two components in the GMSK demodulator can be characterized by the convolution,

$$g(t) \triangleq (f_G * f_{\text{NRZ}})(t) = \frac{1}{2T} \left[\text{erfc} \left(\sqrt{\frac{2}{\ln 2}} \pi B \left(t - \frac{T}{2} \right) \right) - \text{erfc} \left(\sqrt{\frac{2}{\ln 2}} \pi B \left(t + \frac{T}{2} \right) \right) \right]$$

where $\text{erfc}(t) \triangleq \frac{2}{\sqrt{\pi}} \int_t^\infty e^{-\tau^2} d\tau$. It is called the frequency pulse shape because the input to the FM modulator in Figure 2-2 is a train of the time-shifted $g(t - nT)$

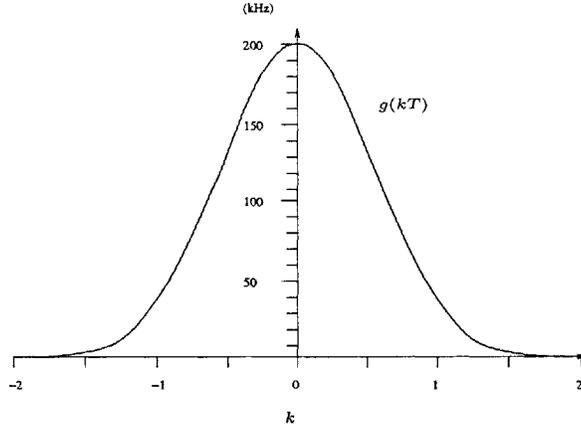


Figure 2-3: Frequency pulse of the GSM modulator

modulated by the symbol b_n , and this train of $\pm g(t)$ drives the frequency of the signal to be transmitted. The phase of the signal is the integral of the frequency pulse train, which yields a train of time-shifted phase variation $q(t - nT)$ modulated by b_n . This characteristic phase variation can be obtained by integrating frequency pulse shape as follows,

$$\begin{aligned}
 q(t) &= \int_{-\infty}^t g(\tau) d\tau \\
 &= 1 + \frac{\sqrt{\ln 2}}{2\pi BT\sqrt{2}} \left[G\left(\sqrt{\frac{2}{\ln 2}}\pi B\left(t - \frac{T}{2}\right)\right) - G\left(\sqrt{\frac{2}{\ln 2}}\pi B\left(t + \frac{T}{2}\right)\right) \right]
 \end{aligned}$$

where

$$\begin{aligned}
 G(x) &\triangleq \int \operatorname{erfc}(x) dx \\
 &= x \operatorname{erfc}(x) - \frac{1}{\sqrt{\pi}} e^{-x^2}
 \end{aligned}$$

For GSM, the bandwidth-time product BT is 0.3. Thus, $g(t)$ is approximately zero outside the interval $[-1.5T, 1.5T]$ as shown in Figure 2-3. We can, therefore, approximate the phase variation by integrating a function that is identical to $g(t)$ but with the tails outside the time interval $[-1.5T, 1.5T]$ trimmed off. More precisely, we define $\phi_0(t)$ to be the scaled and time-shifted phase variation based on the frequency

pulse shape that is tail-trimmed to $[-LT/2, LT/2]$ as follows,

$$\phi_0(t) \triangleq \begin{cases} 0 & t < 0 \\ \frac{\pi}{2} \frac{q(t - \frac{L}{2}T) - q(-\frac{L}{2}T)}{q(\frac{L}{2}T)} & 0 \leq t < LT \\ \frac{\pi}{2} & T \leq t \end{cases}$$

L is the length of the tail-trimmed frequency pulse shape [per unit T] and so it is also the length in T over which the phase varies. With $L = 3$ and $\phi_n(t) \triangleq \phi_0(t - nT)$, the overall GMSK modulated signal $S(t)$ can be approximated as,

$$S(t) \approx \exp \left(j \sum_{n=n_i}^{n_f} b_n \phi_n(t) \right) \quad (2.3)$$

2.4 Approximating GMSK with DBPSK

The purpose of approximating GMSK with Differential Binary Phase Shift Keying (DBPSK) modulation is twofold: first, it suggests the use of the common receiver front-end with matched filtering, which converts the continuous-time received signal to discrete-time received samples ; second, as the consequence of using the matched filtering front-end, a concrete discrete-time measurement model can be obtained for the mathematical analysis in the subsequent demodulation and decoding algorithms.

In the previous section, the GMSK modulation was shown to be approximately the phase modulation with a continuous phase variation $\phi_0(t)$ modulated by $\{+1, -1\}$ symbol sequence. Laurent[9] showed that any such continuous bi-phase modulation with a *finite-length* frequency pulse shape can be decomposed into a sum of amplitude modulated signals if we extend the transmission time to $\pm\infty$ by zero padding the data sequence $\{a_n\}_{n=n_i}^{n_f}$ at the two ends so that its Euclidean image becomes a bi-infinite sequence $\{b_n\}_{n=-\infty}^{\infty}$. In other words, the GMSK signal $S(t)$ can be expressed as $\sum_{K=0}^{M-1} \sum_{N=-\infty}^{\infty} A_{K,N} C_K(t - NT)$ where $\sum_{N=-\infty}^{\infty} A_{K,N} C_K(t - NT)$ is the

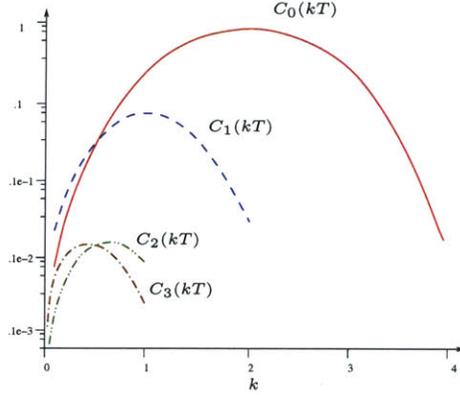


Figure 2-4: Logarithmic plot of the amplitude modulated pulses for GMSK

K -th amplitude modulated signal with pulse shape $C_K(t - NT)$ modulated by the transformed symbol $A_{K,N}$ that depends on the original symbol sequence. The number M of different decomposed pulses and the maximum pulse length grows exponentially and linearly respectively with the length L of the frequency pulse shape. For the interested readers, Appendix A.2 gives the proof and a more precise expression of the amplitude modulation decomposition.

Figure 2-4 shows the *logarithmic* plot of the $M = 4$ different pulses for the GMSK signal approximated by (2.3) with $L = 3$. Since $C_0(t)$ contains the most signal energy, we ignore the components of other pulses to approximate the GMSK signal as one amplitude modulated signal as follows,

$$\begin{aligned}
 S(t) &\approx \sum_{N=-\infty}^{\infty} A_{0,N} C_0(t - NT) & (2.4) \\
 &= \sum_{N=-\infty}^{\infty} C_0(t - NT) \prod_{n=-\infty}^N j b_n
 \end{aligned}$$

The last expression is indeed equivalent to the DBPSK with signal points $j b_n \in \{+j, -j\}$. This simplified continuous-time expression for the transmitted signal will enable us to obtain a simple discrete-time measurement model (3.3) of the received signal from a particular implementation of the GMSK demodulator. The measurement model will then become the base of the mathematical analysis on the SAIC algorithm.

Chapter 3

Background on GSM Signal Reception

3.1 Discrete-time Measurement Model without Interference

So far, the signal $S(t)$ is regarded as deterministic because the data sequence $\{a_n\}_{n=-\infty}^{\infty}$ is known to the transmitter. In the perspectives of the receiver, however, the data sequence is unknown and often modeled as a sequence of independent and identically distributed random variables each of which is equiprobable over the sample space $\{0, 1\}$. To distinguish the random sequence from the deterministic sequence, we will represent all random variables in a sans serif font. For example, $\{b_n\}_{n=-\infty}^{\infty}$ and $S(t)$ are the random symbol sequence and the transmitted signal respectively derived from the random data sequence $\{a_n\}_{n=-\infty}^{\infty}$ whose Euclidean image is $\{s_n\}_{n=-\infty}^{\infty}$. Figure 3-1 illustrates, in the perspectives of the receiver, a *probabilistic model* of how the random process $S(t)$ can be constructed using the DBPSK approximation to GMSK in (2.4) and a random number generator that generates $\{a_n\}_{n=-\infty}^{\infty}$.

The signal $S(t)$ gets partially corrupted in the wireless media before it reaches the receiver. Thus, the received signal $X(t)$ is not in general identical to $S(t)$. The typical corruptions other than the additive white Gaussian noise are: 1) the multipath

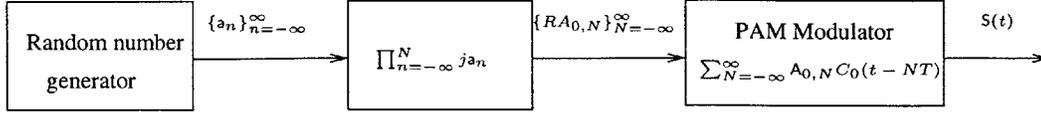


Figure 3-1: Probabilistic model of the GMSK signal

effect or the *frequency-selective fading*, which is due to the reception of the signal from different paths with possibly different attenuation and delay; 2) the Doppler spread or the *time-selective fading*, which is caused by the relative motion between the receiver and transmitter or any reflectors in the path; 3) the co-channel interference and adjacent channel interference, which are caused respectively by other GSM signal in the same and adjacent frequency bands. In this section, the channel model will incorporate the multipath fading and white noise. Single co-channel interference will be incorporated in the model later in Section 4.2. For simplicity of the subsequent mathematical analysis on the SAIC algorithm, some types of degradation are not incorporated in the measurement model. In particular, the Doppler spread is ignored by assuming that the transmission time of the normal burst is short enough that the relative motion between the transmitter and the receiver is negligible. The adjacent channel interference is also left out by assuming that the anti-aliasing filter in the receiver front-end effectively eliminates the out-of-band interference. However, in the computer simulation, both the Doppler spread and the adjacent channel interference will be taken into account according to Section C.3 in [4].

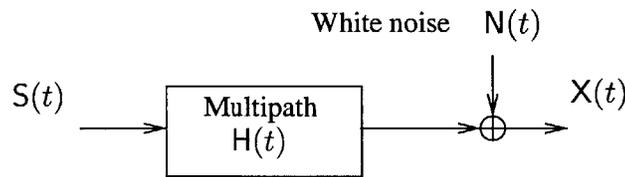


Figure 3-2: Probabilistic channel model without co-channel interference

Figure 3-2 illustrates the interference-free channel as a two-step process. To generate the multipath effect, the signal is first filtered by the *complex* linear time-invariant filter $H(t)$, called the channel impulse response. The *complex* white Gaussian random noise process N with one-sided power spectral density σ_n^2 is then added to the

multipath signal to yield the *complex* received random process $\mathbf{X}(t)$. That is,

$$\begin{aligned}\mathbf{X}(t) &= (\mathbf{H} * \mathbf{S})(t) + \mathbf{N}(t) \\ &= \sum_{m=-\infty}^{\infty} (\mathbf{H} * C_0)(t - mT) \prod_{n=-\infty}^m j\mathbf{b}_n + \mathbf{N}(t)\end{aligned}\quad (3.1)$$

The continuous-time channel model (3.1) can be turned into the discrete-time model by filtering $\mathbf{X}(t)$ with the perfect anti-aliasing filter $\frac{1}{T} \text{sinc}(\frac{\pi}{T}(t))$ followed by sampling at every T . More precisely, let $\mathbf{x}_k \triangleq j^{-k} \langle \mathbf{X}(t), \frac{1}{T} \text{sinc}(\frac{\pi}{T}(t - kT)) \rangle$ be the discrete-time received signals, $\mathbf{n}_k \triangleq \langle \mathbf{N}(t), \frac{1}{T} \text{sinc}(t - kT) \rangle$ be the white noise sequence. Assuming that $(\mathbf{H} * C_0)$ is *perfectly* band limited to $1/2T$,

$$\begin{aligned}\mathbf{x}_k &= j^{-k} \sum_{m'=-\infty}^{\infty} (\mathbf{H} * C_0)(iT - m'T) \left(\prod_{n=-\infty}^{m'} j \right) \left(\prod_{n=-\infty}^{m'} \mathbf{a}_n \right) + \mathbf{n}_k \\ &= \sum_{m=-\infty}^{\infty} (\mathbf{H} * C_0)(mT) \left(\prod_{n=-\infty}^{-m} j \right) \mathbf{s}_{k-m} + \mathbf{n}_k \quad \text{by (2.2)}\end{aligned}$$

Let $\mathbf{h}_m \triangleq (\mathbf{H} * C_0)(mT) \left(\prod_{n=-\infty}^{-m} j \right)$ be the discrete-time channel impulse response¹.

$$\mathbf{x}_k = \sum_{m=-\infty}^{\infty} \mathbf{h}_m \mathbf{s}_{k-m} + \mathbf{n}_k \quad (3.2)$$

which is the desired discrete-time measurement model. Note that $\{\mathbf{n}_k\}_{k=-\infty}^{\infty}$ is a sequence of complex white Gaussian random variable with variance σ_n^2 . $\{\mathbf{s}_k\}_{k=-\infty}^{\infty}$ is a sequence of *independent* and identically distributed random variables each of which is equiprobable over the sample space $\{-1, +1\}$. It is the only real-valued sequence in (3.2) while \mathbf{x}_k , \mathbf{h}_m and \mathbf{n}_k are all complex.

The discrete-time channel impulse response $\{\mathbf{h}_m\}_{m=-\infty}^{\infty}$ usually has most of its energy within a few consecutive samples because the delay spread (per unit T) is often concentrated over a short interval. It is, therefore, approximately a finite-length sequence. Without loss of generality, let $\mathbf{h} = [\mathbf{h}_0 \dots \mathbf{h}_v]$ be the $v + 1$ -tap channel

¹ $\left(\prod_{n=-\infty}^{-m} j \right)$ does not converge. The intended meaning is $\left(\prod_{n=n_i}^{-m} j \right)$ where n_i is small but finite. (See Remark A.2.2)

impulse response where v is the memory length. The discrete-time measurement model becomes,

$$\mathbf{x}_k = \sum_{m=0}^v h_m \mathbf{s}_{k-m} + \mathbf{n}_k \quad (3.3)$$

Since the signal detection is usually performed block-by-block in GSM, it will come in handy to re-organize the variables in (3.3) in vector/matrix form, and express the measurement model in matrix form by replacing the convolution with matrix multiplication. Let $p + 1$ be the observation time (in samples). By defining the following matrices as,

$$\begin{aligned} \mathbf{X} &\triangleq \begin{bmatrix} \Re(\mathbf{x}_k) & \dots & \Re(\mathbf{x}_{k+p}) \\ \Im(\mathbf{x}_k) & \dots & \Im(\mathbf{x}_{k+p}) \end{bmatrix}, & \mathbf{H} &\triangleq \begin{bmatrix} \Re(\mathbf{h}_k) & \dots & \Re(\mathbf{h}_{k-v}) \\ \Im(\mathbf{h}_k) & \dots & \Im(\mathbf{h}_{k-v}) \end{bmatrix} \\ \mathbf{S} &\triangleq \begin{bmatrix} \mathbf{s}_k & \dots & \mathbf{s}_{k+p} \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{k-v} & \dots & \mathbf{s}_{k+p-v} \end{bmatrix}, & \mathbf{N} &\triangleq \begin{bmatrix} \Re(\mathbf{n}_k) & \dots & \Re(\mathbf{n}_{k+p}) \\ \Im(\mathbf{n}_k) & \dots & \Im(\mathbf{n}_{k+p}) \end{bmatrix} \end{aligned} \quad (3.4)$$

the measurement model in matrix form becomes,

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{N} \quad (3.5)$$

The column vectors \mathbf{s} , \mathbf{x} and \mathbf{n} can conveniently represent the sequences of the symbol $\{\mathbf{s}_i\}_{i=k-v}^{k+p}$, received signal $\{\mathbf{x}_i\}_{i=k}^{k+p}$ and white noise $\{\mathbf{n}_i\}_{i=k}^{k+p}$ respectively without specifying the indices. Note that all vectors and matrices are represented in *bold-faced capital* lowercase and uppercase characters respectively. Furthermore, all the matrices defined are *real* matrices while the vectors may be complex. This is because the proposed SAIC algorithm will be expressed in terms of the real matrix arithmetic, which regards the real and imaginary observations as virtual spatial dimensions.

3.2 Multistage Joint Channel-Data Estimation

In the previous section, we described how the receiver turns the continuous-time observation $X(t)$ into discrete-time samples that can be represented by the vector \mathbf{x} .

Although the conversion is not information lossless in general, it allows the subsequent signal detection procedure to be performed in the digital rather than the analog domain that is usually more costly and complex. Similarly, to reduce complexity, the signal detection is divided into two stages— demodulation and decoding. Decoding refers to the process of recovering the source code from the channel code described in Section 2.2 and Appendix A.1. Since the proposed SAIC algorithm modifies on the demodulation but not the decoding step, the demodulation rather than the decoding technique will be the main focus of this project. In this section, we will derive the conventional demodulation technique (without SAIC) called the multistage joint channel-data estimation from the fundamental optimality criterion of minimizing the probability of error.

The goal of the demodulator is to choose the data sequence estimate $\hat{\mathbf{s}}$ that minimizes the probability of error (MPE) based on the received signal, or equivalently, maximizes the a posteriori probability (MAP).

$$\begin{aligned} \text{MAP/MPE: } \hat{\mathbf{s}} &= \arg \max_{\mathbf{s}} p(\mathbf{s}|\mathbf{x}) \\ &= \arg \max_{\mathbf{s}} \int_{\chi(\mathbf{h})} p(\mathbf{s}|\mathbf{x}, \mathbf{h}) d\mathbf{h} \end{aligned}$$

where $p(\mathbf{s}|\mathbf{x}, \mathbf{h})$ denotes the probability density function $p_{\mathbf{s}|\mathbf{x},\mathbf{h}}(\mathbf{s}|\mathbf{x}, \mathbf{h})$ and $\chi(\mathbf{h})$ is the support set of \mathbf{h} . $p(\mathbf{s}|\mathbf{x}, \mathbf{h})$ can be computed based on the measurement model (3.3), assuming some a priori probability for \mathbf{h} . However, to avoid evaluating the integral over $\chi(\mathbf{h})$, the optimality criterion is usually changed to the suboptimal joint MAP as follows,

$$\text{JMAP: } [\hat{\mathbf{h}} \ \hat{\mathbf{s}}] = \arg \max_{[\mathbf{h} \ \mathbf{s}]} p(\mathbf{s}, \mathbf{h}|\mathbf{x})$$

The joint optimization can be broken into two steps: 1) solve for the optimal channel as a function of the signal sequence; 2) substitute the solution from Step 1 into the

JMAP criterion and solve for the optimal signal sequence. That is,

$$\begin{aligned}\hat{\mathbf{s}} &= \arg \max_{\mathbf{s}} \left\{ p(\mathbf{s}, \hat{\mathbf{h}}(\mathbf{s})|\mathbf{x}) \mid \hat{\mathbf{h}}(\mathbf{s}) = \arg \max_{\mathbf{h}(\mathbf{s})} p(\mathbf{s}, \mathbf{h}(\mathbf{s})|\mathbf{x}) \right\} \\ &= \arg \max_{\mathbf{s}} \left\{ p(\mathbf{s}, \hat{\mathbf{h}}(\mathbf{s})|\mathbf{x}) \mid \hat{\mathbf{h}}(\mathbf{s}) = \arg \max_{\mathbf{h}(\mathbf{s})} p(\mathbf{h}(\mathbf{s})|\mathbf{x}, \mathbf{s}) \right\}\end{aligned}$$

In the last equality, $p(\mathbf{s}, \mathbf{h}|\mathbf{x})$ is replaced by $p(\mathbf{h}|\mathbf{x}, \mathbf{s})$ because $p(\mathbf{s}, \mathbf{h}|\mathbf{x})$ can be written as $p(\mathbf{h}|\mathbf{x}, \mathbf{s})p(\mathbf{s}|\mathbf{x})$ where $p(\mathbf{s}|\mathbf{x})$ is a non-negative quantity that does not depend on \mathbf{h} , the variable over which we optimize. Hence, $p(\mathbf{s}|\mathbf{x})$ can be eliminated from the optimization criterion.

To simplify the joint maximization further, we may adopt another suboptimal technique called the iterative approximation. i.e. The optimal channel estimate $\hat{\mathbf{h}}_1$ is calculated given an initial data sequence estimate $\hat{\mathbf{s}}_0$. Given this channel estimate $\hat{\mathbf{h}}_1$, the optimal data sequence estimate $\hat{\mathbf{s}}_1$ is calculated. And given this data sequence estimate, we can refine the channel estimate $\hat{\mathbf{h}}_2$, and so forth. i.e.

$$\begin{aligned}\text{Multistage: } \hat{\mathbf{h}}_i &= \arg \max_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}, \hat{\mathbf{s}}_{i-1}) \\ \hat{\mathbf{s}}_i &= \arg \max_{\mathbf{s}} p(\mathbf{s}|\mathbf{x}, \hat{\mathbf{h}}_i)\end{aligned}$$

The last equality again uses the fact that $p(\mathbf{s}, \hat{\mathbf{h}}_i|\mathbf{x}) = p(\hat{\mathbf{h}}_i|\mathbf{x})p(\mathbf{s}|\mathbf{x}, \hat{\mathbf{h}}_i)$ and that $p(\hat{\mathbf{h}}_i|\mathbf{x})$ does not depend on \mathbf{s} . In GSM, the training sequence can be used as the initial data sequence estimate to start off the iteration. Once the channel is estimated, the entire data sequence over which the channel is relatively constant can be estimated in the second iteration. The process can terminate when the solution or optimality criterion converges. Figure 3-3 illustrates this multistage joint channel-data estimation procedure.

If the iterative approximation is unstable, the error at every estimation stage will accumulate and propagates to the subsequent stages. Hence, in practice, the training sequence code is used to improve the initial estimates so that the iteration needs to be

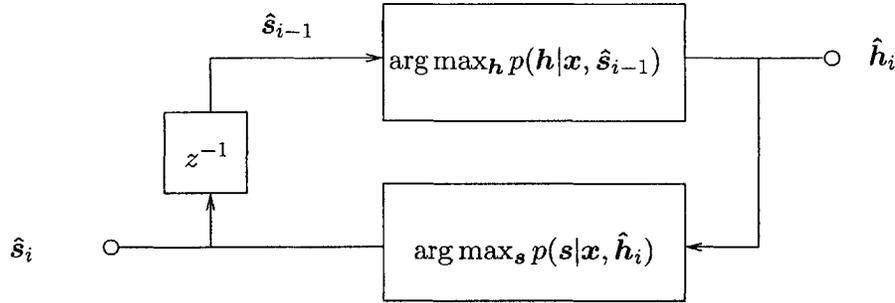


Figure 3-3: Multistage joint channel-data estimation

run only once or twice. Furthermore, since the data sequence is uniformly distributed, the MAP estimate is equivalent to the maximum likelihood (ML) estimate because

$$\begin{aligned}
 \arg \max_{\mathbf{s}} p(\mathbf{s}|\mathbf{x}, \hat{\mathbf{h}}_i) &= \arg \max_{\mathbf{s}} p(\mathbf{x}|\mathbf{s}, \hat{\mathbf{h}}_i)p(\mathbf{s}|\hat{\mathbf{h}}_i) \\
 &= \arg \max_{\mathbf{s}} p(\mathbf{x}|\mathbf{s}, \hat{\mathbf{h}}_i)p(\mathbf{s}) \quad \because \mathbf{s} \text{ is independent of } \mathbf{h} \\
 &= \arg \max_{\mathbf{s}} p(\mathbf{x}|\mathbf{s}, \hat{\mathbf{h}}_i) \quad \because p(\mathbf{s}) \text{ is constant}
 \end{aligned}$$

The ML estimate for the data sequence can be efficiently computed by the soft output maximum likelihood sequence estimation (SO-MLSE) in Section 3.2.2. To estimate the channel, we also change the MAP criterion to the ML criterion because the ML probability can be readily computed without knowing the a priori distribution of \mathbf{h} . In this additive white Gaussian noise model, the ML criterion can be simplified to the minimum distance or the least squares (LS) estimation rule. The training sequence codes of the GSM system possess the constant amplitude zero auto-correlation property, which leads to the simpler but suboptimal CAZAC channel estimation in Section 3.2.1. Figure 3-4 illustrates this joint channel-data estimation implemented in practice in the conventional receiver.

3.2.1 First-Stage LS Channel Estimation with Known Data

The receiver estimates the discrete-time channel impulse response in (3.3) from the observation that corresponds to the training sequence code in a normal burst. The

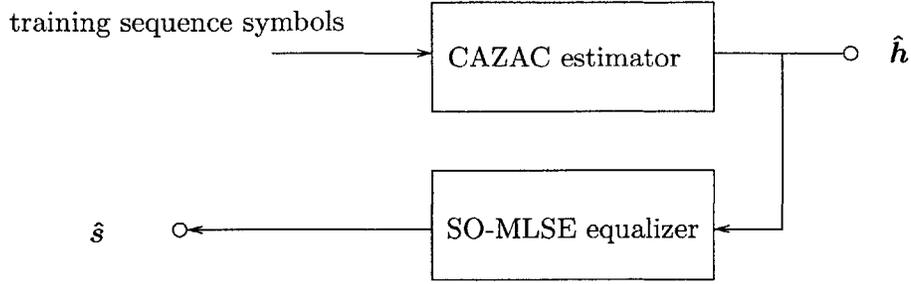


Figure 3-4: Multistage joint channel-data estimation in conventional receiver

measurement model becomes,

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{N} \quad (3.6)$$

$$\begin{bmatrix} \Re(x_0) \cdots \Re(x_{25-v}) \\ \Im(x_0) \cdots \Im(x_{25-v}) \end{bmatrix} = \begin{bmatrix} \Re(h_0) \cdots \Re(h_v) \\ \Im(h_0) \cdots \Im(h_v) \end{bmatrix} \begin{bmatrix} s_0 & \cdots & s_{25-v} \\ \vdots & \ddots & \vdots \\ s_{-v} & \cdots & s_{25-2v} \end{bmatrix} + \begin{bmatrix} \Re(n_0) \cdots \Re(n_{25-v}) \\ \Im(n_0) \cdots \Im(n_{25-v}) \end{bmatrix}$$

which is obtained by substituting $k = 0$ and $p = 25 - v$ in (3.4) and (3.5) without loss of generality. Note that \mathbf{S} (not in san serif font) is a deterministic matrix because the underlying data sequence $\{a_i\}_{i=-v}^{25-v}$ is the known training sequence code.

Using the Gaussian probability density function, the ML estimate is shown as follows to be equivalent to the LS estimate, which minimizes the Euclidean distance between the observation \mathbf{X} and the a priori estimate $\mathbf{H}\mathbf{S}$,

Proof. ML estimate for \mathbf{H} is equivalent to the LS estimate

$$\begin{aligned} \hat{\mathbf{H}}^{\text{ML}} &= \arg \max_{\mathbf{H}} \frac{1}{(\pi\sigma_n^2)^{p+1}} \exp\left(-\frac{\|\mathbf{X} - \mathbf{H}\mathbf{S}\|_{\text{F}}^2}{\sigma_n^2}\right) \\ &= \arg \min_{\mathbf{H}} \|\mathbf{X} - \mathbf{H}\mathbf{S}\|_{\text{F}}^2 \end{aligned} \quad (3.7)$$

where $\|\mathbf{Y}\|_{\text{F}} \triangleq \sqrt{\text{trace}(\mathbf{Y}\mathbf{Y}^{\prime})}$ and \mathbf{Y}^{\prime} denotes the Frobenius norm and the Hermitian respectively. \square

Note that the ML estimate exists but may not be unique. If we let \mathbf{S}^{\dagger} be the *generalized pseudoinverse* of \mathbf{S} , it can be shown that $\mathbf{X}\mathbf{S}^{\dagger}$ is a particular solution with minimum energy (i.e. minimum $\|\mathbf{H}\|_{\text{F}}^2$) and it exists even when \mathbf{S} does not have

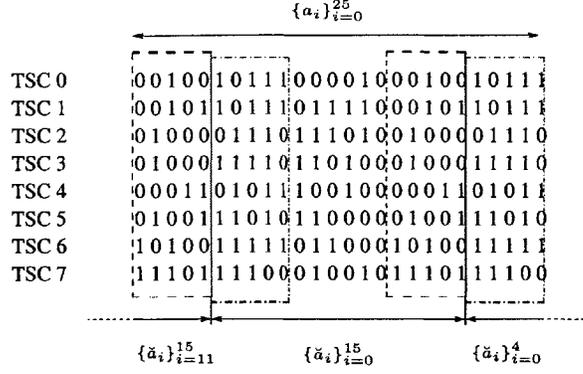


Figure 3-5: Structure of training sequence codes for GSM

full row or column rank. (See Appendix A.3) We will, therefore, always impose the minimum energy constraint so that the ML and LS channel estimates can be uniquely computed as,

$$\hat{\mathbf{H}}^{\text{ML}} = \hat{\mathbf{H}}^{\text{LS}} = \mathbf{X}\mathbf{S}^\dagger \quad (3.8)$$

The pseudoinverse \mathbf{S}^\dagger can be precomputed, stored in memory so that the computational complexity depends only on the matrix multiplication $\mathbf{X}\mathbf{S}^\dagger$ but not the pseudoinverse.

In the conventional receiver, a suboptimal channel estimation technique is adopted to further simplify the matrix multiplication into the convolution (3.17) by exploiting the special structure of the training sequences shown in Figure 3-5. The eight different 26-bit training sequence codes in GSM are derived from a 16-bit sequence $\{\check{a}_k\}_{k=0}^{15}$, called the *midamble code*, whose Euclidean image has the following constant amplitude zero auto-correlation property,

$$\begin{aligned} [(-1)^{\check{a}_i} \circledast (-1)^{\check{a}_{15-i}}]_{i=m} &\triangleq \sum_{i=0}^{15} (-1)^{\check{a}_i} (-1)^{\check{a}_{(i+m) \bmod 16}} \\ &= \begin{cases} 16 & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3.9)$$

$$(3.10)$$

where \circledast denotes the 16-point circular convolution.

As shown in Figure 3-5, the training sequence is a 26-point window of the periodic sequence generated from replicating the midamble code $\{\check{a}_i\}_{i=0}^{15}$. i.e.

$$a_{i-v} = \check{a}_{(i-5) \bmod 16} \quad 0 \leq i \leq 25 \quad (3.11)$$

Combining (3.10) and (3.11), we obtain a useful correlation function,

$$\begin{aligned} \rho(m) &\triangleq \frac{1}{16} \sum_{i=0}^{15} (-1)^{a_{5-v+m+i}} (-1)^{\check{a}_i} \\ &= \begin{cases} 1 & \text{if } m = 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3.12)$$

Suppose $v \leq 5$, let

$$\begin{aligned} \check{\mathbf{s}}_j &\triangleq [\mathbf{0}^{1 \times (5-v+j)} \ \check{\mathbf{s}}^T \ \mathbf{0}^{1 \times (5-j)}]^T \\ \check{\mathbf{S}} &\triangleq [\check{\mathbf{s}}_0 \ \dots \ \check{\mathbf{s}}_v] \frac{1}{16} \end{aligned} \quad (3.13)$$

where $\check{\mathbf{s}}$ is the column vector form of the Euclidean image of the midamble code $\{\check{a}_i\}_{i=0}^{15}$. Then, $\check{\mathbf{S}}$ decouples the channel taps in (3.6) because,

$$\begin{aligned} \mathbf{S}\check{\mathbf{S}} &= \begin{bmatrix} (-1)^{a_0} & \dots & (-1)^{a_{25-v}} \\ \vdots & \ddots & \vdots \\ (-1)^{a_{-v}} & \dots & (-1)^{a_{25-2v}} \end{bmatrix} [\check{\mathbf{s}}_0 \ \dots \ \check{\mathbf{s}}_v] \frac{1}{16} \\ &= \begin{bmatrix} \rho(0) & \dots & \rho(v) \\ \vdots & \ddots & \vdots \\ \rho(-v) & \dots & \rho(0) \end{bmatrix} = \mathbf{I} \quad \text{by (3.12)} \end{aligned} \quad (3.14)$$

$$\therefore \mathbf{X}\check{\mathbf{S}} = \mathbf{H} + \mathbf{N}\check{\mathbf{S}} \quad (3.15)$$

Since $\check{\mathbf{S}}$ is a non-symmetric Toeplitz matrix with each column containing the Euclidean image of the same midamble code, $16\check{\mathbf{S}}'\check{\mathbf{S}}$ is approximately an identity matrix by the constant amplitude zero auto-correlation property in (3.10). In other words, $\mathbf{N}\check{\mathbf{S}}$ is approximately white, and this approximation becomes more accurate for longer midamble codes. If we indeed approximate $\mathbf{N}\check{\mathbf{S}}$ in (3.15) as white Gaussian noise, the ML/LS channel estimate is identical to the mean $\mathbb{E}[\mathbf{X}\check{\mathbf{S}} \mid \mathbf{H}] = \mathbf{X}\check{\mathbf{S}}$, which simplifies

to the following convolution,

$$\hat{\mathbf{H}}^{\text{CAZAC}} \triangleq \mathbf{X}\check{\mathbf{S}} \quad (3.16)$$

$$\begin{aligned} &= \mathbf{X}' [\check{s}_0 \dots \check{s}_v] \frac{1}{16} \\ \hat{h}^{\text{CAZAC},m} &= \frac{1}{16} (x_i * \check{s}_{-i})|_{i=5-v+m} \quad \forall m \in \{0, \dots, v\} \\ &= \sum_{i=0}^{15} x_{i+(5-v+m)} \check{s}_i \end{aligned} \quad (3.17)$$

The last expression is the cross-correlation between the observation sequence $\{rx_i\}_{i=v}^{25}$ and the Euclidean image of the midamble code $\{\check{a}_i\}_{i=0}^{15}$ scaled by $\frac{1}{16}$. $\hat{\mathbf{H}}^{\text{CAZAC}}$ is called the CAZAC channel estimate.

Note that the difference between the CAZAC estimate and the ML estimate is purely the zero mean jointly Gaussian noise, i.e.

$$\begin{aligned} \hat{\mathbf{H}}^{\text{CAZAC}} - \hat{\mathbf{H}}^{\text{LS}} &= \mathbf{X}\check{\mathbf{S}} - \mathbf{X}\mathbf{S}^\dagger \\ &= \mathbf{N}(\check{\mathbf{S}} - \mathbf{S}^\dagger) \end{aligned}$$

In other words, how closely $\hat{\mathbf{h}}^{\text{CAZAC}}$ matches $\hat{\mathbf{h}}^{\text{LS}}$ on average is described by the covariance matrix of $\mathbf{N}(\check{\mathbf{S}} - \mathbf{S}^\dagger)$,

$$\text{Var}[\hat{\mathbf{h}}^{\text{CAZAC}} - \hat{\mathbf{h}}^{\text{LS}}] = \sigma_n^2 (\check{\mathbf{S}} - \mathbf{S}^\dagger)' (\check{\mathbf{S}} - \mathbf{S}^\dagger)$$

which is proportional to the noise variance σ_n^2 and the difference between \mathbf{S}^\dagger and $\check{\mathbf{S}}$, as expected. We can also show the degradation of $\hat{\mathbf{H}}^{\text{CAZAC}}$ in the mean squared distance $\text{E}[\|\mathbf{X} - \hat{\mathbf{H}}\mathbf{S}\|_{\mathbb{F}}^2]$, which is the expectation of the optimality criterion in (3.7) of the LS estimate. For interested readers, this analysis is carried out for the training sequence code TSC0 in Appendix A.4.

Although the CAZAC estimate is strictly suboptimal to the LS estimate and is limited to $v \leq 5$, it is simpler to implement as we only need to store the midamble code and perform convolution rather than to store the pseudoinverse matrix and perform matrix multiplication in the LS estimation case. More importantly, however, the

notion of cross-correlation allows channel estimation without precise time tracking. That is, even if we do not have the precise location of the desired observation sequence $\{x_i\}_{i=5-v}^{20}$ needed to compute the CAZAC estimate, we may compute the correlation in (3.17) with a long enough observation sequence, say $\{x_i\}_{i=k_1}^{k_2}$, so that the unknown interval $[k_1, k_2]$ is likely to contain the desired interval $[5 - v, 20]$. The resulting sequence, say $\{\hat{h}_m\}_{m=0}^{k_2-k_1-14}$, will contain the desired CAZAC channel estimate $\hat{\mathbf{h}}^{\text{CAZAC}}$ if $\{x_i\}_{i=k_1}^{k_2}$ indeed contains $\{x_i\}_{i=5-v}^{20}$. We can take the $v + 1$ -tap subsequence $\hat{\mathbf{h}}^{\text{MC}}$ in $\hat{\mathbf{h}}$ that has the *maximum sum of energy* to be the maximum correlation channel estimate $\hat{\mathbf{h}}^{\text{MC}}$. i.e.

$$\begin{aligned}
\hat{\mathbf{h}}^{\text{MC}} &\triangleq [\hat{h}^{\text{MC},0} \dots \hat{h}^{\text{MC},v}]^T \quad \text{where} \\
\hat{h}_{m'} &= (x_i * \check{s}_{-i})|_{i=k_1+m'} \quad \forall m' \in [0, k_2 - k_1 - 14] \\
j^* &= \arg \max_j \sum_{i=j}^{j+v} |\hat{h}_i|^2 \quad \forall j \in [0, k_2 - k_1 - 14 - v] \\
\hat{h}^{\text{MC},m} &= \hat{h}_{m+j^*} \quad \forall m \in [0, v]
\end{aligned} \tag{3.18}$$

$\hat{\mathbf{h}}^{\text{MC}}$ can be equal to $\hat{\mathbf{h}}^{\text{CAZAC}}$ exactly if the maximum energy interval is the desired interval $[5 - v, 20]$, which happens if the observations outside $[5 - v, 20]$ does not correlate well with the midamble and the noise does not degrade the correlation property too much.

Figure 3-6 illustrates the idea of channel estimation without precise time tracking. The system diagram on the left describes the channel model and the computation of the maximum correlation channel estimate $\hat{\mathbf{h}}^{\text{MC}}$, while the diagram on the right is the corresponding element-by-element description with $v = 4$ used in the actual implementation and $[k_1, k_2] = [-5, 24]$ that is unknown to the receiver.

²If $v < 5$, the sequences $\{x_i\}_{i=0}^{4-v}$ and $\{x_i\}_{i=21}^{25-v}$ are indeed wasted (not used) in the CAZAC estimate even though they depends on the training sequence code but not any unknown data.

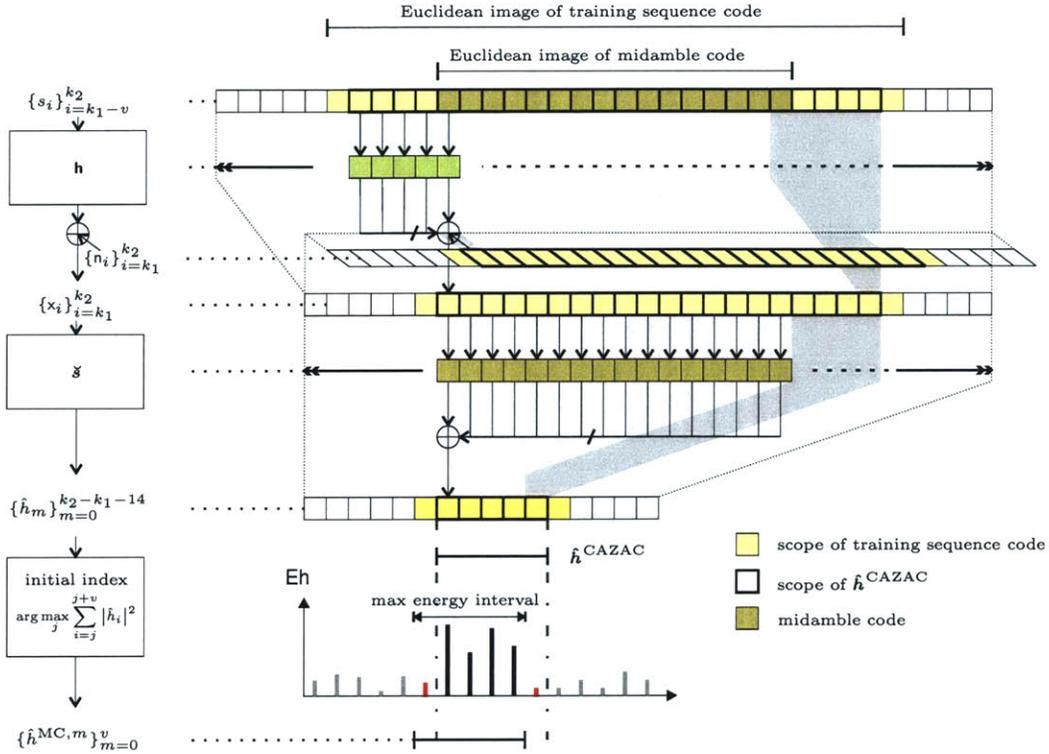


Figure 3-6: Maximum correlation and constant amplitude zero auto-correlation channel estimation

3.2.2 Second-Stage SO-MLSE of Data Burst with Known Channel

The channel estimate $\hat{\mathbf{h}}$ obtained in the first stage is used to estimate the data \mathbf{s} according to the following model,

$$\mathbf{x}_k = \sum_{m=0}^v \hat{\mathbf{h}}_m \mathbf{s}_{k-m} + \mathbf{n}_k \quad (3.19)$$

which is obtained by setting $\mathbf{h} = \hat{\mathbf{h}}$ in (3.3). The unknown data resides in the left and right bursts of the normal burst shown in Figure 2-1. Since the two semi-bursts are separated by the training sequence code, we can estimate them independently. Furthermore, the bi-lateral symmetry of the normal burst suggests that the same estimation technique on the right burst can be applied to the left burst in the time-

reversed fashion³. Thus, we consider \mathbf{s}_k in (3.19) on the time interval $[k_i - v, k_f]$ that covers only the last v training sequence bits, the entire right burst and the following $\min(v, 3)$ tail bits. Some known symbols are included in \mathbf{s} because the first and final observation that depends on the unknown data in the right burst also depends on those known symbols due to the channel memory.

With the specified time interval $[k_i, k_f]$, (3.19) can be expressed in matrix form (different from (3.5)),

$$\begin{bmatrix} x_{k_i} \\ \vdots \\ x_{k_f} \end{bmatrix} = \overset{\check{H}^{(k_f - k_i + 1) \times (k_f - k_i + v + 1)}}{\begin{bmatrix} \hat{h}_0 & \dots & \hat{h}_v & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \hat{h}_0 & \dots & \hat{h}_v \end{bmatrix}} \begin{bmatrix} s_{k_i - v} \\ \vdots \\ s_{k_f} \end{bmatrix} + \begin{bmatrix} n_{k_i} \\ \vdots \\ n_{k_f} \end{bmatrix} \quad (3.20)$$

Let $\mathcal{S} = \{\pm 1\}^{k_f - k_i + v + 1}$ be the set of symbol sequences, and $\mathcal{S}_C \subset \mathcal{S}$ be Euclidean image of the set of channel codewords⁴. The ML symbol sequence estimate $\hat{\mathbf{s}}^{\text{ML}, \mathcal{S}_C}$ is obtained by maximizing the Gaussian distribution function $p(\mathbf{s} | \mathbf{x}, \hat{\mathbf{h}})$ as follows,

$$\begin{aligned} \hat{\mathbf{s}}^{\text{ML}, \mathcal{S}_C} &= \arg \max_{\mathbf{s} \in \mathcal{S}_C} p(\mathbf{x} | \mathbf{s}, \hat{\mathbf{h}}) \\ &= \arg \max_{\mathbf{s} \in \mathcal{S}_C} \prod_{k=k_i}^{k_f} \frac{1}{\pi \sigma_n^2} \exp \left(- \frac{|x_k - \hat{\mathbf{h}}^T \mathbf{s}_k|^2}{\sigma_n^2} \right) \\ &= \arg \min_{\mathbf{s} \in \mathcal{S}_C} \underbrace{\sum_{k=k_i}^{k_f} \overbrace{|x_k - \hat{\mathbf{h}}^T \mathbf{s}_k|^2}^{\text{branch metric } \gamma_k(\mathbf{s}_k)}}_{\text{path metric } \Gamma(\mathbf{s})} \end{aligned} \quad (3.21)$$

Unlike the ML channel estimate in (3.8), there is no closed form expression for the ML symbol sequence estimate because of the constraint $\mathbf{s} \in \mathcal{S}_C$. An exhaustive search through the entire codeword is also impractical because of the exponential growth of time and memory requirement with respect to the length of the codeword.

³Although channel coding introduces correlation between the left and the right bursts, the correlation can be handled separately by the use of soft decisions.

⁴As described in Section 2.2, the channel code indeed covers multiple bursts rather than just the right burst. Here, we suppose that the channel code is only over the right burst to simplify analysis without loss of generality.

However, if we relax the constraint of $\mathbf{s} \in \mathcal{S}_C$ to $\mathbf{s} \in \mathcal{S}$, the minimization in (3.21) can be broken down into multiple optimization stages using dynamic programming. More precisely, let $\mathbf{s}_k = [s_k \dots s_{k-v}]^T$ be the signal state at time k . The minimum cost in (3.21) can be computed by optimizing recursively the path metric generated from the branch metric, define as follows,

$$\begin{aligned} \text{branch metric:} \quad \gamma_k(\mathbf{s}_k) &\triangleq \begin{cases} 0 & \text{if } p(\mathbf{s}_k) = 1 \\ \infty & \text{if } p(\mathbf{s}_k) = 0 \\ |x_k - \hat{\mathbf{h}}^T \mathbf{s}_k|^2 & \text{otherwise} \end{cases} \\ \text{optimal path metric:} \quad \Gamma_k(\mathbf{s}_k) &\triangleq \begin{cases} \gamma_k(\mathbf{s}_k) & \text{if } k = k_i + v \\ \min_{\mathbf{s}_{k-v-1}} \Gamma_{k-1}(\mathbf{s}_{k-1}) + \gamma_k(\mathbf{s}_k) & \text{otherwise} \end{cases} \\ \text{policy:} \quad \pi_{k-1}(\mathbf{s}_k) &= \arg \min_{\mathbf{s}_{k-v-1}} \Gamma_{k-1}(\mathbf{s}_{k-1}) + \gamma_k(\mathbf{s}_k) \end{aligned}$$

The ML estimate $\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} \in \mathcal{S}$ can then be obtained by tracing back the policies for the sequence of states that minimizes the overall path metrics $\Gamma_{k_f}(\mathbf{s}_k)$. This method of calculating the maximum likelihood symbol sequence estimate is called the MLSE equalization[5], and the recursive computation is called the Viterbi algorithm.

Since we relaxed the constraint of $\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}$ being a valid channel code, the channel decoder need to make changes on some of the symbol decisions with the minimal increase in the path metrics to obtain the final estimate $\hat{\mathbf{s}}^{\text{ML},\mathcal{S}_C}$ that is a valid codeword. The conventional channel decoder, however, has been developed assuming additive white Gaussian noise model without the intersymbol interference described by $\hat{\mathbf{h}}$. As a result, rather than minimizing the path metrics, which is the distance in the space of $\check{H}\mathbf{s}$ to \mathbf{x} , the decoder is given an observed codeword sequence $\hat{\mathbf{s}}^f$ and chooses a valid codeword $\mathbf{s} \in \mathcal{S}_C$ that minimizes the distance between \mathbf{s} to that observed codeword $\hat{\mathbf{s}}^f$. In order to use the conventional channel decoder optimally on channel with intersymbol interference, we need to choose the observed codeword sequence $\hat{\mathbf{s}}^f$ to be a sequence, denotes as $\mathbf{s}^r \in \mathbb{R}^{k_f - k_i + v + 1}$, which has the property that the codeword $\mathbf{s} \in \mathcal{S}_C$ that closest to \mathbf{s}^r minimizes the path metrics. This idea is illustrated in Fig-

ure 3-7. The higher dimensional spaces are represented by two-dimensional circles, while the subset relationship is shown by having circles within circles. The points, which represent a codeword, an a priori estimate or an observation sequence, are positioned such that the length of their outgoing arrows towards \mathbf{x} or a codeword corresponds the square root of the path metric or codeword distance respectively. Note from the figure that not only does \mathbf{s}^r exist, there are infinitely many possible sequence for \mathbf{s}^r (lying on the cyan dotted line) including the optimal channel codeword estimate itself $\hat{\mathbf{s}}^{\text{ML},S_c}$.

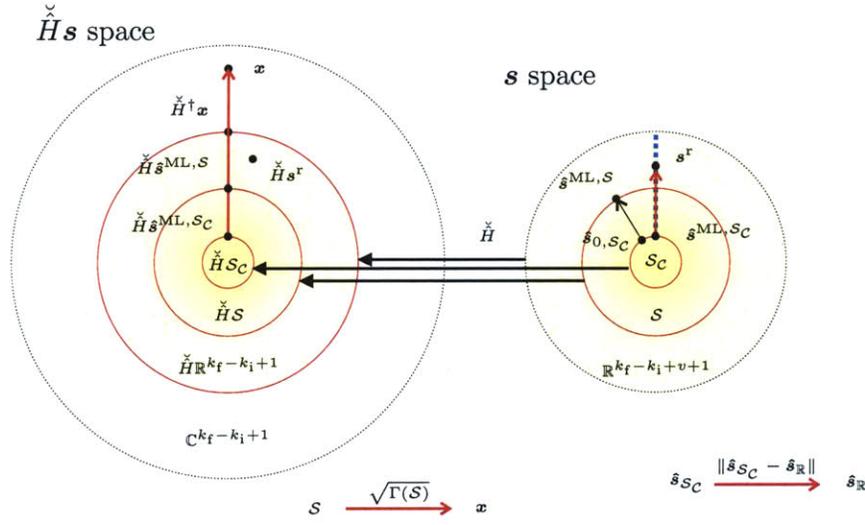


Figure 3-7: Correspondence of path metric in $\check{H}\mathbf{s}$ space to codeword distance to \mathbf{s}^r

An approximate value for \mathbf{s}^r can be derived by assuming that the optimal codeword $\hat{\mathbf{s}}^{\text{ML},S_c}$ is close to $\hat{\mathbf{s}}^{\text{ML},S}$. More precisely, since the path metric is minimum at $\hat{\mathbf{s}}^{\text{ML},S}$ in the set \mathcal{S} , its gradient at $\hat{\mathbf{s}}^{\text{ML},S}$ is small. In other words, a few symbol changes in $\hat{\mathbf{s}}^{\text{ML},S}$ does not increase the path metric significantly. The path metric is therefore approximately linear about $\hat{\mathbf{s}}^{\text{ML},S}$. If $\hat{\mathbf{s}}^{\text{ML},S_c}$ is close to $\hat{\mathbf{s}}^{\text{ML},S}$, the path metric of $\hat{\mathbf{s}}^{\text{ML},S_c}$ can be approximated by the linear interpolation over the path metrics of symbol sequences in \mathcal{S} closest to $\hat{\mathbf{s}}^{\text{ML},S}$ (i.e. the symbol sequences that differ by exactly one symbol from $\hat{\mathbf{s}}^{\text{ML},S}$). To formalize this idea mathematically, let δ_i be the location indicator $[\mathbf{0}^{1 \times i-k_i+v} \ 1 \ \mathbf{0}^{1 \times k_f-i}]^T$ which is one only at the position of the element s_i in \mathbf{s} . Furthermore, let ϵ be the codeword difference $\mathbf{s} - \hat{\mathbf{s}}^{\text{ML},S}$, which can be expressed in

terms of the orthonormal vectors $\boldsymbol{\delta}_i$ as follows,

$$\boldsymbol{\epsilon} \triangleq \sum_{i=k_i}^{k_f} \overbrace{(s_i - \hat{s}^{\text{ML},\mathcal{S}})}^{\epsilon_i} \boldsymbol{\delta}_i \quad (3.22)$$

Note that $s_i = \hat{s}^{\text{ML},\mathcal{S}}$ for $i \in [k_i - v, k_i - 1]$ they are part of the training sequence code. If we also constraint \mathbf{s} to be in \mathcal{S} , each non-zero element of $\boldsymbol{\epsilon}$ must have magnitudes equal to 2, and the same sign as the corresponding element in \mathbf{s} . i.e.

$$\epsilon_i = \begin{cases} 0 & \text{if } s_i = \hat{s}_i^{\text{ML},\mathcal{S}} \\ -2\hat{s}_i^{\text{ML},\mathcal{S}} & \text{otherwise} \end{cases} \quad (3.23)$$

The linear approximation is stated as follows using the approximate linearity of Γ about $\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}$,

$$\begin{aligned} \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} + \boldsymbol{\epsilon}) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}) &= \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} + \sum_{i=k_i}^{k_f} \epsilon_i \boldsymbol{\delta}_i) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}) \quad \text{by (3.23)} \\ &\approx \sum_{i=k_i}^{k_f} (\Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} + \epsilon_i \boldsymbol{\delta}_i) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}})) \end{aligned} \quad (3.24)$$

Combining (3.22), (3.23) and (3.24), we have,

$$\begin{aligned} \Gamma(\mathbf{s}) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}) &\approx \sum_{i=k_i}^{k_f} (\Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} + \epsilon_i \boldsymbol{\delta}_i) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}})) \\ &= \sum_{i=k_i}^{k_f} \overbrace{(\Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}} - 2\hat{s}_i^{\text{ML},\mathcal{S}} \boldsymbol{\delta}_i) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}))}^{r_i \triangleq} \frac{|s_i - \hat{s}_i^{\text{ML},\mathcal{S}}|}{2} \end{aligned}$$

where r_i is called the Ono reliability information, and can be derived using the definition of the path metric Γ in (3.21),

$$r_i = 4 \left(\sum_{m=0}^v |\hat{h}_m|^2 + \hat{s}_i \Re \left[\sum_{i=0}^v \bar{\hat{h}}_i (x_{i+i} - \sum_{j=0}^v \hat{h}_j \hat{s}_{i-j+i}) \right] \right) \quad (3.25)$$

where $\overline{\hat{h}_j}$ denotes the complex conjugate of \hat{h}_j . In words, the reliability of the i -th symbol is the change in the path metrics associated with changing only the decision of the i -th symbol.

Let κ be an arbitrary positive real number. If we set $\hat{\mathbf{s}}^r = \kappa \hat{\mathbf{s}}^{\text{ML},\mathcal{S}} r_i$, we can obtain, in the following, an approximate correspondence between the path metric of \mathbf{s} and the codeword distance from \mathbf{s} to this infinitely many choices of $\hat{\mathbf{s}}^r$ that differs by the positive scaling κ ,

$$\begin{aligned}
\Gamma(\mathbf{s}) - \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}) &\approx \sum_{i=k_1}^{k_f} \frac{1}{2} r_i \left| s_i - \hat{s}_i^{\text{ML},\mathcal{S}} \right| \\
&= \frac{1}{2} \sum_{i=k_1}^{k_f} r_i (1 - s_i \hat{s}_i^{\text{ML},\mathcal{S}}) && \because \left| s_i - \hat{s}_i^{\text{ML},\mathcal{S}} \right| = \frac{1}{2} \left| s_i - \hat{s}_i^{\text{ML},\mathcal{S}} \right|^2 \\
& && = 1 - s_i \hat{s}_i^{\text{ML},\mathcal{S}} \\
&= \frac{1}{4\kappa} \sum_{i=k_1}^{k_f} \left((s_i - \kappa r_i \hat{s}_i^{\text{ML},\mathcal{S}})^2 - (1 - \kappa r_i)^2 \right) \quad \forall \kappa \in \mathbb{R}^+ \\
&= \frac{1}{4\kappa} \left(\|\mathbf{s} - \hat{\mathbf{s}}^r\|^2 - \sum_{i=k_1}^{k_f} (1 - \kappa r_i)^2 \right)
\end{aligned} \tag{3.26}$$

Hence, the codeword that minimizes the path metrics also approximately minimizes the distance to $\hat{s}_i^r = \kappa r_i \hat{s}_i^{\text{ML},\mathcal{S}}$ as desired, and vice versa,

$$\begin{aligned}
\mathbf{s}^r &\triangleq \arg \min_{\mathbf{s} \in \mathcal{S}_c} \Gamma(\mathbf{s}) \\
&\approx \arg \min_{\mathbf{s} \in \mathcal{S}_c} \Gamma(\hat{\mathbf{s}}^{\text{ML},\mathcal{S}}) + \frac{1}{4\kappa} \left(\|\mathbf{s} - \mathbf{s}^r\|^2 - \sum_{i=k_1}^{k_f} (1 - \kappa r_i)^2 \right) \\
&= \arg \min_{\mathbf{s} \in \mathcal{S}_c} \sum_{i=k_1}^{k_f} \|\mathbf{s} - \mathbf{s}^r\|^2
\end{aligned}$$

The overall process of calculating the hard decisions $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{\text{ML},\mathcal{S}}$ and the reliability metric r_i is called the SO-MLSE algorithm proposed by Ono[11].

Chapter 4

SAIC algorithm design

As described in Chapter 1, downlink co-channel and adjacent channel interference occurs at the mobile station because the base transceiver stations at near-by cells uses the same or adjacent physical channels. The effect of the interference is visualized, in Figure 1-1, as the size of the overlap in spectrograms between the desired signal and the interferer. The objective of the SAIC algorithm is to improve the conventional receiver by exploiting the fact that the interferer is an attenuated GSM signal.

In the following sections, we will follow the same approach in Chapter 3 to derive an approximate discrete-time measurement model with interference. The SAIC algorithm developed by Raghu Challa at Qualcomm based on [10] will be introduced and analyzed based on the further simplified single interferer model.

4.1 Discrete-time Measurement Model with Interference

In the perspectives of the receiver, the co-channel and adjacent channel interferers are just independent GMSK signals carrying random data. The data sequences of the i -th co-channel and j -th adjacent channel interferers can be considered as independent and identically distributed equiprobable sequences of ± 1 , denoted as $\{\mathbf{a}_n^i\}_{n=n_1^i}^{n_2^i}$ and $\{\mathbf{a}_n^j\}_{n=n_1^j}^{n_2^j}$ respectively. The probabilistic model in Figure 3-1 can be used to describe

how the transmitted interferers $\underline{Z}^i(t)$ and $\underline{Z}^j(t)$ can be constructed from the DBPSK approximation to GMSK in (2.4) and the Euclidean image $\{\underline{b}_n^i\}_{n=n_i^i}^{n_f^i}$ and $\{\underline{b}_n^j\}_{n=n_i^j}^{n_f^j}$ of the differentially decoded sequences $\{\underline{d}_n^i\}_{n=n_i^i}^{n_f^i}$ and $\{\underline{d}_n^j\}_{n=n_i^j}^{n_f^j}$ (see (2.1) for differential decoding) of the data sequences $\{\underline{a}_n^i\}_{n=n_i^i}^{n_f^i}$ and $\{\underline{a}_n^j\}_{n=n_i^j}^{n_f^j}$ respectively. Each interferer is corrupted by an independent multipath effect, modeled as the convolution with the channel impulse responses \underline{C}^i and \underline{C}^j . The received signal $X(t)$ becomes the superposition of the signal, multipath faded interferers and the white Gaussian noise process. That is,

$$\begin{aligned}
X(t) &= (\mathbf{H} * \mathbf{S})(t) + \sum_{i=0}^{N^i} (\underline{C}^i * \underline{Z}^i)(t) + \sum_{j=0}^{N^j} (\underline{C}^j * \underline{Z}^j)(t) + \mathbf{N}(t) \\
&= \sum_{m=-\infty}^{\infty} (\mathbf{H} * C_0)(t - mT) \prod_{n=-\infty}^m j \underline{b}_n \\
&\quad + \sum_{i=0}^{N^i} \sum_{m=-\infty}^{\infty} (\underline{C}^i * C_0)(t - mT) \prod_{n=-\infty}^m j \underline{b}_n^i + \\
&\quad + \sum_{j=0}^{N^j} \sum_{m=-\infty}^{\infty} (\underline{C}^j * C_0)(t - mT) \prod_{n=-\infty}^m j \underline{b}_n^j + \mathbf{N}(t)
\end{aligned} \tag{4.1}$$

where N^i and N^j are the numbers of the co-channel and adjacent channel interferers.

Using the same simplification in (2.2) and the finite channel memory approximation in (3.3) (i.e. let $\underline{c}^i = [c_0^i \dots c_{v-1}^i]$ and $\underline{c}^j = [c_0^j \dots c_{v-1}^j]$ be the channel impulse responses of the i -th co-channel and j -th adjacent channel interferers respectively), the discrete-time observation sample generated by the receiver front-end at time k becomes,

$$x_k = \sum_{m=0}^v h_m s_{k-m} + \sum_{i=0}^{N^i} \sum_{m=0}^{v^i} c_m^i z_{k-m}^i + \sum_{j=0}^{N^j} \sum_{m=0}^{v^j} c_m^j z_{k-m}^j + n_k \tag{4.2}$$

where z_k^i and z_k^j are defined as the Euclidean image of the data bit \underline{a}_k^i and \underline{a}_k^j for $k \in [n_i^i, n_f^i]$ and $[n_i^j, n_f^j]$ respectively and 0 otherwise. (4.2) is, therefore, the discrete-time measurement model with interference.

4.2 Single Interferer Model

While the continuous-time and discrete-time measurement models in (4.1) and (4.2) are general enough to model a variety of interference scenarios, they are too complex for analysis. In reality, it is common to have one dominant co-channel interferer, the cancellation of which lead to most of the improvement gain. Thus, most of the SAIC algorithms including the one proposed here are designed and analyzed based on a simpler model of one co-channel interferer, which is assumed to be present at all time to avoid the effect of power ramp-up or ramp-down. More precisely, let \mathbf{z}_k be the symbol, which is *non-zero* at any time k , of the dominant co-channel interferer, and $\mathbf{c} = [c_0 \dots c_u]^T$ be its channel impulse response with finite memory u . From (4.2), the discrete-time single interferer model is,

$$\mathbf{x}_k = \sum_{m=0}^v h_m \mathbf{s}_{k-m} + \sum_{m=0}^u c_m \mathbf{z}_{k-m} + \mathbf{n}_k \quad (4.3)$$

In matrix form, the model becomes,

$$\mathbf{X} = \mathbf{H}\mathbf{S} + \mathbf{C}\mathbf{Z} + \mathbf{N} \quad (4.4)$$

where

$$\mathbf{C} \triangleq \begin{bmatrix} \Re(c_k) & \dots & \Re(c_{k-u}) \\ \Im(c_k) & \dots & \Im(c_{k-u}) \end{bmatrix}, \quad \mathbf{Z} \triangleq \begin{bmatrix} \mathbf{z}_k & \dots & \mathbf{z}_{k+p} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_{k-v} & \dots & \mathbf{z}_{k+p-u} \end{bmatrix} \quad (4.5)$$

Various SAIC algorithms developed based on this single interferer model have three common types of additional model assumptions, described as blind, semi-blind, and training sequence code based. We will adopt the *semi-blind* model, which assumes that the receiver knows the training sequence code of the desired signal but not the dominant interferer. In other words, there is no known data in the interferer to estimate its channel using the conventional second stage channel estimation described in Section 3.2.1. This is the most realistic model for the GSM/GPRS asynchronous network because the random misalignment makes it costly to track and use the training sequence of the interferer, even if the sequence is known. In Appendix A.5, each

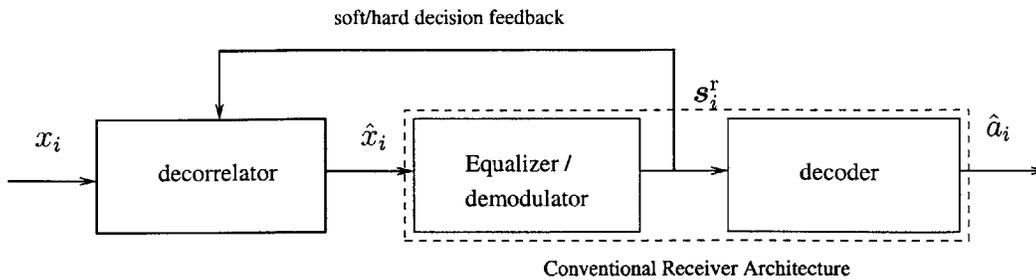


Figure 4-1: Overall SAIC receiver structure

model assumption is briefly described with some SAIC algorithms developed based on those assumptions.

4.3 Proposed Decorrelator-based SAIC Algorithm

The motivation for the proposed SAIC algorithm is to eliminate the observation mostly corrupted by interference so that the remaining observation is approximately interference-free. The conventional signal detection techniques in Section 3.2 can then be applied on the decorrelated observation. Figure 4-1 illustrates the basic SAIC receiver structure. Before getting into the details of the algorithm in the next section, we will introduce three measurement models, called the extended, decorrelated and the optimal decorrelated single interferer models. They reorganize the observation and data in a way convenient for us to describe the decorrelation as matrix multiplication.

4.3.1 Matrix Representation of Decorrelation

The receiver indeed uses two streams of complex-valued observation to compute the decorrelator weight. They are the sequences of odd and even samples obtained from the doubly oversampled continuous-time received signal $X(t)$. They will be denoted by \mathbf{x}^1 and \mathbf{x}^2 in complex vector form, and \mathbf{X}^1 and \mathbf{X}^2 in real matrix form according to (3.4). The single interferer model in (3.5) becomes,

$$\begin{bmatrix} \mathbf{x}^1 \\ \mathbf{x}^2 \end{bmatrix} = \begin{bmatrix} \mathbf{H}^1 \\ \mathbf{H}^2 \end{bmatrix} \mathbf{S} + \begin{bmatrix} \mathbf{C}^1 \\ \mathbf{C}^2 \end{bmatrix} \mathbf{Z} + \begin{bmatrix} \mathbf{N}^1 \\ \mathbf{N}^2 \end{bmatrix} \quad (4.6)$$

where \mathbf{N}^1 and \mathbf{N}^2 are modeled as jointly Gaussian noise that corrupts each subsample sequence.

A simple linear time-invariant decorrelator can be described as a set of space-time finite-impulse-response filter, where the spatial dimension corresponds to the real and imaginary parts of the two consecutive subsamples. To simplify its mathematical representation, we introduce the following extended single interferer model, by simply stacking up the l delayed observation subsamples in the vertical dimension of (4.6),

$$\begin{aligned}
\underline{\mathbf{X}}^{4(l+1) \times (p+1)} \triangleq & \begin{bmatrix} \Re x_k^1 & \dots & \Re x_{k+p}^1 \\ \Im x_k^1 & \dots & \Im x_{k+p}^1 \\ \Re x_k^2 & \dots & \Re x_{k+p}^2 \\ \Im x_k^2 & \dots & \Im x_{k+p}^2 \\ \vdots & \ddots & \vdots \\ \Re x_{k-l}^1 & \dots & \Re x_{k-l+p}^1 \\ \Im x_{k-l}^1 & \dots & \Im x_{k-l+p}^1 \\ \Re x_{k-l}^2 & \dots & \Re x_{k-l+p}^2 \\ \Im x_{k-l}^2 & \dots & \Im x_{k-l+p}^2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{H}^1 & \mathbf{0}^{2 \times l} \\ \mathbf{H}^2 & \mathbf{0}^{2 \times l} \\ \vdots & \vdots \\ \mathbf{0}^{2 \times l} & \mathbf{H}^1 \\ \mathbf{0}^{2 \times l} & \mathbf{H}^2 \end{bmatrix}}_{\underline{\mathbf{H}}^{4(l+1) \times (v+l+1)} \triangleq} \underbrace{\begin{bmatrix} s_k & \dots & s_{k+p} \\ \vdots & \ddots & \vdots \\ s_{k-v-l} & \dots & s_{k-v-l+p} \end{bmatrix}}_{\underline{\mathbf{S}}^{(v+l+1) \times (p+1)} \triangleq} \\
& + \underbrace{\begin{bmatrix} \mathbf{C}^1 & \mathbf{0}^{2 \times l} \\ \mathbf{C}^2 & \mathbf{0}^{2 \times l} \\ \vdots & \vdots \\ \mathbf{0}^{2 \times l} & \mathbf{C}^1 \\ \mathbf{0}^{2 \times l} & \mathbf{C}^2 \end{bmatrix}}_{\underline{\mathbf{C}}^{4(l+1) \times (u+l+1)} \triangleq} \underbrace{\begin{bmatrix} z_k & \dots & z_{k+p} \\ \vdots & \ddots & \vdots \\ z_{k-u-l} & \dots & z_{k-u-l+p} \end{bmatrix}}_{\underline{\mathbf{Z}}^{(u+l+1) \times (p+1)} \triangleq} + \underbrace{\begin{bmatrix} \Re n_k^1 & \dots & \Re n_{k+p}^1 \\ \Im n_k^1 & \dots & \Im n_{k+p}^1 \\ \Re n_k^2 & \dots & \Re n_{k+p}^2 \\ \Im n_k^2 & \dots & \Im n_{k+p}^2 \\ \vdots & \ddots & \vdots \\ \Re n_{k-l}^1 & \dots & \Re n_{k-l+p}^1 \\ \Im n_{k-l}^1 & \dots & \Im n_{k-l+p}^1 \\ \Re n_{k-l}^2 & \dots & \Re n_{k-l+p}^2 \\ \Im n_{k-l}^2 & \dots & \Im n_{k-l+p}^2 \end{bmatrix}}_{\underline{\mathbf{N}}^{4(l+1) \times (p+1)} \triangleq} \\
\underline{\mathbf{X}} = \underline{\mathbf{H}}\underline{\mathbf{S}} + \underline{\mathbf{C}}\underline{\mathbf{Z}} + \underline{\mathbf{N}} & \tag{4.7}
\end{aligned}$$

The decorrelation is now equivalent to pre-multiplying the observed $\underline{\mathbf{X}}$ by the decorrelator matrix $\underline{\mathbf{W}}^{2 \times 4(l+1)}$. The decorrelated observation $\underline{\mathbf{X}}\underline{\mathbf{W}}$, which contain the two rows of observation sequences used as the input to the conventional signal detection, can be described by the decorrelator single interferer model below,

$$\underline{\mathbf{W}}\underline{\mathbf{X}} = \underline{\mathbf{W}}\underline{\mathbf{H}}\underline{\mathbf{S}} + \underline{\mathbf{W}}\underline{\mathbf{C}}\underline{\mathbf{Z}} + \underline{\mathbf{W}}\underline{\mathbf{N}} \tag{4.8}$$

The objective is to choose the decorrelator weights in $\underline{\mathbf{W}}$ such that (4.8) is approxi-

mately an interference-free additive white Gaussian noise model like (3.5). In other words, the optimal choice of \mathbf{W} will bring (4.8) close to the following,

$$\mathbf{W}\tilde{\mathbf{X}} \approx \mathbf{H}^{\mathbf{W}}\mathbf{S} + \mathbf{N}^{\mathbf{W}} \quad (4.9)$$

where $\mathbf{N}^{\mathbf{W}}$ is white Gaussian noise independent of \mathbf{S} , the signal-to-noise ratio remains unchanged compared with (4.7), the interferer is completely eliminated and the channel memory remains as v (i.e. $\mathbf{H}^{\mathbf{W}}$ is a 2-by- v matrix). We will call this the optimal decorrelated single interferer model because it expresses a set of desired criteria to achieve by optimizing \mathbf{W} .

4.3.2 Joint Decorrelator-channel-data Estimation

The set of criteria in (4.9) is often conflicting. For instance, eliminating the interferer completely may reduce the signal-to-noise ratio because it also eliminates the signal that is indistinguishable from the interferer (e.g. consider $\mathbf{H} = \mathbf{C}$ as an extreme case). It appears, therefore, that optimizing one particular criterion may worsen the other criteria too much that the overall optimality is hampered. While the overall optimality criterion is to improve signal detection or, in other words, to minimize the probability of error decoding the transmitted messages, it is rather difficult to express this simple overall criterion precisely in terms of \mathbf{W} , let alone optimizing \mathbf{W} for it. To avoid this difficulty, we will adopt the multistage optimization approach in Section 3.2 and use the least squares optimality criteria for \mathbf{W} . The overall multistage algorithm will be described in this section while the details on the LS decorrelator optimization will be described in the next section.

Suppose an optimal \mathbf{W} exists such that the approximation in (4.9) holds. i.e.

$$\mathbf{W}\tilde{\mathbf{X}} = \mathbf{H}\mathbf{S} + \mathbf{N} \quad (4.10)$$

where the superscript \mathbf{W} on the channel and noise is removed for notational simplicity. We can describe the entire decorrelator-based SAIC algorithm as a joint optimization

over the parameter \mathbf{W} , the channel \mathbf{H} and the symbol matrix \mathbf{S} , for a given extended observation matrix $\underline{\mathbf{X}}$. Using the same multistage estimation approach described in Section 3.2, we can break down this joint optimization into the following three stages with two types of iterations called the inner loop and the outer loop:

1. In the first stage, we obtain the CAZAC channel estimate in (3.17) using the known training sequence code and assuming the interference free model in (3.6).
 [In the subsequent outer loop iterations (to be described with Stage 3), LS channel estimate in (3.8) is used assuming the model (4.10).]
2. In the second stage, we calculate the optimal parameter \mathbf{W} by (4.11) in the least squares sense given the CAZAC channel estimate and the known training sequence code. The first two stages may be iterated several times for convergence and this iteration is called the *inner loop*.
3. In the third stage, the channel estimate and the decorrelated observation $\mathbf{W}\underline{\mathbf{X}}$ obtained previously can be used to generate the soft decisions for the unknown symbol sequence by the SO-MLSE algorithm described in Section 3.2.2. Again, for convergence, the entire estimation process can be iterated several times by feeding back the ML symbol sequence. We will call this iteration the *outer loop*, which contains the inner loop.

Each iteration of the outer loop is costly, mainly because of the computation of the pseudoinverse. Thus, only the reliable symbol estimates selected based on the Ono's reliability information are fed back. Since this feedback symbol sequence does not have the constant amplitude zero-autocorrelation property in (3.10), the LS channel estimate instead of the CAZAC channel estimate need to be used in the first stage.

Algorithm 4-2 is the pseudocode that describes the above multistage optimization more precisely. $\text{Restructure}_{\text{Form 1} \rightarrow \text{Form 2}}[\text{arg}]$ is an operator that converts arg from the format Form 1 to the format Form 2. For convenience, the equations for the restructuring operations are referenced in the pseudocode. N^{inner} and N^{outer} are the numbers of

Procedure 4-2 Pseudocode for the proposed SAIC algorithm

function $\hat{\mathbf{s}}^r \leftarrow \text{SAIC}[\{x_n\}_{n=n_i+v}^{n_f}, \{s_n\}_{n=-v}^{25-v}, N^{\text{outer}}, N^{\text{inner}}, \eta]$
Require: $N^{\text{outer}}, N^{\text{inner}} \in \mathbb{Z}^+, \eta \in \mathbb{R}^+$
 $\hat{\mathbf{H}} \leftarrow \text{Restructure}_{\mathbf{h} \rightarrow \hat{\mathbf{H}}}[\hat{\mathbf{h}}^{\text{CAZAC}}[\{x_n\}_{n=0}^{25-v}, \{s_n\}_{n=-v}^{25-v}]]$ {see (3.4) and (3.17)}

 $\hat{\mathbf{S}} \leftarrow \text{Restructure}_{\{s\} \rightarrow \hat{\mathbf{S}}}[\{s_n\}_{n=-v}^{25-v}]$ {see (3.4)}

 $\hat{\mathbf{X}} \leftarrow \text{Restructure}_{\{x\} \rightarrow \hat{\mathbf{X}}}[\{x_n\}_{n=0}^{25-v}]$ {see (4.7)}

for $i \leftarrow 1$ to N^{outer} **do**

 for $j \leftarrow 1$ to N^{inner} **do**

 $\hat{\mathbf{W}} \leftarrow \hat{\mathbf{H}} \hat{\mathbf{S}} \hat{\mathbf{X}}^\dagger$

 $\hat{\mathbf{H}} \leftarrow \hat{\mathbf{W}} \hat{\mathbf{X}} \hat{\mathbf{S}}^\dagger$

 end for

 $\hat{\mathbf{X}} \leftarrow \hat{\mathbf{W}} \hat{\mathbf{X}}$

 $\hat{\mathbf{s}}^r \leftarrow \text{SO-MLSE}[\text{Restructure}_{\hat{\mathbf{X}} \rightarrow \{x\}}[\hat{\mathbf{X}}], \text{Restructure}_{\hat{\mathbf{H}} \rightarrow \mathbf{h}}[\hat{\mathbf{H}}]]$ {see Section 3.2.2 and (3.4)}

 $\hat{\mathbf{S}} \leftarrow \text{Restructure}_{\hat{\mathbf{s}} \rightarrow \hat{\mathbf{S}}}[\hat{\mathbf{s}}^r]$ {see (3.4)}

 $\hat{\mathbf{X}} \leftarrow \text{Restructure}_{\{x\} \rightarrow \hat{\mathbf{X}}}[\{x_n\}_{n=n_i+v}^{n_f}]$ {see (4.7)}

 for $\hat{\mathbf{s}}_k$ and $\hat{\mathbf{x}}_k$ equal to each column of $\hat{\mathbf{S}}$ and $\hat{\mathbf{X}}$ respectively **do**

 if $\frac{\|\hat{\mathbf{H}} \hat{\mathbf{s}}_k\|^2}{\|\hat{\mathbf{H}}\|^2} < \eta$ **then**

 Remove the columns $\hat{\mathbf{s}}_k$ and $\hat{\mathbf{x}}_k$ from $\hat{\mathbf{S}}$ and $\hat{\mathbf{X}}$ respectively

 end if

 end for
end for

inner and outer loops respectively. To reduce complexity, N^{inner} and N^{outer} are chosen to be small. More specifically, we choose $N^{\text{inner}} = 1$, and $N^{\text{outer}} = 1$ or 2 for the computer simulation. η is the threshold for selecting the reliable symbol estimates. Since the soft-decision is normalized, η corresponds approximately to the number of decisions to feed back. In particular, setting $\eta = 0$ dB will most probably prune out all symbols, while setting $\eta = -100$ dB is likely to include all symbols.

4.3.3 Least Squares Optimization of Decorrelator

In this section, we will first state the least squares criterion used to optimize the decorrelator, and then justify the validity of the choice by its asymptotic behavior by a close examination of the inner loop of first stage channel estimation and second stage decorrelator optimization.

Given the symbol matrix estimate $\hat{\mathbf{S}}$, the channel estimate $\hat{\mathbf{H}}$, and the extended observation matrix $\underline{\mathbf{X}}$, the LS decorrelator is defined as follows,

$$\begin{aligned}\hat{\mathbf{W}} &= \arg \min_{\mathbf{W}} \|\mathbf{W}\underline{\mathbf{X}} - \hat{\mathbf{H}}\hat{\mathbf{S}}\|_{\text{F}}^2 \\ &= \hat{\mathbf{H}}\hat{\mathbf{S}}\underline{\mathbf{X}}^\dagger\end{aligned}\quad (4.11)$$

The n -th iteration of the inner loop can be re-expressed as follows,

$$\hat{\mathbf{H}}[0] = \text{Restructure}_{\mathbf{h} \rightarrow \hat{\mathbf{H}}}[\hat{\mathbf{h}}^{\text{CAZAC}}[\{x_n\}_{n=0}^{25-v}, \{s_n\}_{n=-v}^{25-v}]] \quad (4.12a)$$

$$\hat{\mathbf{W}}[n] = \zeta[n]\hat{\mathbf{H}}[n-1]\hat{\mathbf{S}}\underline{\mathbf{X}}^\dagger \quad (4.12b)$$

$$\hat{\mathbf{H}}[n] = \hat{\mathbf{W}}[n]\underline{\mathbf{X}}\hat{\mathbf{S}}^\dagger \quad (4.12c)$$

where the notation $[n]$ indicates the optimization result of the n -th iteration, and $\zeta[n]$ is a normalization constant to ensure $\hat{\mathbf{W}}$ has a fixed energy, say ζ .¹ (4.12) can be re-expressed as two decoupled difference equations by substituting (4.12b) into

¹ $\zeta[n]$ could have been chosen arbitrarily because it corresponds to a fixed scaling of the observation $\underline{\mathbf{X}}$, which neither enhance nor degrade performance

(4.12c) and vice versa,

$$\hat{\mathbf{W}}[n] = \zeta[n] \hat{\mathbf{W}}[n-1] \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger \quad (4.13a)$$

$$\hat{\mathbf{H}}[n] = \zeta[n] \hat{\mathbf{H}}[n-1] \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger \quad (4.13b)$$

The form of each recursion is indeed equivalent to the *power method*[14] used to estimate the largest eigenvalue. Each row of $\hat{\mathbf{W}}$ (i.e. $\hat{\mathbf{w}}'_1$ and $\hat{\mathbf{w}}'_2$) and $\hat{\mathbf{H}}$ (i.e. $\hat{\mathbf{h}}'_1$ and $\hat{\mathbf{h}}'_2$) converges to a real vector parallel to the maximum left eigenvector of the corresponding system matrix,

$$\begin{aligned} \lim_{n \rightarrow \infty} \hat{\mathbf{w}}_1[n] &= \lim_{n \rightarrow \infty} \hat{\mathbf{w}}_2[n] \\ &= \kappa_1 \text{max-eigenvec } \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger \end{aligned} \quad (4.14a)$$

$$\begin{aligned} \lim_{n \rightarrow \infty} \hat{\mathbf{h}}_1[n] &= \lim_{n \rightarrow \infty} \hat{\mathbf{h}}_2[n] \\ &= \kappa_2 \text{max-eigenvec } \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger \end{aligned} \quad (4.14b)$$

where max-eigenvec is the operation that returns the unit-length maximum eigenvector of its matrix argument. κ_1 is arbitrary because of the arbitrary energy constraint ζ . κ_2 is not arbitrary, however, and it must satisfy (4.12c) that

$$\kappa_2 \text{max-eigenvec } \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger = (\kappa_1 \text{max-eigenvec } \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger \hat{\mathbf{S}} \underline{\mathbf{X}}^\dagger) \underline{\mathbf{X}} \hat{\mathbf{S}}^\dagger$$

Note that the Hermitian operations $(\cdot)'$ here are equivalent to the transposition operations $(\cdot)^T$ because all the vectors and matrices are real.

The result in (4.14) has two significant implications. The obvious one is that it is not optimal to let n goes to infinity. This is because the two output decorrelated observation streams $\hat{\mathbf{w}}'_1 \underline{\mathbf{X}}$ and $\hat{\mathbf{w}}'_2 \underline{\mathbf{X}}$ collapse into one as $\hat{\mathbf{w}}_1$ becomes equal to $\hat{\mathbf{w}}_2$. In other words, the information of the desired signal in the direction orthogonal to $\hat{\mathbf{w}}'_1 \underline{\mathbf{X}} = \hat{\mathbf{w}}'_2 \underline{\mathbf{X}}$ is lost. To see this loss more easily, we can consider a specific interference free case without subsampling nor stacking in time (i.e. $l = 1$). If there is no intersymbol interference (i.e. $v = 0$), it is information lossless to set $\hat{\mathbf{w}}'_1 = \hat{\mathbf{w}}'_2 =$

$[\Re h_0 \Im h_0]$ by the theorem of irrelevance for this particular additive white Gaussian noise model, assuming that the single-tap channel impulse response h_0 is known. If there is intersymbol interference (i.e. $v > 0$), however, having $\hat{\mathbf{w}}'_1 = \hat{\mathbf{w}}'_2$ loses the information in its orthogonal direction.

The second implication of the result is less obvious but the idea is that the direction of the maximum eigenvector is special in helping us to eliminate the interferer. To understand that, we will analyze a joint optimization (not multi-stage) over a one dimensional decorrelator \mathbf{w}' and the corresponding channel \mathbf{h} , and show that the solution is indeed the eigenvectors in (4.14) by choosing the optimality criterion as the decorrelated-observation-to-interference-and-noise-ratio (DOINR) defined as follows,

$$\text{DOINR} \triangleq \frac{\mathbb{E}[\|\mathbf{w}'\underline{\mathbf{X}}\|^2]}{\mathbb{E}[\|\mathbf{w}'\underline{\mathbf{X}} - \mathbf{h}'\mathbf{S}\|^2]} \quad (4.15)$$

where \mathbf{h}' and \mathbf{w}' are functions of the extended observation matrix $\underline{\mathbf{X}}$. The numerator is the expected energy of the decorrelated observation, while the denominator is the expected energy of the noise and interference assuming that the $-\mathbf{h}'\mathbf{S}$ term subtract out the signal from the decorrelated observation. The joint optimization of DOINR is stated as follows,

$$\begin{aligned} (\hat{\mathbf{w}}', \hat{\mathbf{h}}') &= \arg \max_{(\mathbf{w}'(\underline{\mathbf{X}}), \mathbf{h}'(\underline{\mathbf{X}}))} \text{DOINR} \\ &= \arg \max_{(\mathbf{w}'(\underline{\mathbf{X}}), \mathbf{h}'(\underline{\mathbf{X}}))} \frac{\mathbb{E}[\|\mathbf{w}'\underline{\mathbf{X}}\|^2]}{\mathbb{E}[\|\mathbf{w}'\underline{\mathbf{X}} - \mathbf{h}'\mathbf{S}\|^2]} \\ &= \arg \max_{(\mathbf{w}'(\underline{\mathbf{X}}), \mathbf{h}'(\underline{\mathbf{X}}))} \frac{\|\mathbf{w}'\underline{\mathbf{X}}\|^2}{\|\mathbf{w}'\underline{\mathbf{X}} - \mathbf{h}'\mathbf{S}\|^2} \quad \text{given } \underline{\mathbf{X}} = \underline{\mathbf{X}} \end{aligned} \quad (4.16)$$

The last equation can be proved by contradiction: suppose there exists at least one $\underline{\mathbf{X}}$ such that $\frac{\|\mathbf{w}'\underline{\mathbf{X}}\|^2}{\|\mathbf{w}'\underline{\mathbf{X}} - \mathbf{h}'\mathbf{S}\|^2}$ is not minimum, DOINR can be further reduced by changing the value of \mathbf{w}' and \mathbf{h}' evaluated at each particular $\underline{\mathbf{X}}$ to minimize $\frac{\|\mathbf{w}'\underline{\mathbf{X}}\|^2}{\|\mathbf{w}'\underline{\mathbf{X}} - \mathbf{h}'\mathbf{S}\|^2}$. An easier but less vigorous way to understand this is to consider the expectation as an average over multiple normal burst, and $\underline{\mathbf{X}}$ as the realization corresponding to one normal burst. Then, optimizing the average over multiple bursts is equivalent to

optimizing over each burst.

The joint optimization can be done in two steps, by first finding \mathbf{h}' in terms of an arbitrary \mathbf{w}' , and then substitute that into the DOINR again to find the optimal \mathbf{w}' . Knowing the optimal \mathbf{w}' , the optimal \mathbf{h}' can then be calculated using the expression found in the first step. The following describes the two steps in detail,

1. From the denominator of (4.16), we see that the optimal \mathbf{h}' for any arbitrary \mathbf{w}' is $\mathbf{w}' \underline{\mathbf{X}} \mathbf{S}^\dagger$, as it minimizes the denominator for any \mathbf{w}' . Hence,

$$\hat{\mathbf{h}}'(\underline{\mathbf{X}}, \hat{\mathbf{w}}') = \hat{\mathbf{w}}' \underline{\mathbf{X}} \mathbf{S}^\dagger \quad (4.17)$$

2. Imposing the relation (4.17) on \mathbf{h} in (4.16) reduces the joint optimization into a single optimization as follows,

$$\begin{aligned} \hat{\mathbf{w}}' &= \arg \max_{\mathbf{w}'(\underline{\mathbf{X}})} \frac{\|\mathbf{w}' \underline{\mathbf{X}}\|^2}{\|\mathbf{w}' \underline{\mathbf{X}} - \mathbf{w}' \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S}\|^2} \\ &= \arg \max_{\mathbf{w}'(\underline{\mathbf{X}})} \frac{\mathbf{w}' \underline{\mathbf{X}} \underline{\mathbf{X}}' \mathbf{w}}{\mathbf{w}' \underline{\mathbf{X}} (\mathbf{I} - \mathbf{S}^\dagger \mathbf{S}) \underline{\mathbf{X}}' \mathbf{w}} \\ &= \arg \max_{\mathbf{w}'(\underline{\mathbf{X}})} \left(1 - \frac{\mathbf{w}' \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \mathbf{w}}{\mathbf{w}' \underline{\mathbf{X}} \underline{\mathbf{X}}' \mathbf{w}} \right)^{-1} \\ &= \arg \max_{\mathbf{w}'(\underline{\mathbf{X}})} \frac{\mathbf{w}' \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \mathbf{w}}{\mathbf{w}' \underline{\mathbf{X}} \underline{\mathbf{X}}' \mathbf{w}} \end{aligned}$$

Let $\underline{\mathbf{X}} \underline{\mathbf{X}}' = \mathbf{U} \Sigma^2 \mathbf{U}'$ be the reduced form of the singular value decomposition $\begin{bmatrix} \mathbf{U} & \mathbf{U}^\perp \end{bmatrix} \begin{bmatrix} \Sigma^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}' \\ (\mathbf{U}^\perp)'\end{bmatrix}$, where Σ^2 is an invertible diagonal matrix. Then,

$$\begin{aligned} \hat{\mathbf{w}}' &= \arg \max_{\mathbf{w}'(\underline{\mathbf{X}})} \frac{\mathbf{w}' \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \mathbf{w}}{\mathbf{w}' \mathbf{U} \Sigma^2 \mathbf{U}' \mathbf{w}} \\ &= \left(\arg \max_{\mathbf{y}'(\underline{\mathbf{X}})} \frac{\mathbf{y}' \Sigma^{-1} \mathbf{U} \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \mathbf{U}' \Sigma^{-1} \mathbf{y}}{\|\mathbf{y}\|^2} \right) \Sigma^{-1} \mathbf{U} \quad \mathbf{y}' \triangleq \mathbf{w}' \mathbf{U}' \Sigma \\ \hat{\mathbf{w}} &= \kappa_1 \mathbf{U}' \Sigma^{-1} \text{max-eigenvec } \Sigma^{-1} \mathbf{U} \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \mathbf{U}' \Sigma^{-1} \\ &= \kappa_1 \text{max-eigenvec } \underline{\mathbf{X}} \mathbf{S}^\dagger \mathbf{S} \underline{\mathbf{X}}' \end{aligned} \quad (4.18)$$

$$= \lim_{n \rightarrow \infty} \hat{\mathbf{w}}_1[n] \quad \text{by (4.14a)} \quad (4.19)$$

where κ_1 is some constant scalar. The last two equations are obtained by expanding the Hermitian matrix $\underline{\mathbf{X}}\mathbf{S}^\dagger\mathbf{S}\underline{\mathbf{X}}'$ into its singular value decomposition and performing the optimization similar to the one described in Appendix A.3. Substituting (4.19) into (4.17) gives,

$$\begin{aligned}\hat{\mathbf{h}}' &= \lim_{n \rightarrow \infty} \mathbf{w}_1[n]' \underline{\mathbf{X}}\mathbf{S}^\dagger \\ &= \lim_{n \rightarrow \infty} \mathbf{h}_1[n]' && \text{by (4.12c)} \\ &= \kappa_2 \text{max-eigenvec } \hat{\mathbf{S}}\underline{\mathbf{X}}^\dagger\underline{\mathbf{X}}\hat{\mathbf{S}}^\dagger && \text{by (4.14b)}\end{aligned}\tag{4.20}$$

$$(4.21)$$

Hence, the two step optimizations above showed that the maximum DOINR decorrelator in (4.19) and (4.20) are identical to the LS decorrelator (4.14) as the number of inner loop iterations goes to infinity. In other words, If we fix the decorrelated observation energy by appending an arbitrary gain, the inner loop gradually eliminates the interference and noise energy in the model, which is therefore desired. However, in doing so, it collapses the two decorrelated observation streams into one, which causes information loss in the direction orthogonal to the decorrelator. With this argument, in addition to computational complexity and simulation results, we chooses not to do any addition inner loop iterations (i.e. $N^{\text{inner}} = 1$).

Chapter 5

Computer Simulation

The proposed decorrelator-based SAIC algorithm described in Chapter 4 was implemented in the GSM/GPRS simulation based on the conventional receiver architecture described in Chapter 3. In the following sections, we will describe the two DARP test scenarios used to test the algorithm and states the simulation results.

The SAIC/DARP Performance Evaluation is a set of tests for the SAIC feature in the GSM/GPRS downlink receiver. In the current revision[12][1], there are five proposed DARP test scenarios, which mainly differ in the number of interferers, the assumption of the training sequence and time alignment between the interference and the signal. The first two test scenarios, named DTS-1 and DTS-2 in Table 5.1, were implemented and run on four different types of traffic channels that differ mainly in the channel coding scheme. To pass the tests, the frame error rates detected by the receiver, the residual bit error rates of Class 1b and Class 2 bits, if applicable¹, need to be small enough so that the receiver is usable. For example, the frame error rate must be below 1% for all traffic channels. Otherwise, the transmitted speech cannot be recovered and the entire speech frame, which covers several bursts, need to be discarded (see Section 2.2).

Each test scenario in Table 5.1 specifies four key parameters in the interference model. The *interfering signal* column specifies all types of interferer signal present in

¹Two out of the five selected traffic channels, namely TCH/AFS 12.2 and 5.9, do not have Class 2 bits and so Class 2 bit error rate does not apply. Hence, Figure 5-3 does not have the plots for those two logical channels.

Table 5.1: SAIC/DARP reference test scenarios

Test scenario	Interfering signal	Interferer relative power level	TSC	Interferer delay range
DTS-1	CCI1	0 dB	none	no delay
DTS-2	CCI1	0 dB	none	no delay
	CCI2	-10 dB	none	no delay
	ACI1	3 dB	none	no delay
	AWGN	-17 dB	-	-

the model. CCI1 and CCI2 stands for the first (dominant) and the second co-channel interferers respectively. ACI1 stands for the adjacent channel interferer, and AWGN stands for the additive white Gaussian noise. The *interferer relative power level* column gives the ratio of the power of the specified interferer to that of the dominant co-channel interferer CCI1. This ratio is therefore 0dB for the CCI1 itself. Note that the power is measured at the receiver but before the receiver front end. Thus, the adjacent channel interferer in DTS-2 is even 3dB stronger than the dominant co-channel interferer because the source of adjacent channel interferer is closer than that of the co-channel interferer. The *TSC* column specifies whether the interferer is modulated with a training sequence code. In both test scenarios, it has the value none for all interferers, which means that no training sequence codes are used. In other words, all the modulation bits of the interferers are randomly generated. The *interferer delay range* column specifies whether there is any delay in the power profile between the signal and the interference. In both test scenarios, however, there is no such delay, which means that the interferer is present at all time.

For each test scenario being run on each logical channel, the carrier-to-dominant-interferer-ratio (CIR) is varied over a range such that we can interpolate at the C/I_1 that give exactly 1% frame error rate. Note that by changing CIR, we also change the power of all interferers and noise but their relative power are fixed according to the interferer relative power level specified for each test scenario. At the particular CIR at which the frame error rate is 1%, we obtain the residual Class 1b bit error rate and the residual Class2 bit error rate to compare with the required levels. For show the gain in doing SAIC, we will express all these performance metrics (i.e. CIR, residual Class 1b and 2 bit error rates at 1% frame error) as an improvement measured in dB

on the performance of the conventional receiver.

The simulation results are shown in Figure 5-1. Consider Figure 5-1 in particular, there are five subplots, each corresponds to the simulation on one of the five logical channels: full-rate speech traffic channel (TCH/FS), adaptive multirate full-rate traffic channels with rates 12.2 and 5.9 (TCH/AFS 12.2 and 5.9), the adaptive multirate half-rate traffic channels with rates 6.7 and 5.9 (TCH/AHS 6.7 and 5.9) (see [3] for details). In each subplot, the SAIC algorithms in Algorithm 4-2 are run with three different set of parameters as follows,

SAIC without feedback There is no decision feedback ($N^{\text{outer}} = 1$).

SAIC with feedback There is decision feedback ($N^{\text{outer}} = 2$) but no pruning (i.e. $\eta = -100$ dB).

SAIC with feedback & pruning There is both decision feedback ($N^{\text{outer}} = 2$) and pruning (i.e. $\eta = -0.1$ dB).

Each of these variants of the SAIC algorithm is run for the two test scenarios, DTS-1 and DTS-2. The result is plotted as a bar chart whereby the lower blue bar and upper red bars indicates the SAIC gain in DTS-1 and DTS-2 respectively. In addition, the requirement on DTS1 and DTS2 are shown as the blue and red the dashed vertical lines on the left and right respectively.

From the simulation result, we conclude the benefit of the SAIC algorithm with pruning and decision feedback. It passes both test cases in the current SAIC/DARP revision.

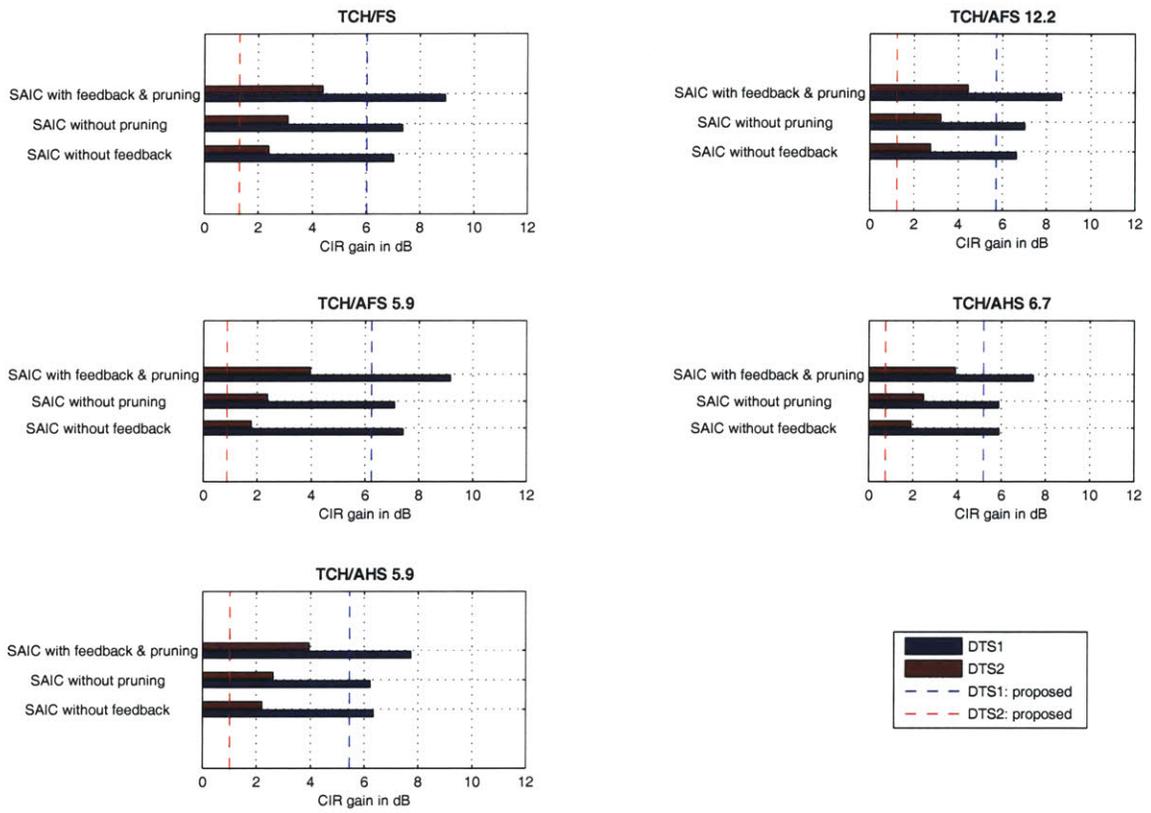


Figure 5-1: Drop in C/I_1 at 1% frame error rate

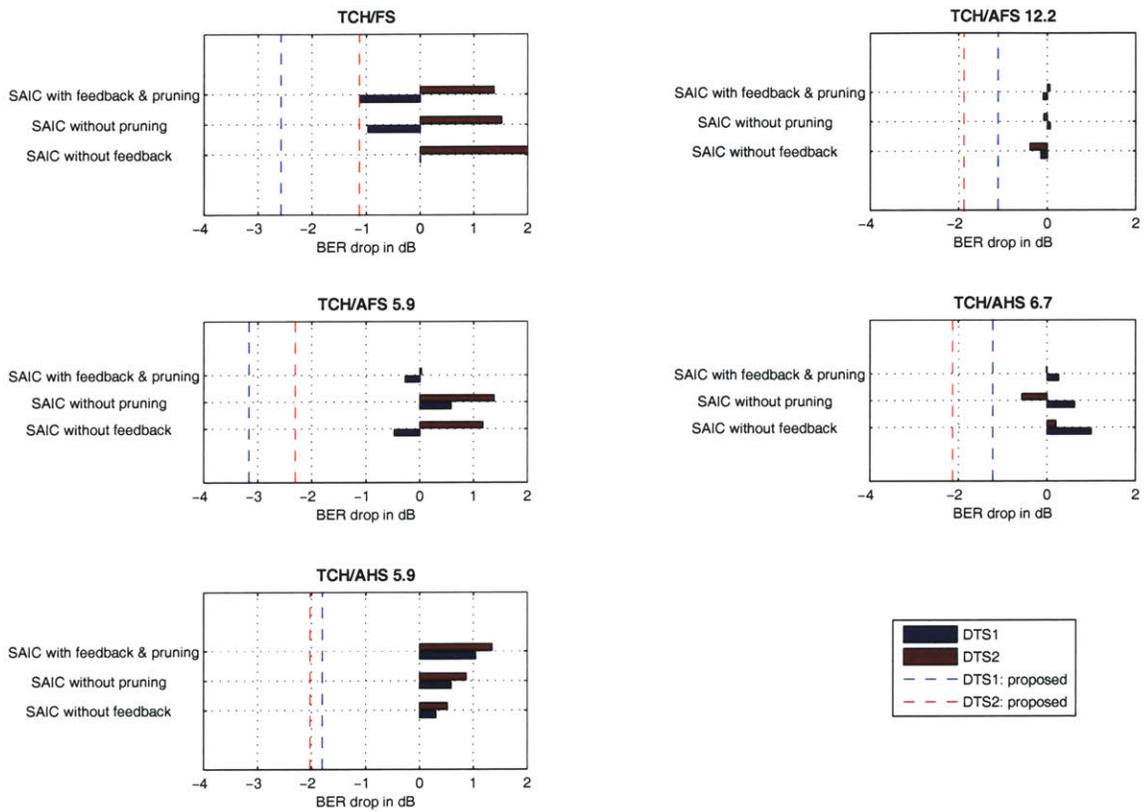


Figure 5-2: Drop in Class 1b bit error rates at 1% frame error rate

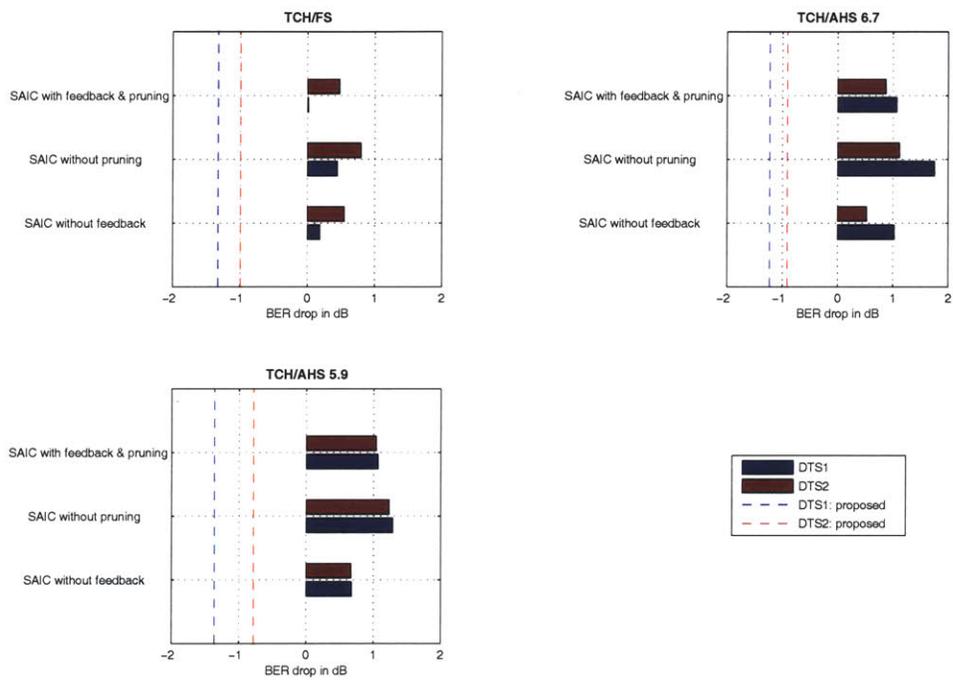


Figure 5-3: Drop in Class 2 bit error rates at 1% frame error rate

Appendix A

Background Theory

A.1 Channel Coding of Full Rate Traffic Channel

Figure A-1 illustrates the channel coding of the full-rate traffic channel. The vocoder encodes a speech segment into a 260-bit sequence d_0, \dots, d_{259} represented by $\{d_k\}_{k=0}^{295}$. The first 50 Class 1a bits $\{d_k\}_{k=0}^{49}$ are CRC-protected by adding 3 parity bits $\{p_k\}_{k=0}^2$ computed using the generator polynomial $g(D) = 1 + D^2 + D^3$. The Class 1 bits $\{d_k\}_{k=0}^{181}$ are then reordered to form the sequence $\{u_k\}_{k=0}^{188}$. More precisely, the 91 even bits $\{d_{2k}\}_{k=0}^{90}$ gets mapped sequentially to the 91 bits of $\{u_k\}_{k=0}^{90}$, followed by the 3 parity bits $\{u_k\}_{k=91}^{93} = \{p_k\}_{k=0}^2$. The *reversed* odd bits $\{d_{2k+1}\}_{k=0}^{90}$ gets mapped to the next 91 bits $\{u_k\}_{k=94}^{184}$. 4 tail bits of 0's are appended to the end $\{u_k\}_{k=185}^{188}$.

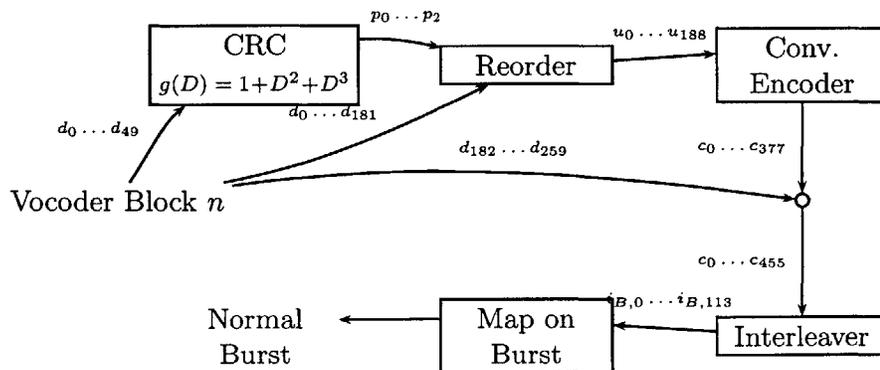


Figure A-1: Channel coding of full rate traffic channel

The reordered class 1 bits are then convolutionally encoded with the generator polynomial matrix,

$$\begin{aligned} G_0 &= 1 && +D^3 & +D^4 \\ G_1 &= 1 & +D & +D^3 & +D^4 \end{aligned}$$

The even and odd bits of the output $\{c_k\}_{k=0}^{377}$ are the output of filtering $\{u_k\}_{k=0}^{188}$ with $G_0(z^{-1})$ and $G_1(z^{-1})$ respectively. The 78 Class 2 bits $\{d_k\}_{k=182}^{259}$ are then appended to the end $\{c_k\}_{k=378}^{455}$ before interleaving.

The interleaver half fills each of the eight (indexed by $B \in \{B_0+4n, \dots, B_0+4n+7\}$) 114-bit blocks $\{i_{B,j}\}_{j=0}^{113}$ with the n -th 456-bit channel-coded vocoder block $\{c_{n,k}\}_{k=0}^{455}$, as follows,

$$\begin{aligned} i_{B,j} &= c_{n,k} \quad , \text{where} \\ B &= B_0 + 4n + (k \bmod 8) \\ j &= 2((49k) \bmod 57) + ((k \bmod 8) \text{div} 4) \end{aligned}$$

Finally, the B -th sequence $\{i_{B,j}\}_{j=0}^{113}$ together with two stealing flags $\{hl_B, hu_B\}$, which are indicators of whether the left and right bursts are stolen for signaling purposes, fills the left and right bursts of the B -th normal burst,

$$\{e_{B,j}\}_{j=0}^{115} = \{i_{B,j}\}_{j=0}^{56}, hl_B, hu_B, \{i_{B,j}\}_{j=57}^{113}$$

The sequences $\{e_{B,j}\}_{j=57}^0$ and $\{e_{B,j}\}_{j=58}^{115}$ will be mapped to the left and right bursts respectively in the normal burst shown in Figure 2-1.

A.2 Amplitude Modulation Decomposition of GMSK Signal

Theorem A.2.1. *If we extend the symbol sequence $\{a_n\}_{n=n_i}^{n_i}$ to $\{a_n\}_{n=-\infty}^{\infty}$ in (2.3) by zero padding the two ends, the extended GMSK signal $S(t)$ can be approximated as*

$$\sum_{K=0}^{M-1} \sum_{N=-\infty}^{\infty} A_{K,N} C_K(t - NT)$$

where $\sum_{N=-\infty}^{\infty} A_{K,N} C_K(t - NT)$ is the K -th amplitude modulated signal with pulse shape $C_K(t - NT)$ modulated by the transformed symbol $A_{K,N}$ that depends only on the original symbol sequence. The number M of different decomposed pulses and the maximum pulse width are 2^{L-1} and $L + 1$ respectively where L is the approximated length that contains most of the energy of the frequency pulse shape (per unit T) of the GMSK signal.

Proof. Let $\rho_N(t)$ be the rectangular function which is 1 within the interval $[NT, (N + 1)T]$ and 0 otherwise. Ignoring the convergence issue related to extending the symbol sequence, the transmitted signal from (2.3) becomes,

$$\begin{aligned} S(t) &\approx \exp\left(j \sum_{n=-\infty}^{\infty} a_n \phi_n(t)\right) \\ &= \sum_{N=-\infty}^{\infty} \exp\left(j \left(\sum_{n=-\infty}^{N-L} a_n \phi_n(t) + \sum_{n=N-L+1}^N a_n \phi_n(t) + \sum_{n=N+1}^{\infty} a_n \phi_n(t) \right)\right) \rho_N(t) \\ &= \sum_{N=-\infty}^{\infty} \exp\left(j \frac{\pi}{2} \sum_{n=-\infty}^{N-L} a_n\right) \prod_{i=0}^{L-1} \exp(j a_{N-i} \phi_{N-i}(t)) \rho_N(t) \\ &= \sum_{N=-\infty}^{\infty} \left(\prod_{n=-\infty}^{N-L} j a_n \right) \left(\prod_{i=0}^{L-1} \exp(j a_{N-i} \phi_{N-i}(t)) \right) \rho_N(t) \end{aligned}$$

The expansion in $\rho_N(t)$ allows us to focus on the time interval $[NT, (N + 1)T]$ so that we can ignore the future data $\{a_n \mid n > N\}$ modulated after the interval and summarizes the effect of the past data as the sum $\sum_{n=-\infty}^{N-L} a_n$. To further simplify the

expression, let

$$f_i(t) = \begin{cases} \sin(\phi_i(t)) & 0 < t - iT \leq LT \\ \cos(\phi_{i+L}(t)) & LT < t - iT \leq 2LT \\ 0 & \text{otherwise} \end{cases}$$

The exponential term becomes,

$$\begin{aligned} \exp(ja_{N-i}\phi_{N-i}(t)) \rho_N(t) &= (\cos(a_{N-i}\phi_{N-i}(t)) + j \sin(a_{N-i}\phi_{N-i}(t))) \rho_N(t) \\ &= (\cos(\phi_{(N-i-L)+L}(t)) + ja_{N-i} \sin(\phi_{N-i}(t))) \rho_N(t) \\ &= (f_{N-i-L}(t) + ja_{N-i}f_{N-i}(t)) \rho_N(t) \end{aligned}$$

Substituting back into the $S(t)$ yields,

$$\begin{aligned} S(t) &\approx \sum_{N=-\infty}^{\infty} \left(\prod_{n=-\infty}^{N-L} ja_n \right) \left(\prod_{i=0}^{L-1} (f_{N-i-L}(t) + ja_{N-i}f_{N-i}(t)) \right) \rho_N(t) \\ &= \sum_{N=-\infty}^{\infty} \left(\prod_{n=-\infty}^N ja_n \right) \rho_N(t) \left(\prod_{i=0}^{L-1} \left(f_{N-i}(t) + \frac{1}{ja_{N-i}} f_{N-i-L}(t) \right) \right) \end{aligned}$$

The product term $\prod_{i=0}^{L-1} \left(f_{N-i}(t) + \frac{1}{ja_{N-i}} f_{N-i-L}(t) \right)$ can be expanded into a series of products of f_{N-i} or f_{N-i-L} for $i \in \{0, \dots, L-1\}$. In any one of these products, f_{N-i} is present if and only if $\frac{1}{ja_{N-i}} f_{N-i-L}$ is not present. Since there are 4 possible values for i , there are a total of 2^L products, the form of which can be represented by an L -bit binary number, each bit of which indicates whether f_{N-i-L} or f_{N-i} is present. Let $\alpha_{K',i} \in \{0, 1\}$ be the $i+1$ -st least significant bit of the L -bit binary expansion of $K' \in \{0, \dots, 2^L - 1\}$. The signal becomes,

$$S(t) \approx \sum_{N=-\infty}^{\infty} \left(\prod_{n=-\infty}^N ja_n \right) \rho_N(t) \left(\sum_{K'=0}^{2^L-1} \prod_{i=0}^{L-1} \frac{1}{(ja_{N-i})^{\alpha_{K',i}}} f_{N-i-L-\alpha_{K',i}}(t) \right)$$

Some of the terms present in different time intervals $[NT, (N+1)T]$ (i.e. different values of N) have the same form. For instance, the terms with $(N, K') = (0, 0)$ and $(N, K') = (1, 1)$ have the same form except that they are non-zero at different

interval. Thus, they can be concatenated as follows,

$$\begin{aligned}
(N = 0, K' = 0) \text{ term} &= \left(\prod_{n=-\infty}^0 j a_n \right) \rho_0(t) \prod_{i=0}^{L-1} f_{-i}(t) \\
(N = 1, K' = 1) \text{ term} &= \left(\prod_{n=-\infty}^1 j a_n \right) \rho_1(t) \prod_{i=0}^{L-1} \frac{1}{(j a_{1-i})^{\alpha_{1,i}}} f_{1-i-L \cdot \alpha_{1,i}}(t) \\
&= \left(\prod_{n=-\infty}^0 j a_n \right) \rho_1(t) \prod_{i=0}^{L-1} f_{-i}(t) \\
(N = 0, K' = 0) + (N = 1, K' = 1) &= \left(\prod_{n=-\infty}^0 j a_n \right) \rho_0(t/2) \prod_{i=0}^{L-1} f_{-i}(t)
\end{aligned}$$

By concatenating all the product terms with the same form over different time interval, we have the final expression for $S(t)$,

$$\begin{aligned}
S(t) &\approx \sum_{N=-\infty}^{\infty} \left\{ \sum_{K=0}^{2^{L-1}-1} \frac{\prod_{n=-\infty}^N j a_n}{\prod_{i=1}^{L-1} (j a_{N-i})^{\alpha_{K,i}}} f_N(t) \left(\prod_{i=0}^{L-2} f_{N-1-i-L \cdot \alpha_{K,i}}(t) \right) \right. \\
&\quad \left. \times \rho_N \left(\frac{t}{2L - \max_{i \in \{0, \dots, L-2\}} (i + L \cdot \alpha_{K,i} + I_{K \neq 0})} \right) \right\} \\
&= \sum_{N=-\infty}^{\infty} \sum_{K=0}^{2^{L-1}-1} A_{K,N} C_K(t - NT)
\end{aligned}$$

where

$$\begin{aligned}
A_{K,N} &\triangleq \frac{\prod_{n=-\infty}^N j a_n}{\prod_{i=1}^{L-1} (j a_{N-i})^{\alpha_{K,i}}} \\
C_K(t - NT) &\triangleq \rho_N \left(\frac{t}{2L - \max_{i \in \{0, \dots, L-2\}} (i + L \cdot \alpha_{K,i} + I_{K \neq 0})} \right) f_N(t) \prod_{i=0}^{L-2} f_{N-1-i-L \cdot \alpha_{K,i}}(t) \\
M &\triangleq 2^{L-1} \\
I_{K \neq 0} &\triangleq \begin{cases} 0 & \text{if } K = 0 \\ 1 & \text{otherwise} \end{cases}
\end{aligned}$$

Therefore, $S(t)$ can be decomposed into a sum of time-shifted and amplitude-modulated pulses. Furthermore, it can be verified that $C_0(t)$ has the maximum pulse

width $L + 1$.

□

Remark A.2.2. Properties of the transformed symbol $A_{K,N}$

1. An undesired ramification of extending the finite data sequence to bi-infinite sequence is that $A_{K,N}$ does not converge for any N and K because the component $\prod_{n=-\infty}^N j$ in $A_{K,N}$ does not converge for any N . In reality, however, the data sequence only has 3 tail bits of 0's at the two ends (Figure 2-1) and the transmission time is finite (0.577ms), which allow us to abuse the notation $\pm\infty$ to ignore the boundary cases without worrying about the convergence issue. Note, however, that the decision to ignore the boundary cases, which corresponds to the interval when the tail bits are transmitted, is strictly suboptimal in signal detection because they convey some information about the channel. Since the tail bits are short compared to the entire normal burst, this degradation is negligible.
2. Not all sequences of $A_{K,N} \in \{1, -1, j, -j\}$ are possible because the mapping from the sequence of $\{a_n\}_{n=n_i}^{n_f}$ to $\prod_{K=0}^{M-1} \{A_{K,N}\}_{N=n_i}^{n_f}$ cannot be surjective by comparing the cardinalities of the domain and the codomain.

A.3 Least Squares Estimation

Consider a general least squares problem in matrix form,

$$\hat{\mathbf{E}} = \arg \min_{\mathbf{E}} \|\mathbf{G} - \mathbf{E}\mathbf{F}\|_{\mathbb{F}}^2 \quad (\text{A.1})$$

where \mathbf{E} is an m -by- k complex matrix of elements e_{ij} where i and j are the row and column indices; and similarly, \mathbf{F} is a k -by- n complex matrix of elements f_{ij} ; \mathbf{G} is an m -by- n complex matrix of elements g_{ij} .

Let $\hat{\mathbf{e}}_i^T$, \mathbf{e}_i^T and \mathbf{g}_i^T be the i -th rows of $\hat{\mathbf{E}}$, \mathbf{E} and \mathbf{G} respectively. Then, the squared

Frobenius norm can be expressed as a sum of squared vector 2-norms,

$$\hat{\mathbf{E}} = \arg \min_{\mathbf{E}} \sum_{i=1}^m \|\mathbf{g}_i^{\mathbf{T}} - \mathbf{e}_i^{\mathbf{T}} \mathbf{F}\|^2 \quad (\text{A.2})$$

$$\hat{\mathbf{e}}_i^{\mathbf{T}} = \arg \min_{\mathbf{e}_i^{\mathbf{T}}} \sum_{i=1}^m \|\mathbf{g}_i^{\mathbf{T}} - \mathbf{e}_i^{\mathbf{T}} \mathbf{F}\|^2 \quad (\text{A.3})$$

which means that each row of \mathbf{E} can be optimized independently from other rows. Let $\mathbf{U}\Sigma\mathbf{V}'$ be the singular value decomposition (not in reduced form; i.e. \mathbf{U} and \mathbf{V} are unitary matrices) of \mathbf{F} , where Σ is the singular value matrix of $\min(k, n)$ singular values σ_j . Then, we can further break down the optimization of $\hat{\mathbf{e}}_i^{\mathbf{T}}$ into the optimization over each transformed element y_{ij} as follows,

$$\begin{aligned} \hat{\mathbf{e}}_i^{\mathbf{T}} &= \arg \min_{\mathbf{e}_i^{\mathbf{T}}} \|\mathbf{g}_i^{\mathbf{T}} - \mathbf{e}_i^{\mathbf{T}} \mathbf{U} \Sigma \mathbf{V}'\|^2 \\ &= \arg \min_{\mathbf{e}_i^{\mathbf{T}}} \|\mathbf{g}_i^{\mathbf{T}} \mathbf{V} - \mathbf{e}_i^{\mathbf{T}} \mathbf{U} \Sigma\|^2 && \because \mathbf{V} \mathbf{V}' = \mathbf{I} \\ &= \left(\arg \min_{\mathbf{y}_i^{\mathbf{T}}} \|\mathbf{g}_i^{\mathbf{T}} \mathbf{V} - \mathbf{y}_i^{\mathbf{T}} \Sigma\|^2 \right) \mathbf{U}' && \text{where } \mathbf{y}^{\mathbf{T}} \triangleq \mathbf{e}_i^{\mathbf{T}} \mathbf{U} \\ \hat{\mathbf{y}}_i^{\mathbf{T}} &\triangleq \arg \min_{\mathbf{y}_i^{\mathbf{T}}} \sum_{j=1}^{\min(k, n)} |[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j - y_{ij} \sigma_j|^2 + \sum_{j=(k, n)+1}^n |[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j|^2 \\ \hat{y}_{ij} &= \begin{cases} \arg \min_{y_{ij}} |[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j - y_{ij} \sigma_j|^2 & j \leq \min(k, n) \\ \text{anything} & \text{otherwise} \end{cases} \end{aligned}$$

where $[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j$ denotes the j -th element in the vector $\mathbf{g}_i^{\mathbf{T}} \mathbf{V}$.

If $j \leq \min(k, n)$ and $\sigma_j \neq 0$, then $|[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j - y_{ij} \sigma_j|^2$ can be minimized to 0 by choosing $y_{ij} = \frac{[\mathbf{g}_i^{\mathbf{T}} \mathbf{V}]_j}{\sigma_j}$. If $j > \min(k, n)$ or $\sigma_j = 0$, however, choosing any y_{ij} is equally optimal as they get multiplied by zero in the optimality criterion. However, choosing $y_{ij} = 0$ in those cases will minimize the magnitude of $\mathbf{y}_i^{\mathbf{T}}$, or the 2-norm of $\mathbf{e}_i^{\mathbf{T}} \mathbf{U}$, which is also the 2-norm of $\mathbf{e}_i^{\mathbf{T}}$ since \mathbf{U} is unitary. Thus, the Frobenius norm of \mathbf{E} will be minimized, because its square is the sum of the squared 2-norm of each row $\mathbf{e}_i^{\mathbf{T}}$. In other words, we get the following unique solution by further imposing the minimum

energy requirement on \mathbf{E} ,

$$\hat{y}_{ij} = \begin{cases} 0 & \text{if } j > \min(k, n) \text{ or } \sigma_j = 0 \\ \frac{[\mathbf{g}_i^T \mathbf{V}]_j}{\sigma_j} & \text{otherwise} \end{cases}$$

$$\hat{\mathbf{y}}_i^T = \mathbf{g}_i^T \mathbf{V} \Sigma^\dagger$$

where Σ^\dagger is equal to Σ except that the corresponding non-zero entries are reciprocal of each other, and

$$\hat{\mathbf{e}}_i^T = \mathbf{g}_i^T \mathbf{V} \Sigma^\dagger \mathbf{U}'$$

Hence, we obtain the LS estimate of \mathbf{E} ,

$$\hat{\mathbf{E}} = \mathbf{G} \overbrace{(\mathbf{V} \Sigma^\dagger \mathbf{U}')}^{\mathbf{F}^\dagger \triangleq}$$

$$= \mathbf{G} \mathbf{F}^\dagger$$

where the matrix $\mathbf{V} \Sigma^\dagger \mathbf{U}'$ denoted by \mathbf{F}^\dagger is called the generalized pseudoinverse of \mathbf{F} .

A.4 Comparing CAZAC and LS Channel Estimates

The measurement model (3.6) with the training sequence TSC0 in Figure 3-5 and channel memory $v = 4$ will be used in this section to compare the CAZAC and LS channel estimates in (3.17) and (3.8) using $\text{Var}[\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}]$ and $\text{E}[\|\mathbf{X} - \hat{\mathbf{H}}\mathbf{S}\|_{\mathbb{F}}^2]$ as described in Section 3.2.1,

The training sequence TSC0, its corresponding data matrix \mathbf{S} in (3.6), the pseudoinverse \mathbf{S}^\dagger , and the matrix $\check{\mathbf{S}}$ in (3.13) are,

$$\{a_i\}_{i=-4}^{21} = 0, 0, 1, 0, 0, \underbrace{1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1}_{\text{midamble } \{\check{a}_i\}_{i=0}^{15}}$$

$$\mathbf{S} = \begin{bmatrix} 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

$$\mathbf{S}^\dagger \approx \frac{1}{21} \begin{bmatrix} 0.97 & 0.71 & -0.74 & 0.82 & 0.99 \\ -0.84 & 0.92 & 0.92 & -0.65 & 0.65 \\ 0.80 & -0.74 & 1.07 & 1.02 & -0.66 \\ -0.97 & 0.74 & -0.71 & 0.99 & 0.82 \\ -0.80 & -1.07 & 0.74 & -0.66 & 1.02 \\ -1.10 & -0.89 & -0.89 & 0.82 & -0.82 \\ 1.17 & -1.29 & -1.29 & -1.19 & 1.19 \\ 0.97 & 1.07 & -1.10 & -1.35 & -1.18 \\ 0.94 & 1.25 & 0.89 & -0.99 & -1.19 \\ 0.77 & 1.25 & 1.25 & 1.02 & -1.02 \\ -1.00 & 0.92 & 1.28 & 1.35 & 0.82 \\ 0.97 & -1.10 & 1.07 & 1.18 & 1.35 \\ 0.97 & 0.71 & -0.74 & 0.82 & 0.99 \\ 1.10 & 0.89 & 0.89 & -0.82 & 0.82 \\ -1.17 & 1.29 & 1.29 & 1.19 & -1.19 \\ 0.97 & -1.10 & 1.07 & 1.18 & 1.35 \\ 0.97 & 0.71 & -0.74 & 0.82 & 0.99 \\ -0.84 & 0.92 & 0.92 & -0.65 & 0.65 \\ 0.80 & -0.74 & 1.07 & 1.02 & -0.66 \\ -0.97 & 0.74 & -0.71 & 0.99 & 0.82 \\ -0.80 & -1.07 & 0.74 & -0.66 & 1.02 \\ -1.10 & -0.89 & -0.89 & 0.82 & -0.82 \end{bmatrix}, \quad \check{\mathbf{S}} = \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 \\ -1 & -1 & 1 & -1 & 0 \\ -1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 0 & 1 & 1 & -1 & 1 \\ 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note that each column of \mathbf{S}^\dagger and $\check{\mathbf{S}}$ contains 21 and 16 non-zero entries respectively, the magnitudes of which are similar and sum to 1 exactly, and the signs of which are the same for the same pair of row and column indices. Thus, the main difference between CAZAC estimate and LS estimate is that the CAZAC estimate does not fully utilize the observation sequence in the estimate of each tap. More precisely, the zero entries of the i -th column of $\check{\mathbf{S}}$ corresponds to those observations that are not used in computing the estimate of the i -th channel tap. The degradation resulting from this can be summarized by increase in the distance $\mathbb{E}[\|\mathbf{X} - \hat{\mathbf{H}}\mathbf{S}\|_{\mathbb{F}}^2]$,

$$\begin{aligned} \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{H}}_{\text{LS}}\mathbf{S}\|_{\mathbb{F}}^2] &= \mathbb{E}[\|\mathbf{H}\mathbf{S} + \mathbf{N} - (\mathbf{H}\mathbf{S} + \mathbf{N})\mathbf{S}^\dagger\mathbf{S}\|_{\mathbb{F}}^2] \\ &= \sigma_n^2 \text{trace}(\mathbf{I} - \mathbf{S}^\dagger\mathbf{S}) \\ &= 17\sigma_n^2 \\ \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{H}}_{\text{CAZAC}}\mathbf{S}\|_{\mathbb{F}}^2] &= \sigma_n^2 \text{trace}((\mathbf{I} - \check{\mathbf{S}}\mathbf{S})'(\mathbf{I} - \check{\mathbf{S}}\mathbf{S})) \\ &\approx 18.7\sigma_n^2 > \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{H}}_{\text{LS}}\mathbf{S}\|_{\mathbb{F}}^2] \end{aligned}$$

We can also compare the difference between the two estimates more directly with the covariance matrix $\text{Var}[\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}]$ because the difference $\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}$ is indeed the colored zero-mean jointly Gaussian noise $\mathbf{N}(\check{\mathbf{S}} - \mathbf{S}^\dagger)$, which can be summarized

by its second-order statistics,

$$\text{Var}[\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}] \approx \sigma_n^2 \begin{bmatrix} 0.0147 & 0.0001 & 0.0001 & -0.0048 & 0.0116 \\ 0.0001 & 0.0147 & 0.0048 & -0.0001 & -0.0001 \\ 0.0001 & 0.0048 & 0.0151 & 0.0004 & 0.0008 \\ -0.0048 & -0.0001 & 0.0004 & 0.0151 & -0.0047 \\ 0.0116 & -0.0001 & 0.0008 & -0.0047 & 0.0163 \end{bmatrix}$$

from which the normalized energy different between the channel estimates can be calculated in terms of the signal-to-noise ratio $\text{SNR} \triangleq \frac{\sigma_n^2}{\text{E}[\|\mathbf{h}\|^2]}$,

$$\begin{aligned} \frac{\text{E}[\|\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}\|^2]}{\text{E}[\|\mathbf{h}\|^2]} &= \frac{\text{trace}(\text{Var}[\hat{\mathbf{h}}_{\text{CAZAC}} - \hat{\mathbf{h}}_{\text{LS}}])}{\text{E}[\|\mathbf{h}\|^2]} \\ &\approx \frac{0.076}{\text{SNR}} \end{aligned}$$

A.5 Different Model Assumptions on the Single Interferer Model

The single interferer model described in Section 4.2 can further take on three different model assumptions described as blind, Semi-blind and training Sequence code Based. In this section, we will describe the differences between this model assumptions and quote some examples of SAIC developed based on each model assumption.

The training sequence code based SAIC algorithms assume that the receiver knows that training sequence code for both the desired carrier and the interferer so that their channel impulse responses \mathbf{H} and \mathbf{C} can both be tracked using techniques described in Section 3.2. The well-known Joint Maximum Likelihood Sequence Estimation (JMLSE) and the Joint Maximum A posteriori Probability Signal Detection (JMAPSD)[6] fall under this category. The idea is immediately obvious if we stack up the interferer and the desired signal in (3.5) as follows,

$$\mathbf{X} = [\mathbf{H} \ \mathbf{C}] \begin{bmatrix} \mathbf{s} \\ \mathbf{z} \end{bmatrix} + \mathbf{N}$$

By incorporating the interferer as part of the signal, this reorganized matrix form has the same structure as the interference-free model in (3.5). Thus, the joint channel-

data estimation technique in Section 3.2 can be applied here to estimate the joint channel $[\mathbf{h} \ \mathbf{c}]$ and demodulate the joint signal and interferer $[\mathbf{s} \ \mathbf{z}]^T$. The technique is called the joint demodulation. It has great performance because the training sequence code based model assumption is the strongest among the three model assumptions and the joint demodulation can fully exploit the fact that the interferer is a GSM signal by actually demodulating it. The assumption is, however, unrealistic and the demodulator is very complex computationally because the running time of the Viterbi decoder used is exponential in the channel memory length, which is roughly doubled in the joint channel $[\mathbf{h} \ \mathbf{c}]$.

The blind model assumption is the opposite extreme of the training sequence code based model assumption. training sequence code is not assumed for the desired signal nor the interferer. Thus, adaptive filtering technique is usually used to continuously track the channel, or it would be impossible to get a good channel estimate that converges. An example is the blind JMAPSD algorithm[7]. It assumes an all-pole channel model to reduce the search space and then uses the Kalman filter to adaptively update the model from the received signal. Its performance depends partly on how well the channel model reflects reality, and how fast the channel estimate converges.

The most realistic model assumption for the asynchronous GSM network is the semi-blind model, which is in between the training sequence code based assumption and the blind assumption. More precisely, training sequence code is assumed only for the desired signal but not the interferer because the desired signal is assumed to be well tracked in time, while the interferer can have a random misalignment that causes the unknown data sequences of the interferer to overlap in time with the training sequence code of the signal. An example is the Iterative-semi-blind JMLSE algorithm[13], which iteratively improves the channel estimate using the least mean square adaptive equalizer. The proposed decorrelator design in this project also assumes the semi-blind model. However, rather than to jointly demodulate the signal and the interference, it attempts to eliminate the component in the observation sequence corrupted by the interference so that the conventional receiver structure can be reused.

Bibliography

- [1] Cingular, Erisson, Infineon, Intel, Motorola, Nokia, Nortel, Philips, Qualcomm, Siemens, Texas Instruments, and Vodafone. CR 45.005-092 rev 2, introduction of DARP performance requirements (rel-6). Change request GP-042829, 3GPP GERAN, November 2004.
- [2] ETSI. Digital cellular telecommunications system (phase 2+); channel coding (GSM 05.03 version 8.5.1 release 1999). Technical Report ETSI EN 300 909 V8.5.1, ETSI, November 2000.
- [3] ETSI. Digital cellular telecommunications system (phase 2+); multiplexing and multiple access on the radio path (GSM 05.02 version 8.5.1 release 1999). Technical Report ETSI EN 300 908 V8.5.1, ETSI, November 2000.
- [4] ETSI. Digital cellular telecommunications system (phase 2+); radio transmission and reception (GSM 05.05 version 8.4.1 release 1999). Technical Report ETSI EN 300 910 V8.4.1, ETSI, October 2000.
- [5] G. D. Forney. Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference. *IEEE Transactions on Information Theory*, IT-18(3):363–378, may 1972.
- [6] K. Giridhar, Sujai Chari, John J. Shynk, Richard P. Gooch, and Douglas J. Artman. Joint estimation algorithms for cochannel signal demodulation. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 3:1497–1501, may 1993.

- [7] K. Giridhar, John J. Shynk, Amit Mathur, Sujai Chari, and Richard P. Gooch. Nonlinear techniques for the joint estimation of cochannel signals. *IEEE Transactions on communications*, 45(4):473–484, apr 1997.
- [8] k. Murota and K. Hirade. GMSK modulation for digital mobile radio telephony. *IEEE Transactions on Communications*, COM-29(7):1044–1050, jul 1981.
- [9] Pierre A. Laurent. Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP). *IEEE Transactions on Communications*, COM-34(2):150–160, feb 1986.
- [10] Jen-Wei Liang and Arogyaswami J. Paulraj. Two stage CCI/ISI reduction with space-time processing in TDMA cellular networks. *Signals, Systems and Computers*, 1:607–611, nov 1996.
- [11] S. Ono, H. Hayashi, T. Tomoko, and K. Noriaki. A MLSE receiver for the GSM digital cellular system. In *Proc. of the 44th IEEE Vehicular Technology Conference*, pages 230–233, jun 1994.
- [12] Rapporteur. SAIC/DARP status (rev 2). Technical Report GP-042888, 3GPP GERAN, November 2004.
- [13] Hendrik Schoeneich and Peter A. Hoeher. Single antenna interference cancellation: iterative semi-blind algorithm and performance bound for joint maximum-likelihood interference cancellation. *GLOBECOM*, pages 1716–1720, 2003.
- [14] Jan J. Tuma. *Handbook of numerical calculations in engineering*. McGraw-Hill Publishing Company, 1989.