

Receiver-based Algorithms for Bidirectional Nonlinear Satcom Channels

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

In Partial Fulfillment

of the requirements of the Degree

Master of Science (Electrical Engineering)


by

Muhammad E Qureshi

May 2014

APPROVAL SHEET

The thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Muhammad E Qureshi 
AUTHOR

The thesis has been read and approved by the examining committee:

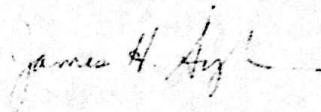
Stephen Wilson

Advisor

Toby Berger

Zongli Lin

Accepted for the School of Engineering and Applied Science:



Dean, School of Engineering and Applied Science

May
2014

Acknowledgements

First and foremost, I thank God for all his blessings and favors.

Next, I would like to express my deepest gratitude to my advisor and mentor, Professor Stephen Wilson, for his continuous support, guidance, patience, and leadership; and for extending this amazing opportunity to undertake graduate studies as his student. I would also like to thank him for his dedication, commitment, and the countless times he helped me deal with the obstacles that came along the road.

I would like to thank Professors Toby Berger and Zongli Lin for serving on my defense committee and also for the inspiration and knowledge they imparted in their classrooms.

I would like to thank my friends, who shared pleasant times with me and supported me through the tough times.

I would also like to express my gratitude to National Science Foundation (NSF) for making this project possible by providing financial support under grant number 1116997.

Finally, I would like to thank my family, especially my loving parents M.Shoaib and Zohra for their constant support, love, and affection, without which this would all be impossible.

Thank you all!

Abstract

Bidirectional satellite communication between earth terminals on the same frequency at the same time is an emerging means of doubling spectral efficiency on satellite channels. With a linear satellite repeater, cancellation of self-interference can be done with standard echo cancelling methods. However, satellite amplifiers are normally operated in a highly-nonlinear regime, for maximum power output, and this complicates the interference removal process. Our research studies two algorithms for ‘inverting’ the nonlinearity at the receiving terminal, making the resulting channel roughly linear, so echo cancellation again becomes attractive. These methods are 1) a MAP detection approach which selects the most-probable choice among candidate input sequences, given the observed amplifier output; and 2) a particle filtering approach which approximates the MMSE estimate of the input signal, given the output sequence. These approaches are compared at different SNR’s and different amplifier backoff’s, and it is concluded that the particle filtering approach is more robust in the presence of downlink noise, while also offering reasonable receiver complexity.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 7 |
| 1.1 | Thesis Organization | 9 |
| 1.2 | Other Candidate Approaches | 9 |
| 2 | Problem Description | 11 |
| 2.1 | Transmitter | 11 |
| 2.1.1 | 16-APSK | 12 |
| 2.1.2 | Pulse Shaping Filter | 12 |
| 2.1.3 | Windowing Function | 14 |
| 2.2 | Relay | 15 |
| 2.3 | Receiver | 17 |
| 2.4 | Adaptive Echo Cancellation | 17 |
| 2.5 | Centroid-based Decision | 19 |
| 3 | MAP Detection Algorithm | 23 |
| 3.1 | Algorithm | 23 |
| 3.1.1 | Inversion of Saleh model | 25 |
| 3.1.2 | MAP sequence selection | 27 |
| 3.1.3 | MAP root selection procedure | 29 |
| 3.1.4 | Matched filtering and down-sampling and adaptive cancellation | 31 |
| 3.2 | Performance Evaluation | 32 |
| 3.2.1 | Single user | 32 |
| 3.2.1.1 | Effect of SNR | 33 |
| 3.2.1.2 | Effect of backoff | 34 |
| 3.2.2 | Bidirectional case | 34 |

| | | |
|----------|---|-----------|
| 3.2.2.1 | Effect of SNR | 35 |
| 3.2.2.2 | Effect of backoff | 37 |
| 3.2.2.3 | Constellation quality and error rate curves | 39 |
| 3.3 | Summary | 41 |
| 4 | Particle Filtering | 42 |
| 4.1 | Overview | 42 |
| 4.2 | State Dynamics Modeling for the Particle Filter | 43 |
| 4.3 | Particle Filtering Algorithm | 46 |
| 4.4 | Linear Channel Testing | 50 |
| 4.4.1 | PAM | 51 |
| 4.4.1.1 | The Effect of SNR | 51 |
| 4.4.1.2 | The Number of Particles | 53 |
| 4.4.1.3 | Effect of Higher-Order State Models | 54 |
| 4.4.2 | QPSK | 55 |
| 4.4.2.1 | The Effect of SNR | 56 |
| 4.4.2.2 | The Number of Particles | 58 |
| 4.4.3 | 16-APSK | 59 |
| 4.5 | Nonlinear Channel | 63 |
| 4.5.1 | Effect of backoff | 63 |
| 4.5.2 | Effect of SNR | 67 |
| 4.5.3 | Effect of number of particles | 68 |
| 4.5.4 | Bidirectional transmission | 69 |
| 4.5.5 | Effect of SNR | 70 |
| 4.5.6 | Effect of number of particles | 71 |
| 4.5.7 | Effect of backoff | 72 |
| 4.6 | Performance Evaluation | 76 |
| 5 | Summary and Future Work | 78 |
| 5.1 | Summary of Results | 78 |
| 5.2 | Future work | 81 |
| 5.2.1 | MAP detector | 81 |
| 5.2.2 | Particle filtering | 81 |

| | | |
|----------|--|-----------|
| A | Algorithm For Reduction to Strictly-Proper Form | 83 |
| B | Obtaining Observer-Canonical Form | 85 |

List of Figures

| | | |
|------|--|----|
| 1.1 | Two terminals and a satellite relay [29] | 8 |
| 2.1 | Block diagram for the transmitters | 12 |
| 2.2 | 16-APSK constellation | 13 |
| 2.3 | Time domain representation of the FIR pulse shaping filter | 13 |
| 2.4 | Frequency response of the pulse shaping filter | 14 |
| 2.5 | Depiction of tapered sine window coefficients | 15 |
| 2.6 | Satellite relay's block diagram | 15 |
| 2.7 | Plots of Saleh model AM-AM and AM-PM characteristics | 16 |
| 2.8 | Receiver block diagram | 18 |
| 2.9 | Ideal inversion leads to an overall linear channel with gain of 1 | 18 |
| 2.10 | Structure of adaptive echo canceller | 19 |
| 2.11 | Adaptive filter structure | 19 |
| 2.12 | Voronoi tessellation of the complex plane | 20 |
| 2.13 | Received constellation and the corresponding centroids | 21 |
| 2.14 | Voronoi tessellation of the complex plane | 21 |
| 3.1 | Flowchart of MAP inverter's algorithm | 24 |
| 3.2 | Identification of various zones in the Saleh AM-AM characteristic | 27 |
| 3.3 | Depiction of an ambiguous sequence | 28 |
| 3.4 | Result of above convolution provides value of required correlation between samples | 30 |
| 3.5 | The ideal sum of two (synchronized) 16-APSK constellations | 32 |
| 3.6 | MSE vs E_{Sat}/N_o for single user case | 33 |
| 3.7 | MSE vs IBO with SNR =35 dB for single user case | 34 |
| 3.8 | MSE vs E_{Sat}/N_o with IBO = -5 dB for bidirectional case | 35 |
| 3.9 | Locus tracking of a sub sequence using MAP inverter at various SNRs | 36 |

| | | |
|------|---|----|
| 3.10 | Standard tracking error between the true and the estimated signal, SNR = 30 dB, backoff = -5 dB | 36 |
| 3.11 | MSE vs IBO with SNR = 35 dB for bidirectional case | 38 |
| 3.12 | MSE vs SNR at various backoff levels, bidirectional case | 38 |
| 3.13 | Recovered constellation for SNRs 25(left) and 30(right) dB with IBO =-5 dB | 39 |
| 3.14 | Recovered constellation for SNRs 35(left) and 40(right) dB with IBO =-5 dB | 40 |
| 3.15 | SER plots for MAP inverter | 40 |
| 4.1 | Pole-zero plots for original Butterworth filter and strictly-proper approximation | 45 |
| 4.2 | Frequency response plots for original Butterworth filter and strictly-proper approximation | 45 |
| 4.3 | Flowchart of particle filter based inverter algorithm | 48 |
| 4.4 | Transmitter set up for linearized channel | 50 |
| 4.5 | Receiver's Block Diagram | 51 |
| 4.6 | Tracking results with various SNR's with 100 particles. | 52 |
| 4.7 | MSE vs. SNR(dB) | 52 |
| 4.8 | Tracking results with varying number of particles at an SNR of 15 dB. | 53 |
| 4.9 | MSE vs. number of particles | 54 |
| 4.10 | MSE vs. N_p for 2^{nd} and 4^{th} -order state models | 55 |
| 4.11 | Tracking results for QPSK for SNRs 15 dB (left) and 20 dB (right), $N_p=100$, Bidirectional case | 56 |
| 4.12 | Tracking results for QPSK for SNRs 28 dB (left) and 35 dB (right), $N_p=100$, Bidirectional case | 57 |
| 4.13 | MSE vs SNR for Single-user (left) and Bidirectional (right) cases with $N_p = 100$ | 57 |
| 4.14 | Constellations for QPSK for SNRs 15 dB (left) and 20 dB (right), $N_p=100$, Bidirectional case | 58 |
| 4.15 | Constellations for QPSK for SNRs 28 dB (left) and 35 dB (right), $N_p=100$, Bidirectional case | 58 |
| 4.16 | MSE vs number of particles at SNR= 30 dB, Bidirectional case | 59 |
| 4.17 | Tracking results for 16-APSK, for SNRs 15 dB (left) and 20 dB (right), $N_p = 100$, Bidirectional case | 60 |
| 4.18 | Tracking results for 16-APSK, for SNRs 28 dB (left) and 35 dB (right), $N_p = 100$, Bidirectional case | 60 |
| 4.19 | MSE vs SNR for 100 particles, 16-APSK, linear channel | 61 |
| 4.20 | Constellations for 16-APSK, for SNRs 15 dB (left) and 20 dB (right), $N_p = 100$, Bidirectional case | 61 |
| 4.21 | Constellations for 16-APSK, for SNRs 28 dB (left) and 35 dB (right), $N_p = 100$, Bidirectional case | 62 |

| | | |
|------|---|----|
| 4.22 | MSE vs N_p , with SNR = 30 dB, Bidirectional case | 62 |
| 4.23 | MSE vs. IBO with $N_p = 100$ and SNR = 35 dB, Single-user case | 64 |
| 4.24 | Constellation plots for SNR = 35 dB, $N_p=100$ IBO = -12 dB (top-left), -10 dB (top-right), -8 dB (bottom-left), -7 dB (bottom-right) | 65 |
| 4.25 | Constellation plots for SNR = 35 dB, $N_p=100$ IBO = -6 dB (top-left), -5 dB (top-right), -4 dB (bottom-left), -3 dB (bottom-right) | 66 |
| 4.26 | P_s vs. IBO with $N_p = 100$ and SNR = 35 dB, Single-user case. | 66 |
| 4.27 | MSE vs. SNR for 16-APSK, IBO = -4 dB, $N_p = 100$ | 67 |
| 4.28 | MSE vs. N_p for 16-APSK, with SNR= 30 dB, IBO= -5 dB, Single-user case | 68 |
| 4.29 | P_s vs. N_p for 16-APSK, with SNR =30 dB, IBO =-5 dB, Single-user case | 69 |
| 4.30 | Tracking for varying SNR at IBO= -6dB and $N_p = 500$ | 70 |
| 4.31 | MSE vs SNR, IBO= -6 dB and $N_p = 500$ | 71 |
| 4.32 | MSE vs. N_p for 16-APSK, with SNR= 30 dB, IBO= -5 dB, Bidirectional case | 71 |
| 4.33 | P_s vs. N_p for 16-APSK, with SNR = 30 dB, IBO = -5 dB, Bidirectional case | 72 |
| 4.34 | Constellation plots for SNR= 35 dB, $N_p = 100$ IBO= -12 dB (top-left), -10 dB (top-right), -8 dB (bottom-left), -7 dB (bottom-right) | 73 |
| 4.35 | Constellation plots for SNR= 35 dB, $N_p = 100$ IBO = -6 dB (top-left), -5 dB (top-right), -4 dB (bottom-left), -3 dB (bottom-right) | 74 |
| 4.36 | P_s vs. IBO with $N_p = 200$ and SNR = 35 dB, Bidirectional case | 75 |
| 4.37 | MSE vs. IBO with $N_p = 200$ and SNR= 35 dB, Bidirectional case | 75 |
| 4.38 | Tracking MSE of particle filter | 76 |
| 4.39 | P_s vs SNR for particle filter | 77 |
| 5.1 | Comparison of tracking MSE of MAP inverter (left) and particle filter (right) | 79 |
| 5.2 | Comparison of SER plots for MAP inverter(left) and particle filter(right) | 80 |
| 5.3 | Computational time comparison of MAP inverter(left) and particle filter(right) | 81 |

Chapter 1

Introduction

This thesis deals with a specific problem in bidirectional (same frequency) satellite communication, wherein two uplink signals transmit data on the same carrier frequency. A satellite relay amplifies the sum signal (without demodulation and reencoding) and rebroadcasts to both communicating terminals. With this method, we achieve up to double the spectral efficiency in bits/second/Hz that a traditional two-way system achieves, either by use of non-overlapping frequencies or by time-sharing. Figure 1 shows a simple depiction of the operational setup.

In the information theory literature this constitutes an instance of the two-way relay channel, formulated originally by Shannon. Our choice for relay processing, however, is not the most general relay, but merely a traditional amplify-and-forward relay.

For linear relaying amplifiers, it is known that amplify-forward relaying can perform relatively close to the cut-set bound on the capacity region for the two-way relay channel, and this is beneficial because the relay does not require sophisticated on-board processing. Amplify-forward processing enjoys complete flexibility to uplink modulation/coding choice, and does not require user synchronization at the satellite. What makes the protocol work in essence is that the downlink signal contains self-interference (or echo), which each terminal knows as side-information, since the given terminal initiated the uplink it receives after delay. Consequently, standard echo cancellation techniques [14], are feasible for the linear relay channel.

Satellite amplifiers (often called transponders) are nonlinear, as is any amplifier ultimately. In satellite communication, it's desired to operate near the maximum power operating point where the strongest downlink is obtained, and where the DC-to-RF efficiency is highest. The nonlinearity is most simply modeled as a memoryless nonlinearity imposing AM-to-AM and AM-to-PM distortion to the input RF signal. In this

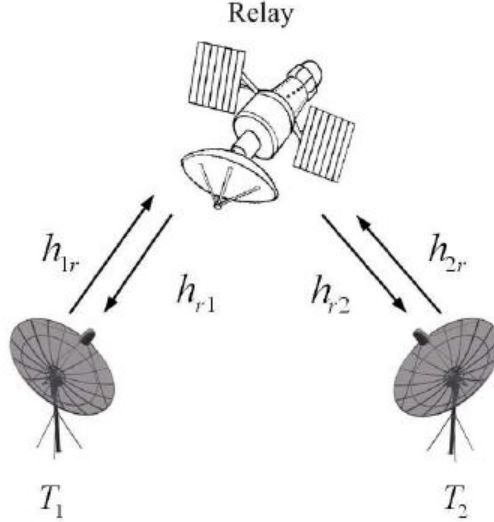


Figure 1.1: Two terminals and a satellite relay [29]

case each receiver terminal receives a nonlinear function of the sum of two uplink signals, and elimination or cancellation of the echo is no longer straight-forward as in the linear amplifier case. **Therein lies the origin of the central problem of this thesis—how can we simply process the received signal to extract the signal of interest, given the available side-information?**

We will study two approaches, both rather heuristically-motivated. They have advantages of being independent of constellation size, relative delay and carrier phase/frequency difference between the two uplink signals, and can even tolerate differing data rates, a common situation in real applications. Common to both methods is the aim of recovering a close approximation to the amplifier input waveform (sum of two RF signals), and if this is roughly successful we have ‘inverted’ the nonlinearity, whence standard echo cancellation becomes feasible. The methods in brief terms are

1. a MAP detection algorithm operating on the received signal for reconstructing the most-likely waveform at the amplifier input. Based on knowledge of the amplifier characteristic and on a statistical model for the received continuous-time signal, we do MAP detection on ambiguous slices of the signal (where the nonlinearity is non-invertible). This is detailed and analyzed in Chapter 3.
2. waveform reconstruction using a Bayesian framework based on particle filtering, by trying to construct the a posteriori p.d.f for the input signal, given the noisy downlink measurement. From this we output the MMSE estimate, namely the mean of the posterior random variable. This method is discussed in Chapter 4.

Results to date indicate both methods are successful at achieving the objective, though the particle filtering method seems more robust to downlink noise in the low SNR¹ range as demonstrated later. Detailed complexity studies need to be made to assess the practical feasibility for real-time processing at data rates of tens of MHz. A more extensive set of operational conditions should ultimately be tested, including asynchronous uplink operation, sensitivity to amplifier modeling error, etc.

1.1 Thesis Organization

In Chapter 2, detailed descriptions of the transmitter, relay, and receiver processing is given, establishing notation. Of particular importance here is the Saleh model adopted for our research, a common model for satellite amplifiers that provides a memoryless saturating nonlinearity.

Chapter 3 is devoted to the MAP detection approach, wherein ambiguous (non-uniquely-invertible) intervals in the data are identified, and candidate sample sequences are formulated, then evaluated according to *a posteriori* probability over the ambiguous spans.

Chapter 4 presents our particle filtering approach, which is a Monte Carlo approach in Bayesian methods for estimating the posterior probability density given a state model which is observed nonlinearly with additive noise. The chapter reviews the particle filter methodology and tests the algorithm in conjunction with echo cancellation for 16-APSK transmission of two signals.

Finally Chapter 5 provides some conclusions drawn from the research, and identifies topics for further study.

1.2 Other Candidate Approaches

The end-to-end model from modulator inputs to the sampled matched filter output at one receiver can be viewed as a two-input, one-output Volterra model [26] expressing both nonlinearity and memory. To the output of this model would be added complex Gaussian white discrete-time noise. If the parameters of a system were exactly known, including relative symbol timing on the uplink as well as relative carrier phase rotation and input backoff, then the Volterra model can be derived and truncated to some finite memory order. From this, the optimal sequence decoder for one of the sequences would be a Viterbi-style trellis decoder with number of states equaling M^L , where M is the constellation size and L is the memory order of the Volterra system. Side information about one sequence is incorporated into the branch labels of this trellis.

¹Through out this document we use SNR and E_{sat}/N_o interchangeably in the context of nonlinear channel.

Though formally optimal, we do not see this as practical on two counts. First, decoder complexity is typically infeasible for alphabets as large as 16, say. Second, it seems impractical to keep learning and updating a Volterra model as the signal synchronization changes slowly with time.

Predistortion, a *transmitter-based* pre-compensation for nonlinearity, has received considerable attention in wireless research recently, but this can only be done in this application on-board the satellite at the waveform level, and this is not compatible with legacy satellites. (Separate predistortion at each of the uplink terminals is unable to correctly compensate for nonlinearity.) Moreover, predistortion cannot ultimately linearize the system beyond the saturation point of the amplifier, still leaving a peak-limited amplifier.

Nonlinear receiver-based equalizers have been studied [4] for single-user reception over nonlinear channels, but it's unclear how the equalizer methodology can incorporate side-information in the two-user problem.

Thus, we believe the most sensible approach to signal processing for the two-signal problem is a pragmatic one: to try to restore a quasi-linear channel by inferring the sum signal at the input to the nonlinear amplifier, then resorting to echo cancellation methods appropriate for linear channel bidirectional relaying. Though not optimal in any sense, we believe it presents a practical alternative, and in particular does not depend on any assumptions about constellation, relative symbol timing at the satellite, or carrier phase rotation which will always be present. Even unequal data rates from the two terminals fits within our framework. Finally, the techniques presented here are transparent to whether coding is applied on either signal, or to the details of the coding.

In the remainder of this thesis, we describe two approaches for this signal inference problem.

Chapter 2

Problem Description

In this Chapter we discuss our problem setup. The satellite acts as a relay between two ground terminals that wish to exchange data, simultaneously on the same frequency. We denote the two ground terminals by T_1 and T_2 , and the satellite relay by R as shown earlier in Figure 1.1. These terminals cannot communicate directly, but only through a satellite relay that forms the link between the two terminals. To accomodate duplexing at the satellite and earth terminals, we assume a traditional frequency plan—uplinks are in one frequency band, say 14 GHz, while downlinks are in another band, say 12 GHz. Thus all terminals are transmitting and receiving at the same time. A final assumption is that T_1 and T_2 are in the footprint of the satellite antenna beam.

We next break down the system into transmitter, relay, and receiver blocks and explain the structure and functionality of each block.

2.1 Transmitter

Both the terminals operate on the same principle. We start defining the problem at the transmitting terminal T_1 , with reference to Figure 2.1.

The user at T_1 wants to send a stream of data d_1 which is drawn from a constellation, \mathcal{C} . This constellation is 16-ary amplitude phase shift keying (APSK) in most of our research, but it can be any QAM constellation. In order to simulate continuous-time processing we upsample this sequence by eight, producing d_{1up} . This sequence is then passed through a pulse shaping filter whose impulse response is h_1 . The pulse shaping filter used is a finite impulse response (FIR) root-raised-cosine (RRC) filter. The pulse shaping filter creates a complex, bandlimited signal s_1 which is transmitted to the satellite relay R . The transmitter at T_2 works on the same principle and aims to transmit the sequence d_2 . d_{2up} is generated by upsampling d_2 . d_{2up} is

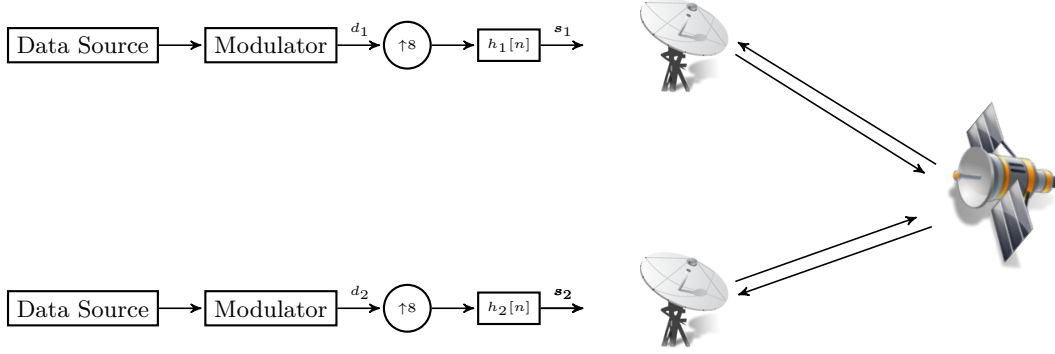


Figure 2.1: Block diagram for the transmitters

then fed to pulse shaping filter $h_2[n]$ and the resulting waveform s_2 is transmitted to R . The structure of the transmitter is shown in Figure 2.1 below.

2.1.1 16-APSK

In most of our study, we transmit data drawn from 16-APSK. This is a choice with high bandwidth efficiency in bps/Hz, and is part of the DVB-S2 standard for satellite relaying. 16-APSK was initially adopted because of its relatively low Peak-to-Average-Power Ratio (PAPR). This attribute is important in communications on nonlinear channels [12] as the low PAPR means that the peaks are relatively closer to the average power and therefore the overall nonlinear distortion induced is relatively small [8]. The constellation consists of an inner ring and an outer ring. The symbols belonging to the inner ring form a typical 4-QAM constellation. The outer ring has 12 symbols where the symbols represent a 12-ary phase shift keying (PSK) constellation. There are different standards that dictate the ratio between the outer ring radius and the inner ring radius. In our study the ratio is 2.57. Below we give a mathematical and graphical description of the constellation.

$$d_{R_{in}} = a \exp j\left(\frac{\pi}{4} + i_{in} \frac{\pi}{2}\right), \quad i_{in} \in \{0, 1, 2, 3\} \quad (2.1)$$

$$d_{R_{out}} = 2.57a \exp j\left(\frac{\pi}{12} + i_{out} \frac{\pi}{2}\right), \quad i_{out} \in \{0, 1, \dots, 11\} \quad (2.2)$$

where a is a scale factor. This set is shown below when a is chosen such that the average constellation energy is 1. Notice the relatively uniform spacing of points within a circle constraint.

2.1.2 Pulse Shaping Filter

The pulse shaping filter used in our work is an FIR filter based on the RRC filter as alluded to earlier. The filter has a length of 65 and hence total length of 8 symbols. The figures below show the impulse response and the frequency response of the filter. The cutoff frequency of the filter is $\frac{1}{8}$, a consequence of

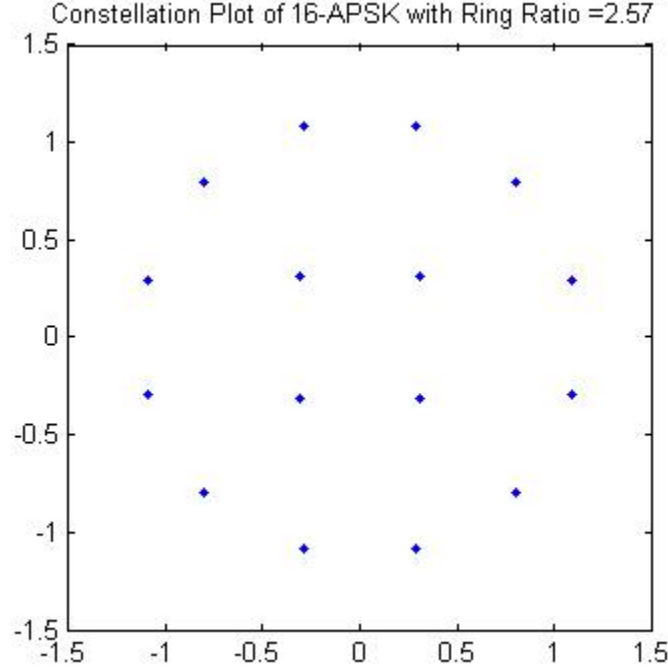


Figure 2.2: 16-APSK constellation

oversampling by 8. Furthermore, a roll-off factor, $\beta = 0.25$ is chosen as a practical design choice. As can be seen from the figure below, the filter coefficients are non-zero at the tails and this truncation can lead to small ‘discontinuities’ at the filter’s output. We found this adversely impacted our MAP reconstruction method. This is because the MAP inverter relies on the smoothness of the locus and the presence of small ‘discontinuities’ when coupled with additive noise can be detrimental the reconstruction process. To appease this issue we added a time-domain window to taper the filter coefficients to zero.

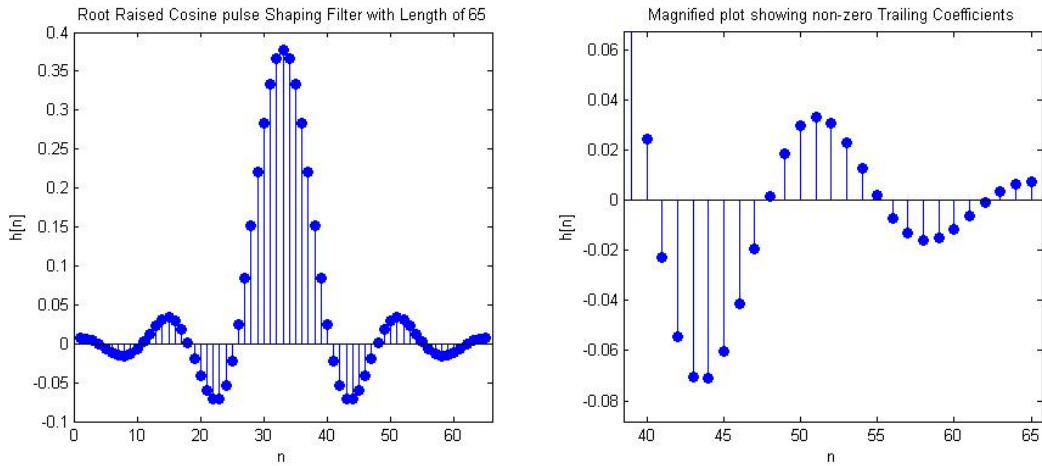


Figure 2.3: Time domain representation of the FIR pulse shaping filter

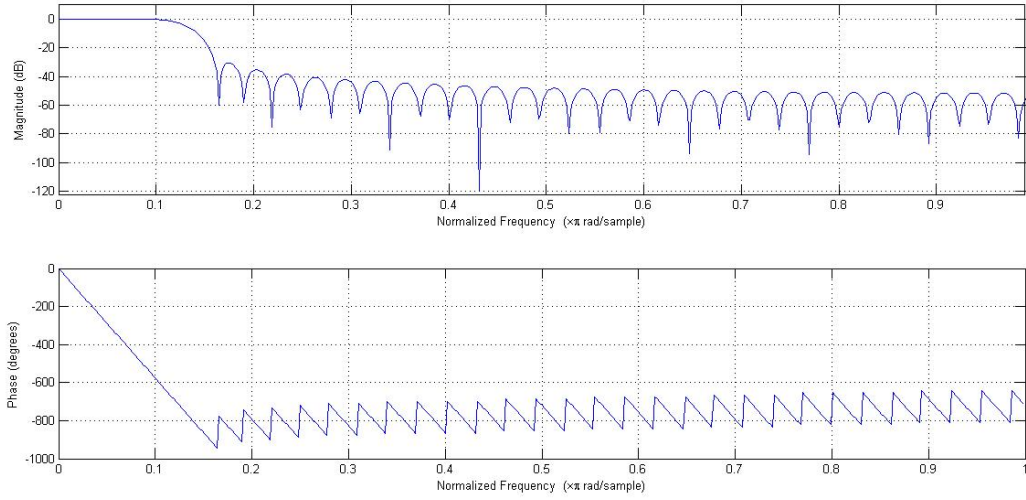


Figure 2.4: Frequency response of the pulse shaping filter

2.1.3 Windowing Function

This problem of discontinuous tail coefficients is rectified by using a window. We tested various windows during the implementation of the algorithm. Windowing does reduce the non-smooth behavior of the RRC filter. However, a negative consequence is the introduction of intersymbol interference (ISI) as the Nyquist property is no longer obtained.¹ For this reason we designed a tapered sine window which is shown below. The window is concatenation of a rectangular window with tapered sine function at each end.

$$f(x) = \begin{cases} \sin\left(\frac{(w_l-n)\pi}{2(l_1-1)}\right) & 1 \leq n \leq l_1 \\ 1 & l_1 < n < w_l - l_1 \\ \sin\left(\frac{(n-w_l)\pi}{2(l_1-1)}\right) & w_l - l_1 \leq n \leq w_l \end{cases}$$

In the above equation, w_l corresponds to the length of the window, and the variable l_1 refers to the length of the tapered sine function at either end of the rectangular window. The time domain representation of the window is shown below. Simulations suggested that the window did remove the discontinuity effects introduced by the original pulse shaping filter and the consequential ISI introduced was not severe, as will be seen later.

¹The original filter has some small amount of ISI because of filter truncation.

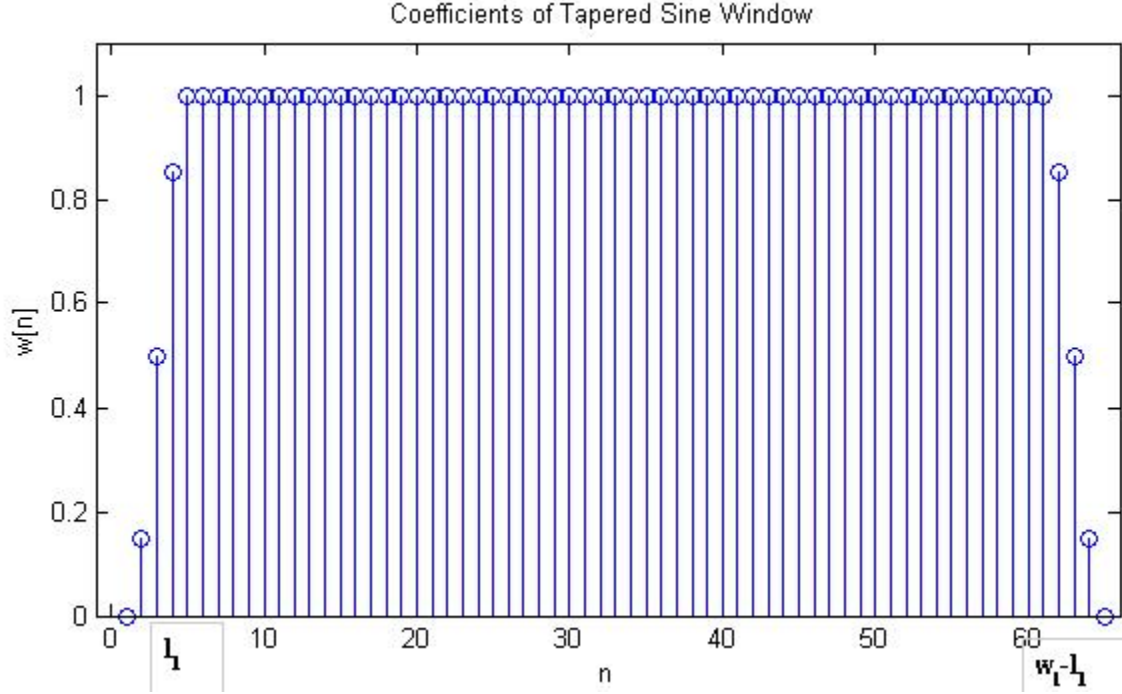


Figure 2.5: Depiction of tapered sine window coefficients

2.2 Relay

At the relay the two transmitted signals s_1 and s_2 arrive from T_1 and T_2 respectively. The two signals are added together and the aggregate signal is

$$S = s_1 + s_2 \quad (2.3)$$

The relay works on the principle of amplify and forward and thus amplifies the aggregate signal S as shown in Figure 2.6. The aggregate signal is then fed to the amplifier represented as $g(\cdot)$ in Figure 2.6.

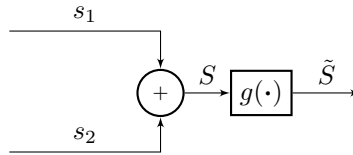


Figure 2.6: Satellite relay's block diagram

The uplink signals have power control to adjust the aggregate power level at the amplifier input. The input backoff (IBO) is normally expressed in dB, and determines how close to saturation the amplifier is operated. At small backoffs, typically $IBO \geq -3$ dB, the input signal is driven harder into the saturation

zone and the input-output characteristic is highly nonlinear, whereas at larger backoffs, say $IBO \leq -6$ dB, the amplifier demonstrates a nearly linear input-output characteristic and the signal seldom traverses into saturation. The input backoff IBO is related to the base 10 scale factor ρ via the following relation.

$$\rho = 10^{\frac{IBO}{10}}$$

(Each uplink would need power control to manage the aggregate power backoff at the satellite.)

As mentioned earlier, we assume the HPA is characterized by the Saleh model for nonlinear amplitude and phase distortions, [22]. The AM-AM and AM-PM characteristics of the Saleh model are shown below and graphically illustrated in Figure 2.7.

$$A(S) = \frac{\alpha_a u}{1 + \beta_a u^2} \quad (AM - AM) \quad (2.4)$$

$$\Phi(S) = \frac{\alpha_\phi u^2}{1 + \beta_\phi u^2} \quad (AM - PM) \quad (2.5)$$

where $u = |S|$ is the input signal magnitude. Both AM-AM and AM-PM characteristics depend only on the magnitude of the input signal.

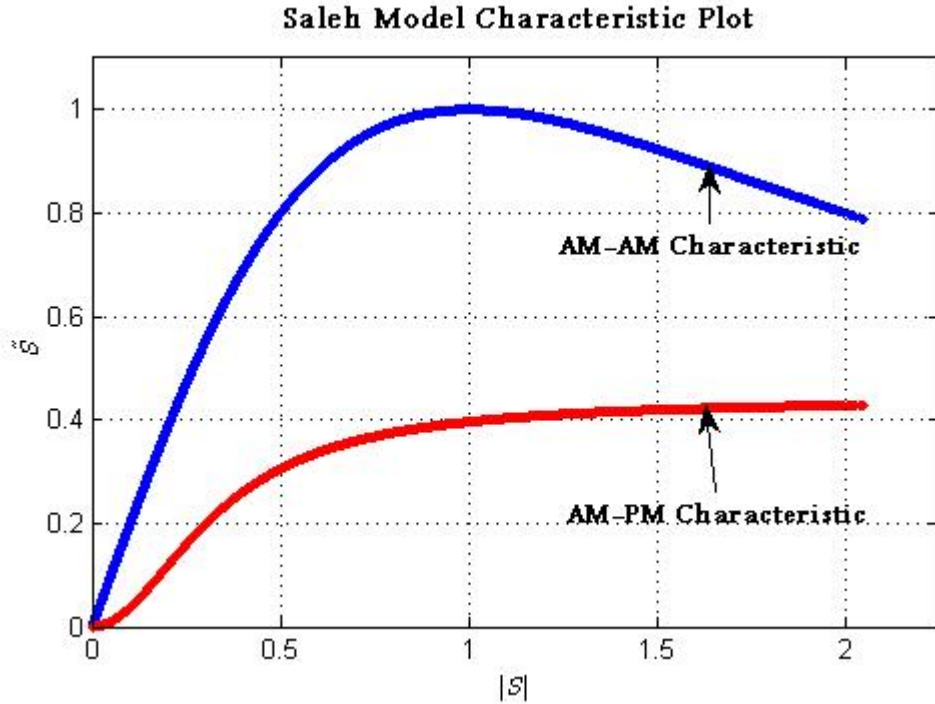


Figure 2.7: Plots of Saleh model AM-AM and AM-PM characteristics

The combined effects of the AM-AM and the AM-PM relations yield the following net effect on the

complex input S to the HPA. The input signal S can be written as

$$S = u \exp j\psi,$$

and the output of the Saleh model \tilde{S} can be written as

$$\begin{aligned}\tilde{S} &= A(S)e^{j\Phi(S)}e^{j\psi} \\ &= A(S)e^{j(\psi+\Phi(S))}\end{aligned}$$

expressing the nonlinear amplitude transfer characteristic as well as the amplitude-dependent phase modulation.

This distorted signal \tilde{S} is then relayed down to each ground terminal.

2.3 Receiver

The relay transmits the signal \tilde{S} to both the ground terminals. Now the terminals T_1 and T_2 act as receivers. Again we use one receiver to describe the structure and functionality as both receivers are identical in terms of structure and functionality. The receiver structure is depicted in Figure 2.7. The additive noise at the receiver further corrupts the received signal. The noise is modeled as additive white complex Gaussian noise (AWGN). Specifically, $n \sim \mathcal{CN}(0, \sigma^2)$ where σ^2 is the complex noise variance of the noise process at eight-times oversampling.

The goal is to recover the sequence S from the observed sequence \mathbf{Y} with high fidelity, say with small rms error. As discussed in Chapters 3 and 4, two approaches for recovering the aggregate waveform at the amplifier input have been studied. Whichever method is employed, we apply adaptive echo cancellation based on the side information available at each terminal and recover \hat{s}_j where j corresponds to the terminal where the message s_j is generated. The recovered signal \hat{s}_j is then passed through the matched filter and down-sampled to recover the original data sequence \hat{d}_j .

In the following section, details on the adaptive canceller are provided.

2.4 Adaptive Echo Cancellation

At the output of the matched filter, we have an approximation to what would have been received on a linear channel, though the original noise has been nonlinearly-modified by the inversion process. The Figure below illustrates this model.

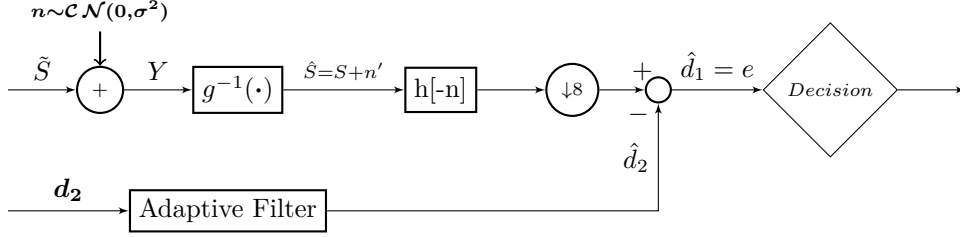


Figure 2.8: Receiver block diagram

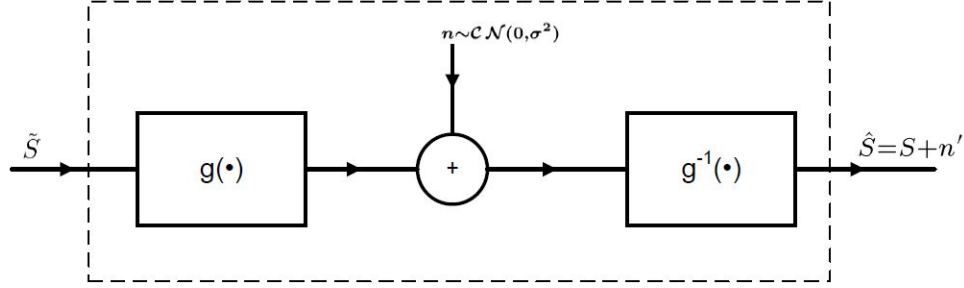


Figure 2.9: Ideal inversion leads to an overall linear channel with gain of 1

The particular adaptive filter deployed here is the least mean square (LMS) adaptive noise canceller [14], which is standard in the signal processing literature. We have a reference signal (the side-information sequence) and an input signal. The input signal contains the signal of interest which is corrupted in a linear fashion by the reference signal. Since we have information about the reference signal we can create an estimate of the reference signal in the input signal, then subtract it from the input signal. The relative symbol delay is easily incorporated into the adaptive filter. The error is then used to update the coefficients of the adaptive filter. The general structure and mathematics behind the adaptive filter are given below in Figure 2.10 and the subsequent equations.

We assume,

$$\hat{D} = \hat{d}_1 + \alpha \hat{d}_2$$

where α is an arbitrary complex constant. With reference to the diagram we also have

$$\hat{d}_2 = \sum_{i=0}^{Q-1} \hat{w}_i^*(n) d_2(n-i)$$

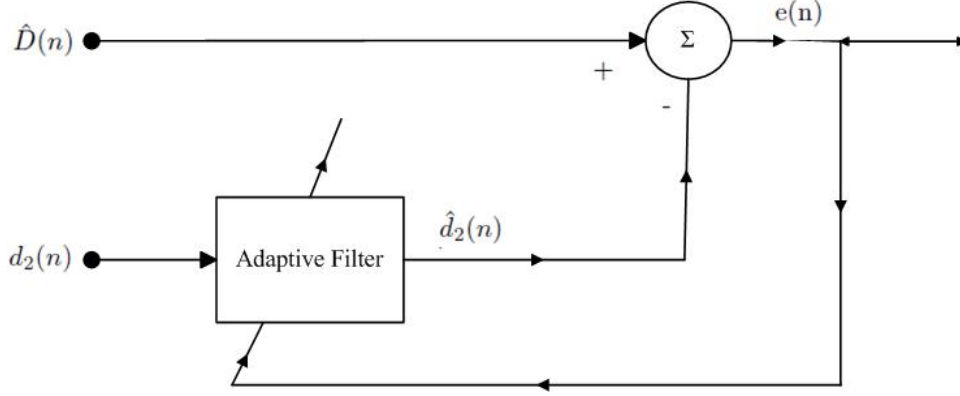


Figure 2.10: Structure of adaptive echo canceller

$$e(n) = \hat{D}(n) - \hat{d}_2$$

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu e^*(n) d_2(n-i), \quad i = \{0, 1 \dots Q-1\}$$

with μ a small adaptation scale factor that controls convergence speed and steady-state tracking jitter.

The Q -tap adaptive filter has complex weights at each tap as shown in Figure 2.11. These tap weights evolve with time and settle quickly to a set of weights that minimizes the mean-square value of $e(n)$, which is equivalent to the best removal of the echo. Decisions for the desired signal are then made based on $e(n)$.

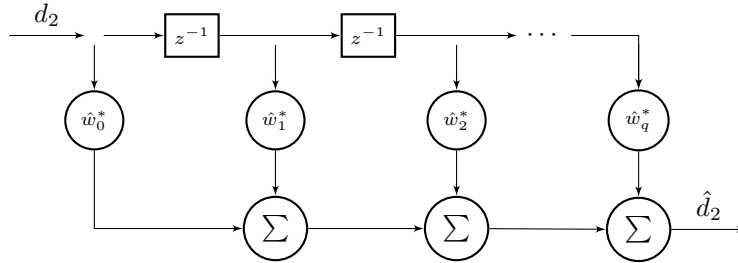


Figure 2.11: Adaptive filter structure

2.5 Centroid-based Decision

At the output of the adaptive noise canceller we have an estimate of the received signal. The next task is to map each received symbol to the symbol in the original constellation. This could be challenging as the received constellation might be degraded and distorted during the transmission. The received constellation might be rotated and translated because of the complex noise or be distorted due to the nonlinearity in the channel. Voronoi Tessellations are typically used to make decision zones for the received symbols. The

16-APSK constellation would be split into 16 non-overlapping zones each of which corresponds to a symbol in the constellation. This is shown for the ideal case in Figure 2.12.

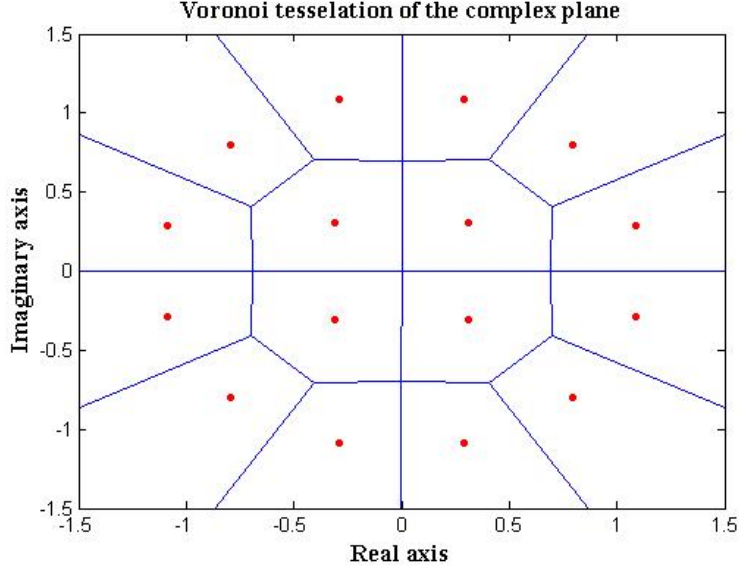


Figure 2.12: Voronoi tessellation of the complex plane

Another approach is the centroid based approach. Centroids are good ways to decide which Voronoi zone the received symbols should be classified into. The centroids are calculated by passing a known stream of symbols through channel and observing the received symbol \hat{a}_i for every a_i in the alphabet sent. The centroid corresponding to each transmitted letter a_i is obtained by taking the expected value of all N received \hat{a}_i 's at the receiver. By the law of large numbers we can use the mean instead if we use a long enough sequence.

$$c_i = \sum_{n=1}^N \hat{a}_{i,n} \quad (2.6)$$

This is also shown below in Figure 2.13, where the green scattered points are the residual of the adaptive canceller.

We can use training sequences to create tables of centroid for different values of backoff that can be deployed as per need. Once we have the centroids we can use them to find the modulus of the complex difference between a received symbol and all the centroids. The symbol in the original constellation that

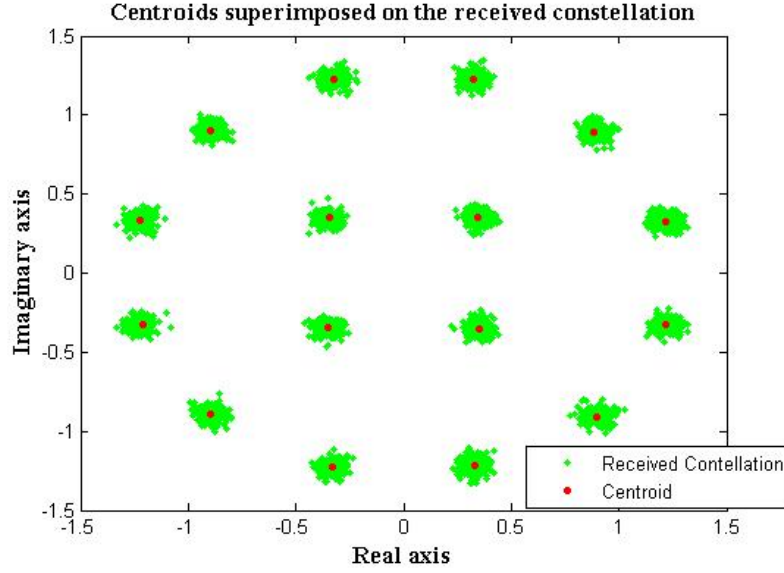


Figure 2.13: Received constellation and the corresponding centroids

is associated with the centroid that yields the smallest absolute difference is declared as the transmitted symbol. Another Voronoi tessellation of the received constellation is shown in Figure 2.14 where we can see that the constellations are contained inside the demarcated complex plane. An error occurs when the wrong centroid is selected and thus a wrong transmitted symbol is associated with the received symbol. The centroid based decision making is later used to evaluate the performance of the each algorithm.

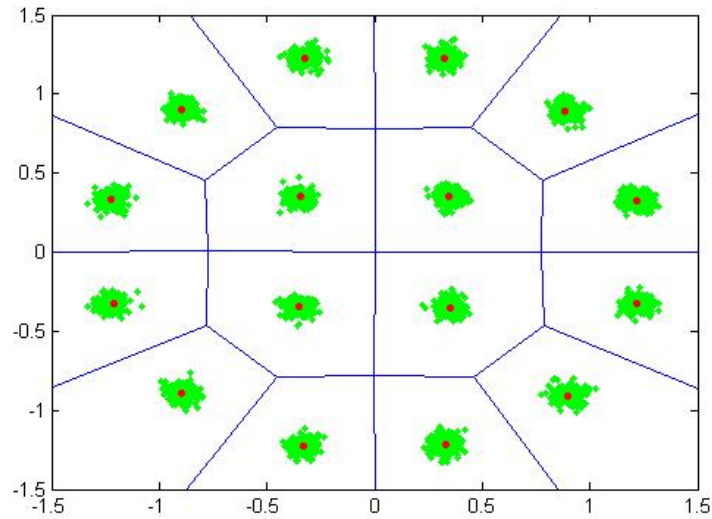


Figure 2.14: Voronoi tessellation of the complex plane

Now that we have defined the structure and functionality of the transmitter, relay, and the receiver, we now address how to obtain the inverter $g^{-1}(\cdot)$ as shown in Figure 2.8. In the next two chapters, we describe, analyze, and evaluate the two inverters that we have proposed. We start with the MAP inverter, the particle filter discussion follows in Chapter 4.

Chapter 3

MAP Detection Algorithm

3.1 Algorithm

This algorithm takes a ‘noiseless’ perspective on the inversion problem, and recognizes that based on the observation of the received signal there can be at most two amplifier input samples that could have produced this observation. We assume we know the amplifier characteristic, and solve for these two complex values. However, assuming we have some knowledge of the maximum possible amplitude of the input signal, which depends on the chosen backoff and the peak-to-average power ratio, we argue that only when the absolute value of the output of the nonlinear channel falls in some range slightly below the saturated output value, there exists two possible roots, and we call this an *ambiguous sample*. In other cases we can infer with reasonable certainty that the correct root lies in the certainty zone as shown in Figure 3.2 , and as will be discussed later.

When the input signal amplitude is large, there are typically several contiguous ambiguous samples, each of which has an associated pair of roots for the input sample. Over this ambiguous interval or span, we evaluate the prior probability of all possible 2^N input sequences, where N is the ambiguous span length. This probability density is a multidimensional complex Gaussian model with covariance matrix matched to the characteristics of the original pulse-shaped signal. More detail is provided on this below.

In Figure 3.1, we present the flow chart of the MAP inverter. The first step in the processing is to find complex input values corresponding to the observed complex output, i.e. we invert the nonlinearity sample-by-sample. Our model for the nonlinearity is the well-known Saleh model, which is a memoryless complex-baseband model for amplifiers, primarily traveling wave tubes, [22]. Figure 2.7 shows the characteristic plots of AM-AM and the AM-PM relations of the Saleh model assumed in this research. Note that the

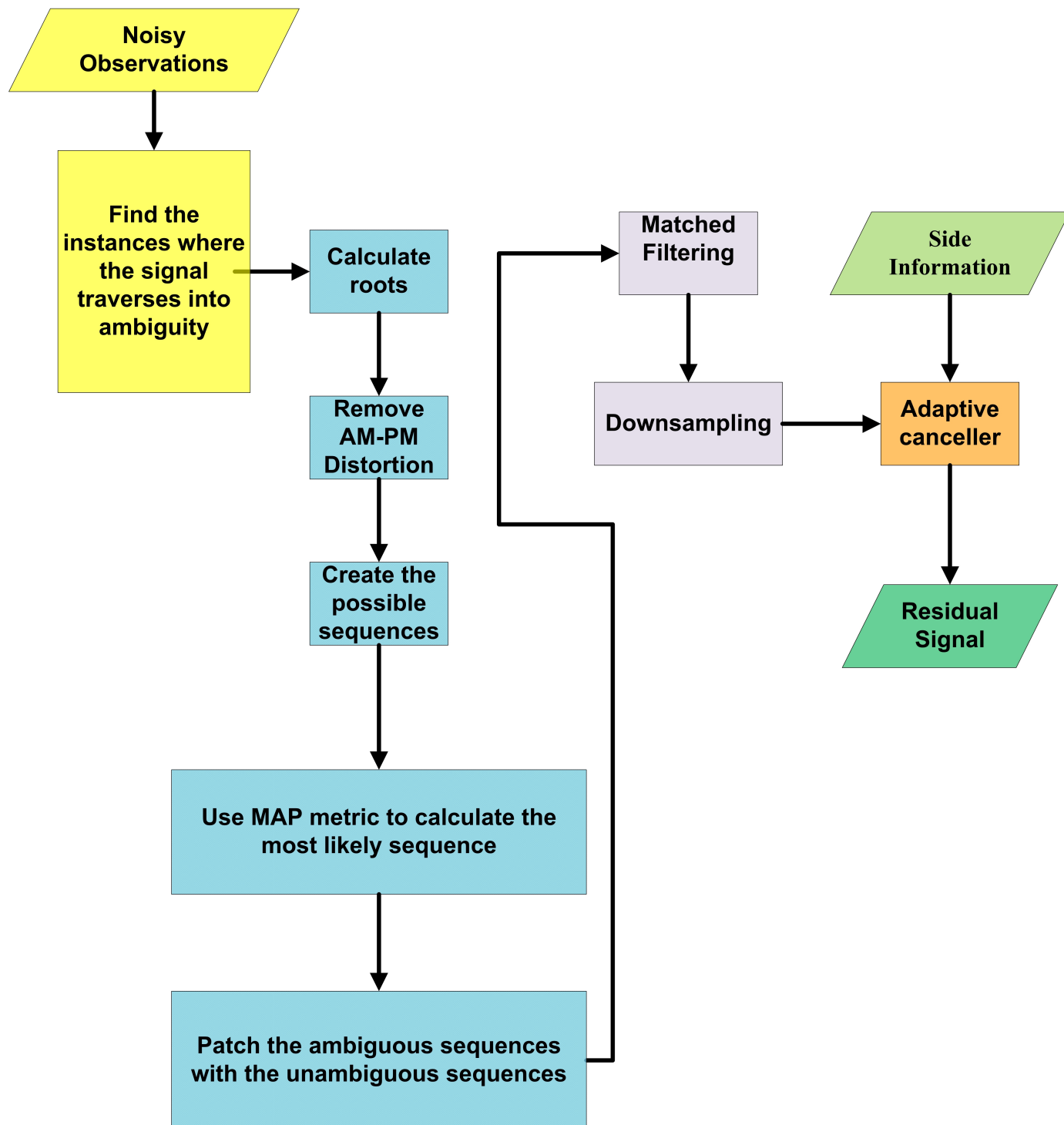


Figure 3.1: Flowchart of MAP inverter's algorithm

characteristic has been normalized so that the input amplitude corresponding to saturation is 1, and the output is also 1 at this point.

3.1.1 Inversion of Saleh model

We start by reiterating the assumption that the noise power is small compared to the signal power, *i.e.* the downlink SNR is high. This allows us to conclude that the additive noise is negligible and the received signal $Y \approx \tilde{S}$. Then we can write

$$Y = \frac{\alpha u}{1 + \beta u^2} \exp(j\angle Y) = \frac{\alpha |S|}{1 + \beta |S|^2} \exp(j\angle Y)$$

Where we have used a slight change of notation and introduced $u = |S|$.

As evident from Figure 2.7 the AM-AM characteristic of Saleh model is highly nonlinear, though for IBO much smaller than 0 dB, we operate in a quasi-linear regime. We wish to avoid this operational mode though as output power is small, and DC-to-RF efficiency is poor.

The mathematical inversion of the AM-AM relation is shown below.

$$A = |Y| = \frac{\alpha_a u}{1 + \beta_a u^2}$$

So we have a series of equations

$$\begin{aligned} A(1 + \beta_a u^2) &= \alpha_a u \\ A + \beta_a A u^2 &= \alpha_a u \\ (\beta_a A) u^2 - \alpha_a u + A &= 0 \end{aligned}$$

Using the quadratic formula we readily obtain the two roots for the input amplitude u as

$$u = \frac{\alpha_a \pm \sqrt{\alpha_a^2 - 4A^2\beta_a}}{2A\beta_a} \quad (3.1)$$

Because A will not exceed 1 under the high SNR assumption, both roots lie approximately on the Saleh curve depicted in Figure 2.7 and are real. In the case where additive noise cannot be neglected there exists the possibility that the magnitude of the noisy observation may be greater than 1. In such a situation we clip the magnitude down to 1. We define the two roots as

$$\begin{aligned} r_{lp} &= \frac{\alpha_a - \sqrt{\alpha_a^2 - 4A^2\beta_a}}{2A\beta_a} \\ r_{hp} &= \frac{\alpha_a + \sqrt{\alpha_a^2 - 4A^2\beta_a}}{2A\beta_a} \end{aligned}$$

which could be called ‘inner’ and ‘outer’ roots, since one will have magnitude smaller than 1, and the other’s magnitude exceeds 1.

Once we have obtained the two possible input amplitudes we substitute them into the AM-PM characteristic. This yields the corresponding AM-PM conversion that was introduced by the amplifier. Once we have the AM-PM conversion, it is just a matter of subtracting it from the phase of the received signal \tilde{S} . The phase distortion quantities ϕ_1 and ϕ_2 are

$$\phi_1 = \frac{\alpha_\phi r_{lp}^2}{1 + \beta_\phi r_{lp}^2},$$

$$\phi_2 = \frac{\alpha_\phi r_{hp}^2}{1 + \beta_\phi r_{hp}^2},$$

The quantities ϕ_1 and ϕ_2 are then used to calculate the two final complex input samples, specifically

$$r_l = r_{lp} e^{j(\angle \tilde{S} - \phi_1)}$$

$$r_h = r_{hp} e^{j(\angle \tilde{S} - \phi_2)}$$

Thus, we produce a pair of complex samples for the unknown input sequence to the amplifier for every ambiguous received measurement.

Once we have the roots for the transmitted sequence, the next step is to decide the string of samples that was most likely sent. (By operating over a span of samples we have some chance of inverting this fundamentally nonlinear system by exploiting the known probability structure of the input sequence.)

This is greatly simplified if we realize that many of the samples *are* invertible, namely those corresponding to small input amplitude. But we can only observe the output amplitude A . If we place some upper bound on the size of $u = |S|$, based on known IBO and PAPR, we can identify a zone of ambiguity as shown in Figure 3.2. Let this maximum value that $|S|$ can take on be S_{max} , and the output produced by this value, A^* , be the threshold point above which there is ambiguity in the selection of correct root.

$$A^* = \frac{\alpha_a S_{max}}{1 + \beta_a S_{max}^2}$$

The value of A^* is developed using Monte Carlo methods. Long sequences with a given backoff are generated and the maximum values are recorded and used to calculate A^* . Look-up tables are created to store the values of A^* for different operating backoffs. The motivation behind this is that in saturation, *i.e.* beyond A^* , for each Y value there are two possible values for S and we have further processing to do, but when $|Y| \leq A^*$, ambiguity about the input does not exist.

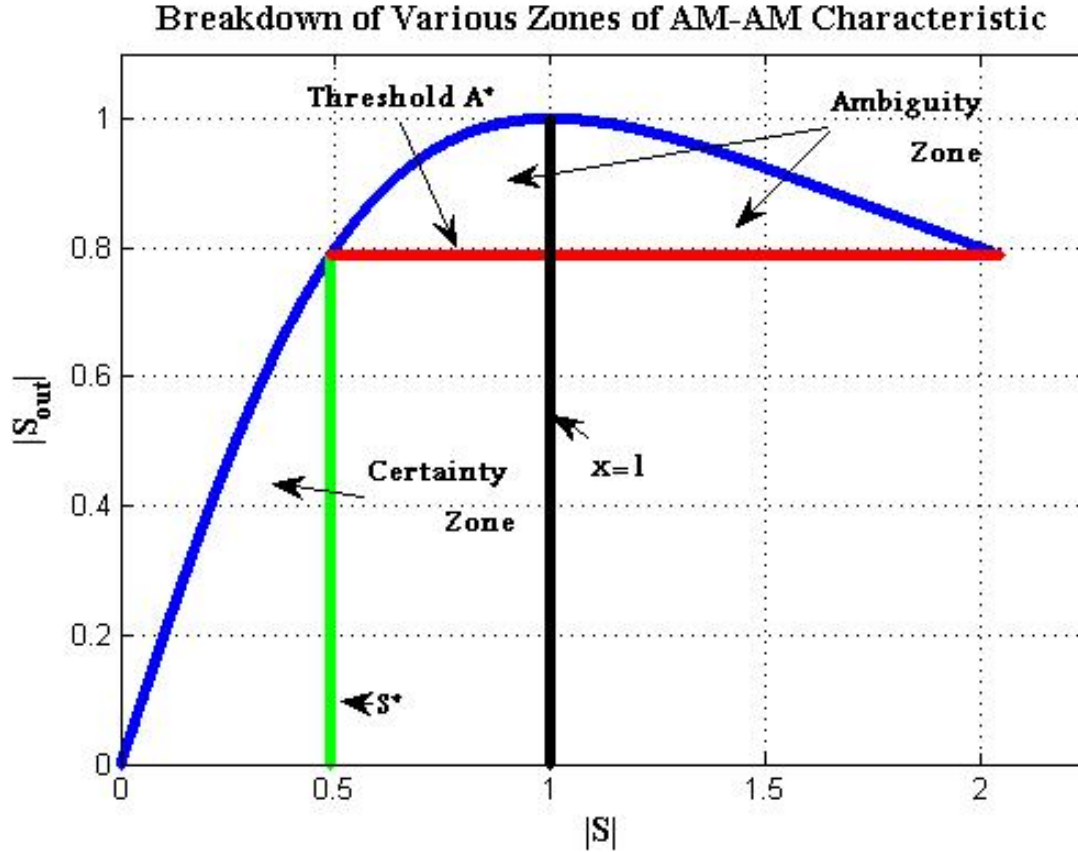


Figure 3.2: Identification of various zones in the Saleh AM-AM characteristic

3.1.2 MAP sequence selection

The received sequence is parsed into segments of contiguous ambiguous spans, beginning with traversal into the dual-root zone, and ending with leaving this zone. To process each ambiguous span, where there is uncertainty about the input sequence, we prepend three non-ambiguous samples, and append at the end two non-ambiguous samples. We call these *anchor points* for the probability calculation. These anchor points provide certainty at the starting and ending of the ambiguous sequence. Since we have certainty about the points before the sequence enters into the ambiguous zone we can use this known history and smoothness information based on the correlation between consecutive samples to help decide the best choice of roots in the ambiguous zone.

The length of the ambiguous subsequences can range from a few samples to a few hundred samples based on the operating input backoff at the satellite. In these long spans the ambiguous sequence is parsed into subsequences of length N to reduce the computational complexity. For the first N samples in the ambiguous

zone the leading anchor points are used and once these N samples are estimated they serve as anchors for the subsequent N samples. Therefore is imperative to estimate these initial samples with as little error as possible. More anchor points therefore furnish more information about the past and help estimating the initial ambiguous samples with little error. An ambiguous sequence is shown in Figure 3.3 for the case when the ambiguous span is N samples.

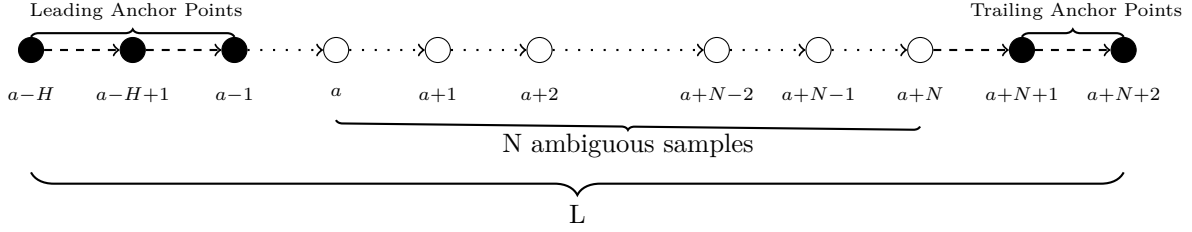


Figure 3.3: Depiction of an ambiguous sequence

Since each ambiguous sample can take two possible values, for a sequence of length N there are 2^N possible sequences. For example, if $N = 2$ then there are 4 possible sequences of inner and outer roots, shown below in matrix form.

$$\begin{pmatrix} r_l & r_l \\ r_l & r_h \\ r_h & r_l \\ r_h & r_h \end{pmatrix}$$

Now to decide an ambiguous sequence of length N we append the leading and trailing anchor points and obtain a sequence of length $L = N + 5$ as shown in Figure 3.3. For computational purposes we define a $2^N \times L$ matrix \mathbf{J} , in which the first three columns have the same entries in all 2^N rows, corresponding to leading anchor points. Moreover, the last two entries in each row correspond to the trailing anchor points. The N entries in the middle, however, can take on two possible values and there are 2^N such possible combinations akin to all the possible binary number sequences represented by N root pairs. The structure of \mathbf{J} is illustrated below.

$$J = \begin{pmatrix} r_{l_{a-H}} & \cdots & r_{l_{a-1}} & r_{l_a} & r_{l_{a+1}} & \cdots & r_{l_{a+N-2}} & r_{l_{a+N-1}} & r_{l_{a+N}} & r_{l_{a+1}} & r_{l_{a+2}} \\ r_{l_{a-H}} & \cdots & r_{l_{a-1}} & r_{l_a} & r_{l_{a+1}} & \cdots & r_{l_{a+N-2}} & r_{l_{a+N-1}} & r_{h_{a+N}} & r_{l_{a+N+1}} & r_{l_{a+N+2}} \\ \vdots & \cdots & \vdots & r_{l_a} & r_{l_{a+1}} & \cdots & r_{l_{a+N-2}} & r_{h_{a+N-1}} & r_{l_{a+N}} & \vdots & \vdots \\ \vdots & \cdots & \vdots & r_{l_a} & r_{l_{a+1}} & \cdots & r_{l_{a+N-2}} & r_{h_{a+N-1}} & r_{h_{a+N}} & \vdots & \vdots \\ \vdots & \cdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots \\ r_{l_{a-H}} & \cdots & r_{l_{a-1}} & r_{l_a} & r_{h_{a+1}} & \cdots & r_{h_{a+N-2}} & r_{h_{a+N-1}} & r_{h_{a+N}} & r_{l_{a+N+1}} & r_{l_{a+N+2}} \\ r_{l_{a-H}} & \cdots & r_{l_{a-1}} & r_{h_a} & r_{h_{a+1}} & \cdots & r_{h_{a+N-2}} & r_{h_{a+N-1}} & r_{h_{a+N}} & r_{l_{a+N+1}} & r_{l_{a+N+2}} \end{pmatrix}$$

The decision problem in the ambiguous zone is a binary decision problem. We have to decide which of the 2^N sequences is most probable (hence the name MAP).

3.1.3 MAP root selection procedure

The method used to select which of r_l or r_h is more appropriate is described now. We make another assumption here that the samples are normally-distributed, after pulse shaping, despite the discrete-alphabet constellation. This is especially true in the bidirectional case where we have two users at the input of the nonlinear channel. The assumption of Gaussianity was further tested experimentally and it was observed that the samples approximately fit a Gaussian distribution.

The L -dimensional multivariate Gaussian pdf is given by

$$f(x) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|} \exp\left[-\frac{1}{2}(\mathbf{x} - \mu)^\dagger \Sigma^{-1}(\mathbf{x} - \mu)\right]$$

where μ is the vector of means and Σ is the covariance matrix.

Maximum a posteriori probability (MAP) estimation is used to estimate the sequence $\hat{\mathbf{u}}$ based on the observed sequence \mathbf{Y} , that maximizes the above pdf *i.e.*,

$$\hat{\mathbf{S}} = \arg \max_{\mathbf{x}} f(\mathbf{x}|\mathbf{y})$$

Alternatively, this can be implemented by simply minimizing the quadratic form in the exponential of the density function. Moreover, taking into account the fact that the constellation has mean zero the MAP estimation problem evolves into the following:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \mathbf{x}^\dagger \Sigma^{-1} \mathbf{x}$$

The correlation matrix is obtained by using the correlation properties introduced by the pulse shaping filter. The convolution of the pulse shaping filter and its flipped version, *i.e.* the matched filter, is used to compute

the correlation and the cross-correlation between the samples in the sequence. In this case due to the symmetry of the pulse shaping filter the convolution and the correlation operations are the same. If we assume that the pulse shaping filter is of length M , then the convolution results in a sequence of length $2M - 1$. The maximum value of the above convolution occurs at the M^{th} sample. This is depicted in Figure 3.4 below.

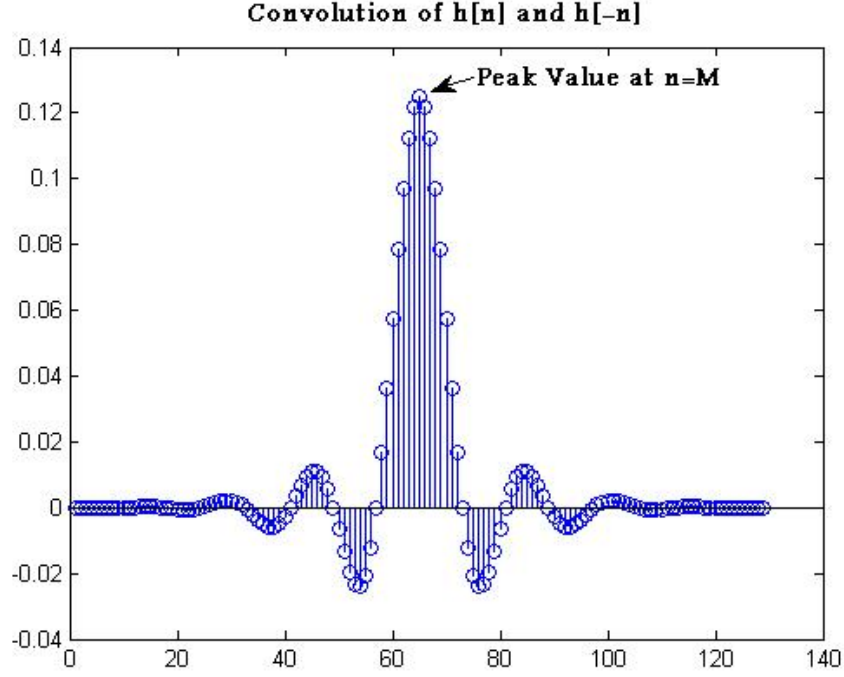


Figure 3.4: Result of above convolution provides value of required correlation between samples

The convolution result is normalized and then a suitable number of values in the neighborhood of the peak value are selected to match the length L of the ambiguous span as described previously. Moreover, this is also scaled by the input backoff to have the right scale, or variance.

A $L \times L$ Toeplitz matrix, \mathbf{K} , is created from the auto-correlation function for the samples. The Toeplitz matrix is portrayed below.

$$K_{L \times L} = \begin{pmatrix} z_M & z_{M+1} & z_{M+2} & z_{M+3} & \cdots & \cdots & z_{M+L-2} & z_{M+L-1} \\ z_{M-1} & z_M & z_{M+1} & z_{M+2} & \ddots & \ddots & z_{M+L-3} & z_{M+L-2} \\ z_{M-2} & z_{M-1} & z_M & z_{M+1} & \ddots & \ddots & z_{M+L-4} & z_{M+L-3} \\ z_{M-3} & z_{M-2} & z_{M-1} & z_M & \ddots & \ddots & z_{M+L-5} & z_{M+L-4} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ z_{M-(L-1)} & z_{M-(L-2)} & z_{M-(L-3)} & z_{M-(L-4)} & \cdots & \cdots & z_{M-1} & z_M \end{pmatrix}$$

Moreover, because the result $Z = h[n] \star h[-n]$ is symmetric, the Toeplitz matrix is also symmetric, and it can be rewritten as follows and then scaled by the input backoff.

$$K_{L \times L} = \begin{pmatrix} z_M & z_{M+1} & z_{M+2} & z_{M+3} & \cdots & \cdots & z_{M+L-2} & z_{M+L-1} \\ z_{M+1} & z_M & z_{M+1} & z_{M+2} & \ddots & \ddots & z_{M+L-3} & z_{M+L-2} \\ z_{M+2} & z_{M+1} & z_M & z_{M+1} & \ddots & \ddots & z_{M+L-4} & z_{M+L-3} \\ z_{M+3} & z_{M+2} & z_{M+1} & z_M & \ddots & \ddots & z_{M+L-5} & z_{M+L-4} \\ \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ z_{M+L-1} & z_{M+L-2} & z_{M+L-3} & z_{M+L-4} & \cdots & \cdots & z_{M+1} & z_M \end{pmatrix}$$

$$\mathbf{K}_{TOEP_{L \times L}} = \sigma_\rho^2 \mathbf{K}$$

Now the MAP estimation can be restated as

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{J}_i} \mathbf{J}_i^T \mathbf{K}_{TOEP}^{-1} \mathbf{J}_i \quad \text{where } i = \{1, 2, \dots, 2^N\}$$

The above calculation is performed for each row and the result is a scalar quantity which serves as the a-posteriori probability of the i^{th} row as the transmitted sequence. We store the metrics in a $2^N \times 1$ vector, and the row with minimum entry in is the row in J that corresponds to the best estimate of the transmitted sequence among the 2^N possible sequences.

Finally, we concatenate all the sequences we have certainty about with the estimates of the ambiguous sequences to obtain the estimated signal \hat{S} .

3.1.4 Matched filtering and down-sampling and adaptive cancellation

We assumed a high SNR at the receiver which allowed us to use the algorithm defined above to ‘invert’ the nonlinearity, resulting in a virtual linear channel. We pass the estimated signal to the matched filter which we have described in detail previously. The output of the matched filter is then downsampled by 8 with the correct downsampling phase depending on the delay of the received signal. For the purposes of this document we ignore the link delay by assuming that we exactly know what it is and that we have compensated for it. The downsampled signal \hat{D} in the ideal case would be all the possible combinations that result from adding the symbols in two constellation points. The ideal combination of the two synchronized 16-APSK signals with the same average energy yield the constellation as shown in Figure 3.5. Realistically, however, the recovered total constellation is not as well defined as there is linear and nonlinear ISI that result from

the effects of windowing and the nonlinear channel respectively. The signal \hat{D} is then sent to the adaptive filter. The adaptive filter at each terminal uses the side information to delete the signal that originated at the terminal from the aggregate signal. The residual of the adaptive canceller is the estimate of our desired data signal.

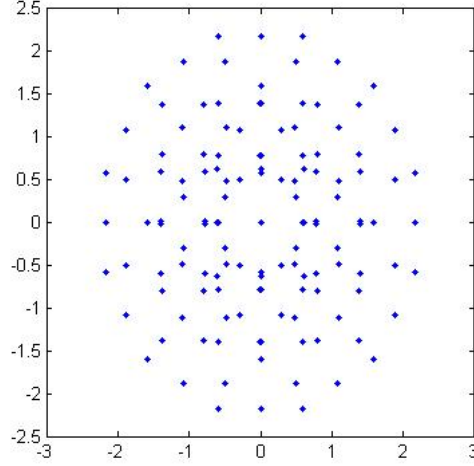


Figure 3.5: The ideal sum of two (synchronized) 16-APSK constellations

3.2 Performance Evaluation

We now demonstrate the performance of the algorithm by simulating it in Matlab. Here we use two methods to evaluate the performance of the inverter. Since the MAP inverter is a waveform tracker we first evaluate the tracking performance by using the mean squared error (MSE) between the true signal and the estimated signal. The second criterion used to judge the performance of the MAP inverter is the symbol error rate (SER) based on the final constellations obtained at the output of the linear adaptive canceller. We perform these analysis by varying the received SNR, which is analogous to E_{sat}/N_o , and the input backoff. Before we go and evaluate the performance of the inverter for bidirectional communication over a nonlinear channel we gauge its performance on the nonlinear channel with a single user *i.e.* a setup with one transmitter and one receiver. The performance over the bidirectional setup with the nonlinear channel ensues.

3.2.1 Single user

The single-user problem is studied first as the PAPR of the this case is lower and the signal spends a lower amount of time in the saturation zone shown in Figure 3.2. Moreover, it allows us to make important

conclusions about the general performance of the algorithm. The two parameters we are interested in are the SNR at the receiver and the input backoff at the input of the nonlinear channel.

3.2.1.1 Effect of SNR

Intuitively, one should guess that the SNR is inversely proportional to the MSE and the SER. We first study the locus tracking results. For this we fixed the backoff at -5 dB and swept across the SNR values from 20 dB to 40 dB with increments of 4 dB. We demonstrate the results below in figure. The plot in Figure 3.6 confirms our intuition that the as SNR increases the MSE gets smaller. Moreover, the MSE decreases monotonically, at least over the range shown in Figure 3.6. Moreover we see that the slope is approximately linear. There is about an order of magnitude decrease in the MSE with the increase in SNR by a decade.

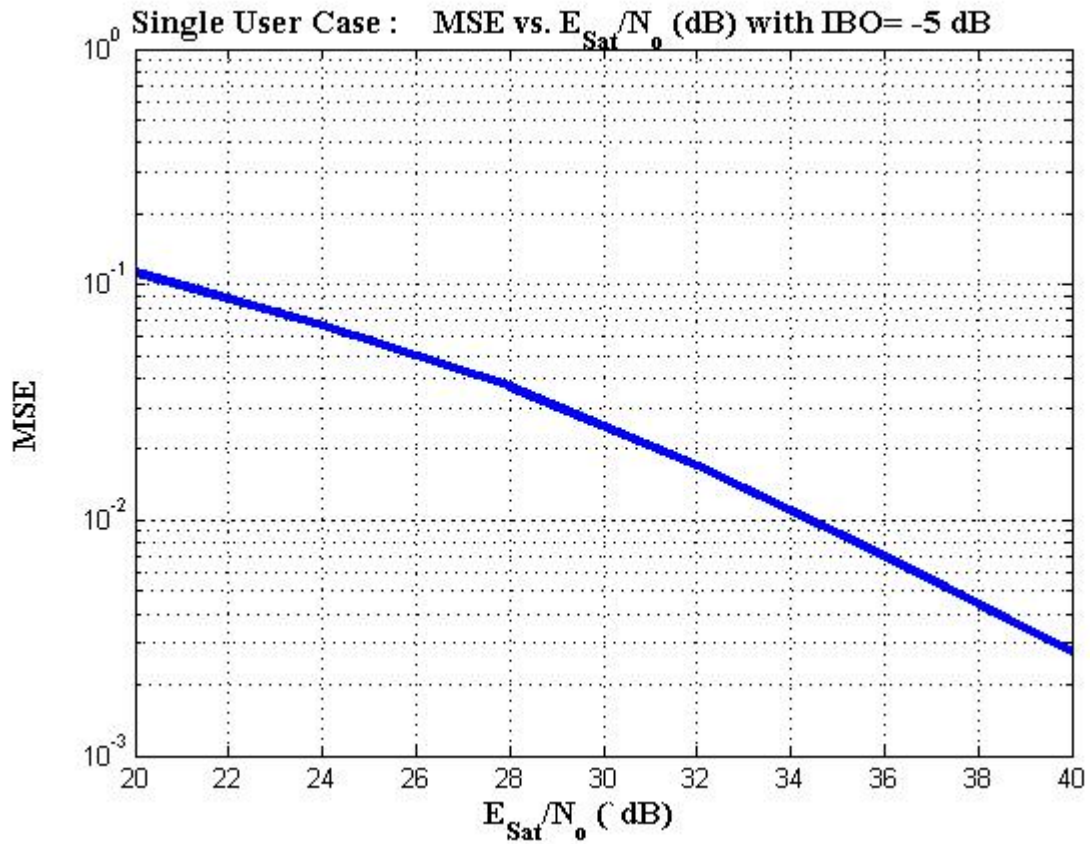


Figure 3.6: MSE vs E_{Sat}/N_o for single user case

3.2.1.2 Effect of backoff

As discussed previously, the backoff controls the severity of the nonlinearity *i.e.* with smaller backoffs the amplifier is driven harder into saturation and therefore the nonlinear distortion introduced is more severe. Before actually running the simulations we reiterate that with a harder drive level we will have more episodes where the signal is in the ambiguous zone depicted in Figure 3.2. This implies that the inverter has to make a greater number of estimations and therefore increases the chances of making error in locus tracking. We now present the results of the simulation below in Figure 3.7 where the SNR was fixed at 35 dB, and the backoff was varied from -10 dB to -2 dB. The algorithm was simulated using the same realization of the input signal for each backoff.

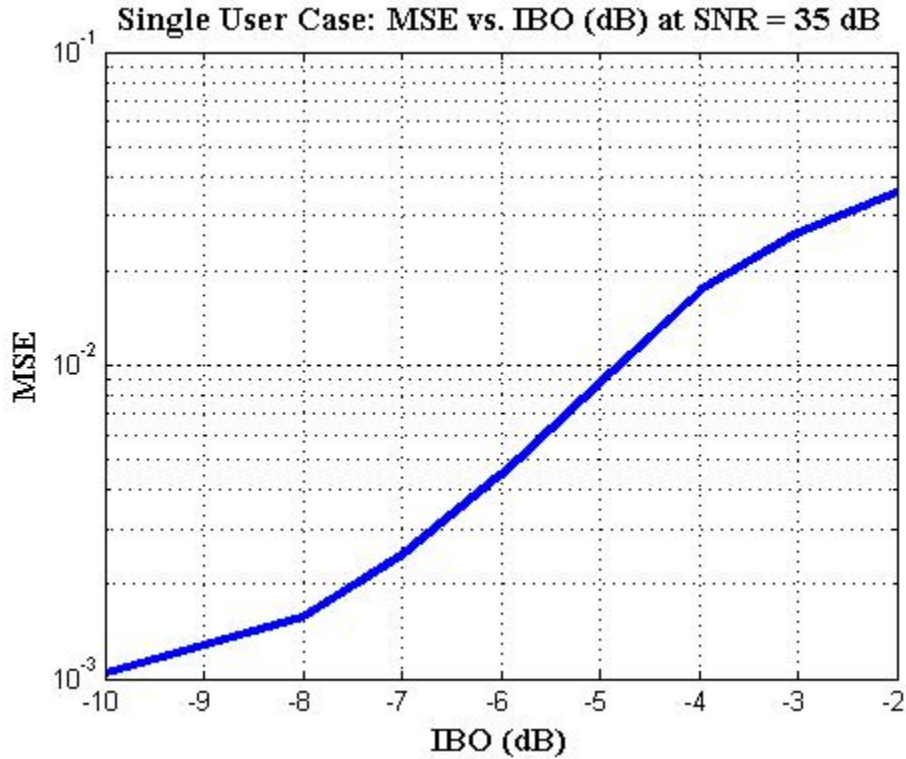


Figure 3.7: MSE vs IBO with SNR =35 dB for single user case

3.2.2 Bidirectional case

We now divert our attention back to the bidirectional case which is the problem of interest. We now try to repeat the same experiments for bidirectional case and see if and to what extent the results produced in the single-user case hold. Again the criteria used for evaluation is MSE. In addition we now also calculate the

SER based on the constellations recovered at the output of the adaptive canceller. We start by first studying the implications of SNR in the bidirectional case.

3.2.2.1 Effect of SNR

As mentioned earlier, in the bidirectional case the signal tends to stay longer in saturation. The reason is that the aggregate signal can be much larger where the two individual signals interfere constructively and therefore drives the amplifier harder. This introduces nonlinear distortion that is more severe than in the single user case. As the number of ambiguous episodes increase, so does the chance of MAP inverter making more tracking errors. We now present the plot of MSE as a function of SNR for the bidirectional case. We fixed the backoff at -5 dB and then simulated the algorithm over a series of SNRs. The results are shown in Figure 3.8. As expected the MSE does not decrease with the same slope as the single user case and decreases more slowly with the increased SNR.

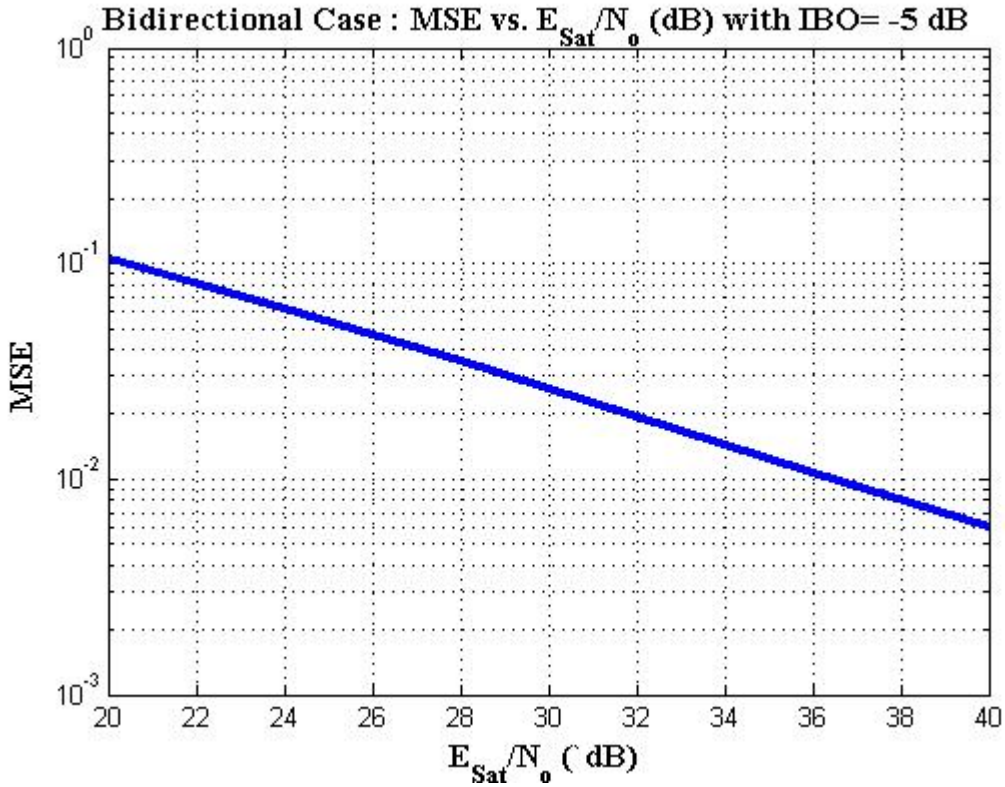


Figure 3.8: MSE vs E_{Sat}/N_o with IBO = -5 dB for bidirectional case

We now present some of the results of locus tracking in the two user case at different SNRs. These locus plots are good qualitative indicators of performance and we notice that they are increasingly noisy as SNR

drops, but no huge loss of track is in evidence.

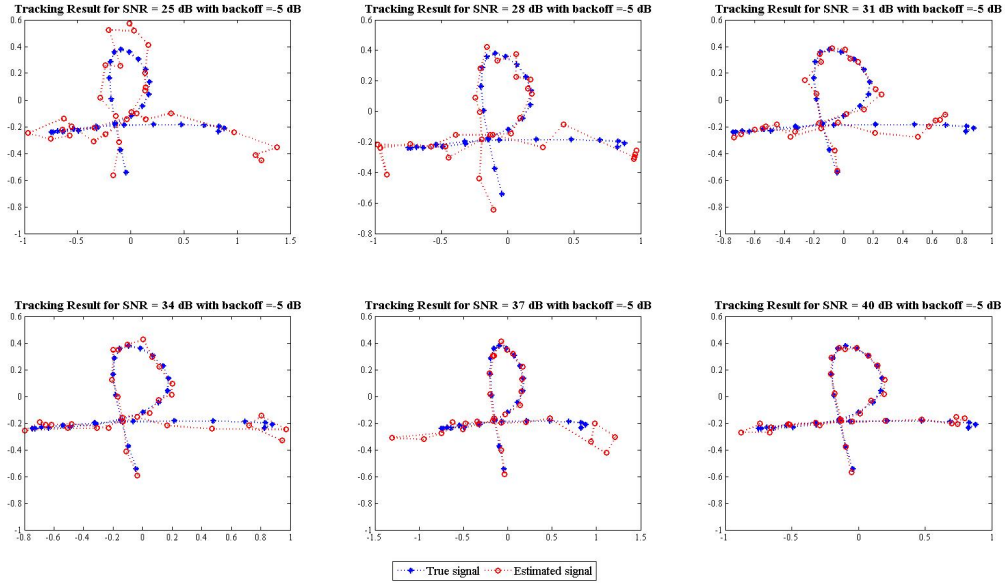


Figure 3.9: Locus tracking of a sub sequence using MAP inverter at various SNRs

A plot of the tracking error magnitude over a span of 10000 samples is shown in Figure 3.10, showing generally small error due to channel noise, but occasional large errors where the wrong sequence is selected by the MAP inverter. So we conclude that the tracking performance gets better with increasing SNR.

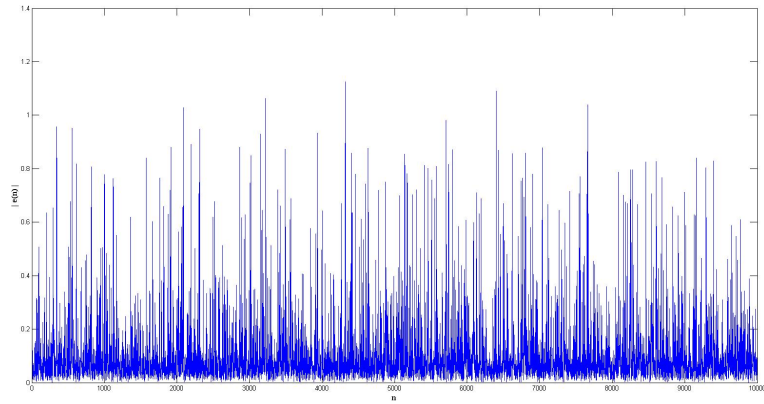


Figure 3.10: Standard tracking error between the true and the estimated signal, SNR = 30 dB, backoff = -5 dB

3.2.2.2 Effect of backoff

In the bidirectional case, the two signals can interfere constructively and produce an aggregate signal with peak power higher than the average power. The frequency of aggregate power being higher than the average power increases at lower backoffs. The consequence of this is that the amplifier is now driven harder into saturation thus producing more nonlinearly distorted signals more often. To reduce the effect of saturation the signal has to be backed off considerably more than in the single user case to achieve similar results. Also in the two user case higher PAPR implies that the threshold amplitude for ambiguity, A_{\star} , is going to be lower. Consequently, our certainty zone, as shown in Figure 3.2, has shrunk and now the MAP estimator is evoked more often and has to invert instances where the distortion is more severe than in the single user case. We now compare the effect of different input backoffs for a fixed SNR. In addition, for a given SNR, additive noise has the same average power for each backoff. However, higher backoffs reduce the signal power and as result we see a much lower effective SNR at the receiver. However, since larger backoffs lead to signal operating mostly in the certainty zone, there are less inferential errors from the MAP inverter as it is evoked less frequently and are mostly attributed to the additive noise. But with smaller backoffs, the power average signal power does increase but the signal has a higher degree of nonlinear distortion. We depict this in Figure 3.11. Notice the MSE increases much faster as compared to the single user case when backoff is reduced. We also show the MSE versus SNR trend for various backoffs for the MAP inverter in Figure 3.12. The MSE decreases monotonically over the range of SNRs used to run the simulation. However, at lower backoffs the slopes are much flatter and the MSE is not reduced significantly by increasing the SNR. This is a consequence of the nonlinear distortion.

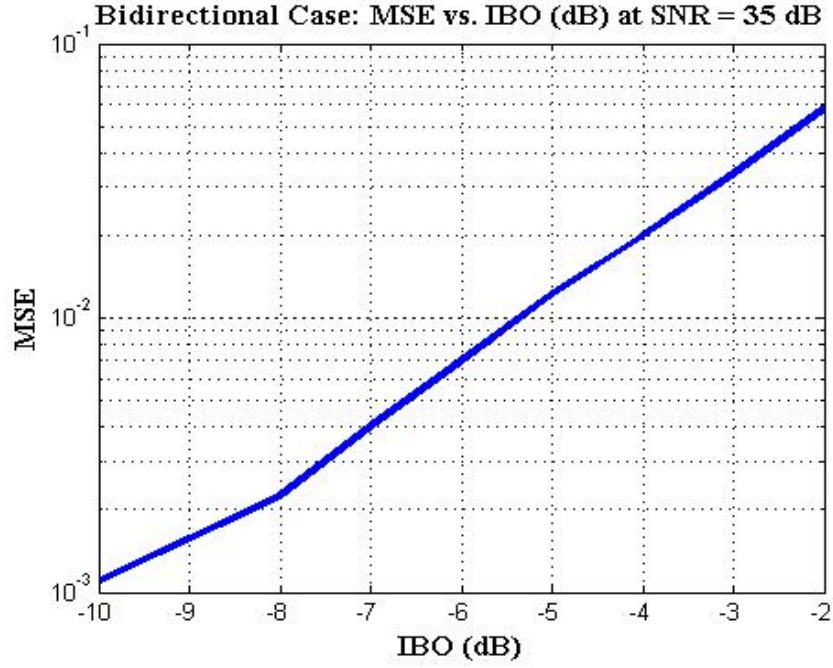


Figure 3.11: MSE vs IBO with SNR = 35 dB for bidirectional case

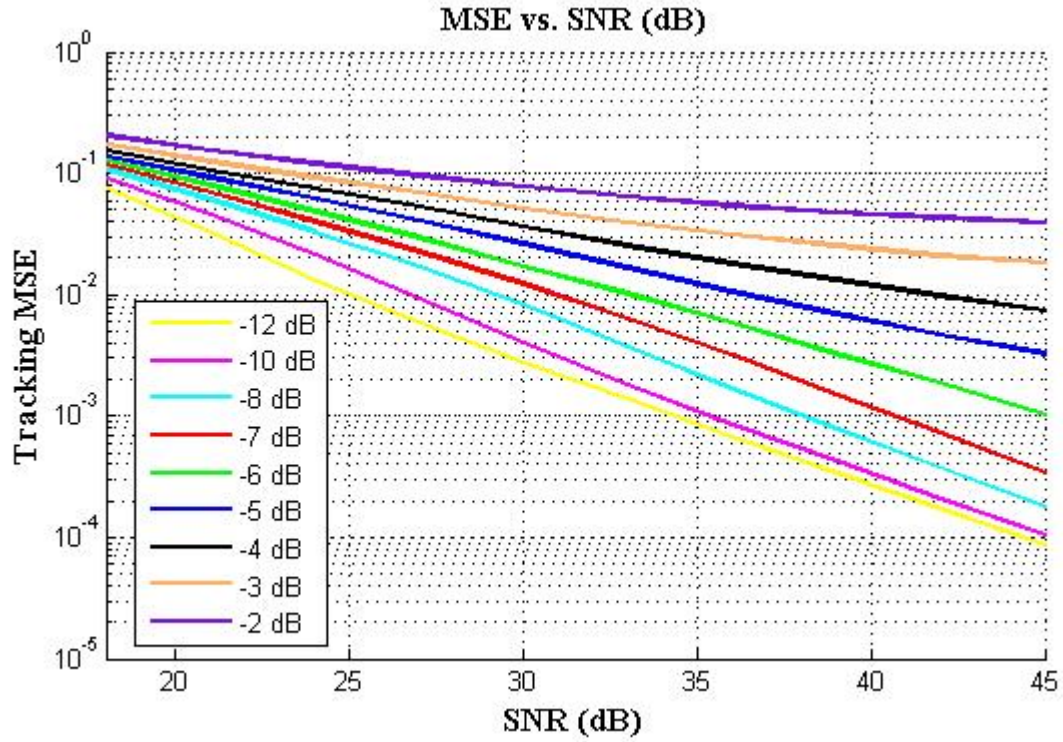


Figure 3.12: MSE vs SNR at various backoff levels, bidirectional case

3.2.2.3 Constellation quality and error rate curves

We now show the final constellations obtained at the output of adaptive canceller for 4 different SNRs at an input backoff of -5 dB. The constellations are shown in Figures 3.13 and 3.14. From visual inspection we notice that the quality of constellations get better as the SNR is increased. At lower SNRs we see a additive Gaussian noise around the constellation points. However, at higher SNRs, *e.g.* 40 dB, we notice the reduction in the additive noise but nonlinear artifacts appear in the constellation. The outer constellation points exhibit radial distortion. Also, notice that the inner points experience less distortion as expected since their loci are smaller magnitude. The reason the distortion on the constellation points is more radially distributed is less clear. The noise can cause the inverter to incorrectly choose the wrong root and since the two phase corrected roots almost lie on a radial line we observe a radial scatter around the outer constellation points.

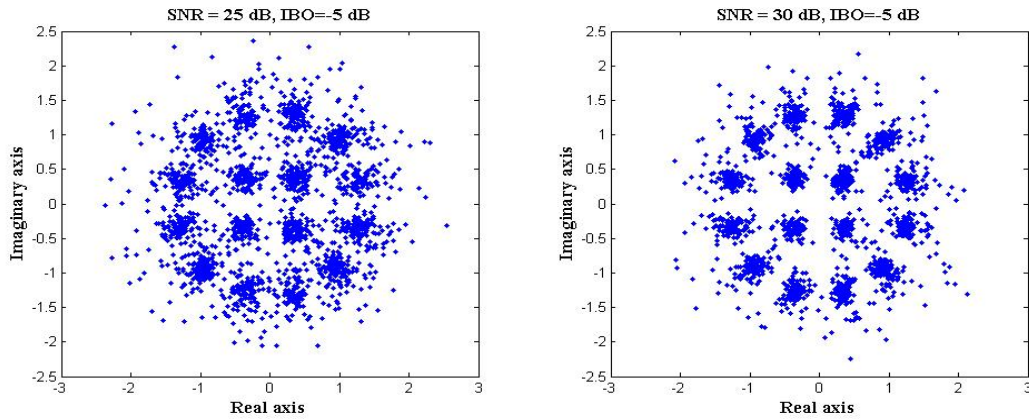


Figure 3.13: Recovered constellation for SNRs 25(left) and 30(right) dB with IBO = -5 dB

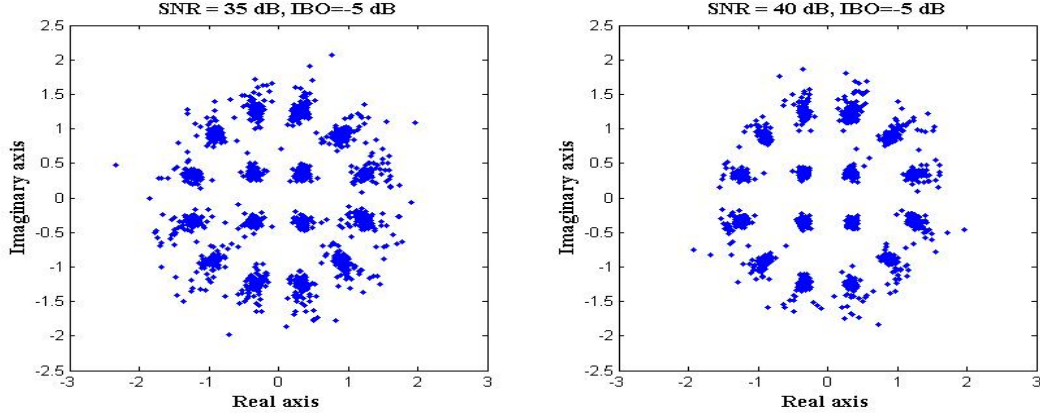


Figure 3.14: Recovered constellation for SNRs 35(left) and 40(right) dB with IBO = -5 dB

Next we evaluate the performance of the inverter from the stand point of final constellation recovery and the symbol error rate (SER). At the output of the adaptive canceller, the centroid based approach defined in Chapter 2 is used to make decision about each estimated symbol. Each time the transmitted symbol and the estimated symbol don't match we register an error. We use 100,000 symbols to generate the SER curves. The SER curves for various backoffs are shown below in Figure 3.15.

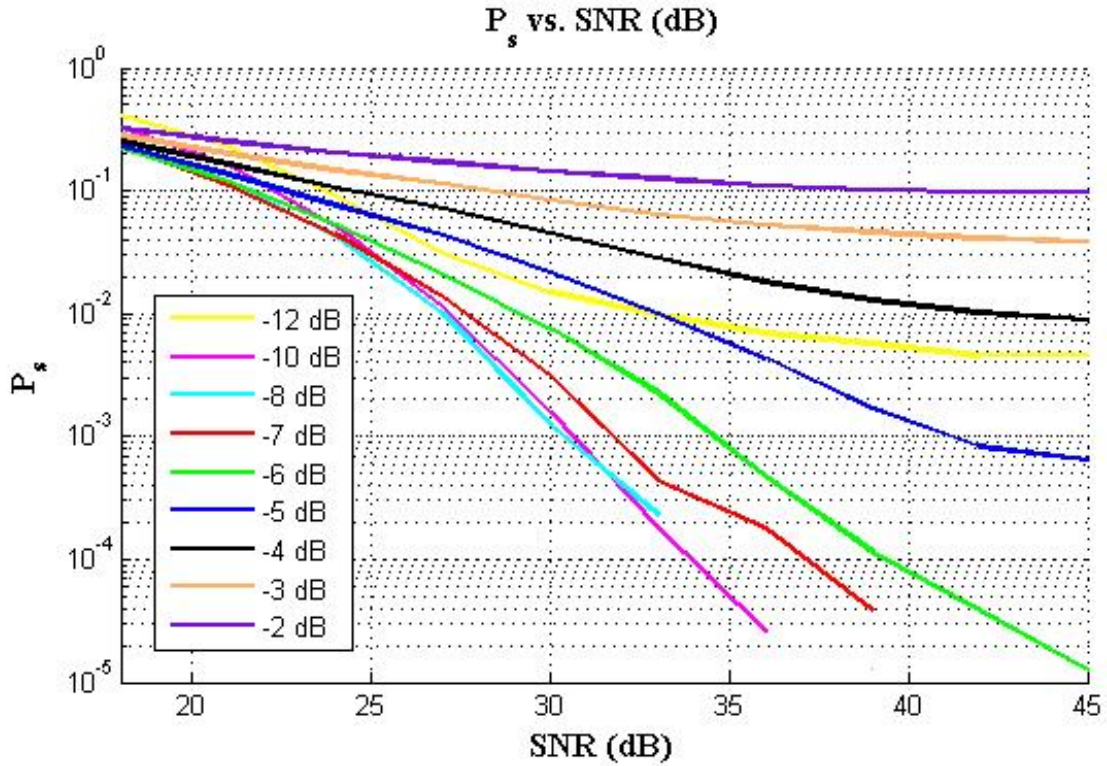


Figure 3.15: SER plots for MAP inverter

We observe that at small backoffs the error rates hit a floor and increasing the SNR does not improve the error rate. This is a consequence of the severity of nonlinear distortion at smaller backoffs. At intermediate backoffs *i.e.* from -8 to -5 dB we do see a decrease in symbol error probability with increased SNR.

3.3 Summary

In this Chapter we addressed the two-terminal satellite communication problem where the satellite relay is used by the terminals simultaneously. We allow for the signals to interfere at the relay and the aggregate signal is amplified and forwarded to each ground terminal. The amplification model is highly nonlinear and we presented a probabilistic algorithm to successfully invert the nonlinearities introduced at the relay. Moreover, we demonstrated that MAP detection approximately creates an overall linear channel with gain 1. Then we used side-information along with adaptive echo cancellation to get rid of the interfering signal and recover an estimate of the desired signal from the other user.

A primary concern about this method is computational complexity when the span length L is large, as the number of hypotheses to be evaluated grows exponentially with L . Parsing into subsequences is a work-around on this, but this introduces performance degradation due to uncertain anchor points for subsequent segments. Also, the method is sensitive to downlink noise and does not perform well when the SNR's fall below 25 dB, although the mean-square tracking error improves monotonically with SNR. Therefore we need a more robust method that can tackle the complications introduced by AWGN at the receiver. To tackle additive noise better we propose a particle filter based waveform tracking solution which is discussed in the next chapter.

Chapter 4

Particle Filtering

4.1 Overview

In contrast to the hypothesis-testing approach in Chapter 3 for recovery of the amplifier input signal, this Chapter explores a Monte Carlo method for estimating the state of a dynamical system, observed in noise after a nonlinear transformation. The method specifies model dynamics and distributions associated with the state and observation noise, and propagates a swarm of particles that are samples, or realizations, from the approximate posterior p.d.f. for the state, given the observation sequence.

State estimation has a long history, and in the case of linear models with Gaussian noises, it's known that the Kalman filter supplies the MMSE estimate and the MAP estimate for the state. Our problem assumes a linear model for the signal dynamics, but it's driven by a discrete-alphabet sequence. Moreover, the observation equation is nonlinear (the output of the nonlinear amplifier). Thus, no simple procedure exists for updating the state's p.d.f. as observations arrive.

Particle filtering [10] has emerged in the last 15 years as a popular state estimation methodology in nonlinear or non-Gaussian situations, and we study its application for the central problem here, namely recovery of the amplifier input from noisy observations of the amplifier output.

A general dynamic system model is governed by a state-space model consisting of a transition equation and an observation equation, [20] .

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \\ \mathbf{z}_k &= \mathbf{h}(\mathbf{x}_k, \mathbf{v}_{k-1})\end{aligned}$$

where \mathbf{x} is a state vector, \mathbf{z}_k is an observation vector, and $\mathbf{w}_k, \mathbf{v}_k$ are state and observation noises respectively. Either or both of the equations can be nonlinear, and the noises may or may not be separable. The evolution of the state is modeled as a stochastic process and the goal is to estimate the state according to some criterion (MMSE, MAP). In the completely linear case the Kalman filter provides the MMSE estimate, and if the noise processes in addition are Gaussian, then the Kalman filter also produces the MAP state estimator. However in the nonlinear case one has to resort to several suboptimal algorithms like the extended Kalman filter (EKF), unscented Kalman filter (UKF), or particle filter, etc [17]. In our research we tried using the EKF but it displayed a propensity to diverge when operating in the saturation zone, since linearization about the current estimate often can give corrections in the wrong direction. We moved to the particle filter as the state estimation tool.

In the following sections, we first develop a suitable state-space model for the signal of interest. Then we review the operation of particle filtering and show its application to Saleh nonlinearity inversion and the results it yields. We first study its application on linear 4-PAM transmission for a single user, then extend to 4-PAM with nonlinearity, and eventually end up with two-user 16-APSK cases.

4.2 State Dynamics Modeling for the Particle Filter

Particle filtering requires a state-space formulation of the signal to be estimated. This involves approximating an FIR (pulse shaping) system with a low-order IIR system, which we next describe.

First recall, as in Chapter 2, the transmitted signal is produced by an FIR filter, after upsampling. The RRC filter is designed in *Matlab* using the command ‘*firrcos*’ with the following parameters.

$$\begin{aligned} \text{Rolloff} & : \beta = 0.25 \\ \text{Cutoff frequency} & : F_c = 0.125 \\ \text{Sampling frequency} & : F_s = 8 \end{aligned}$$

We need an IIR model that is not only stable and easy to implement, but also has dynamical properties similar to the FIR pulse shaping filter. An easy choice that can fit the requirement is a simple second-order discrete-time Butterworth filter. The filter has a normalized cutoff frequency of 1/8, consistent with 8 times over-sampling. Moreover, other advantages of a having a second-order filter is a simpler state model which can be easily analyzed, and the relation between the two states is very easily understood by their graphical depiction on the two dimensional plane. The behavior of the second-order model is a good indicator of how higher-order models will behave under similar conditions.

The transfer function of the obtained Butterworth filter is

$$H(z) = 0.03 \frac{z^2 + 2z + 1}{z^2 - 1.454z + 0.5741} \quad (4.1)$$

The next step is to obtain the state-space model. We can easily obtain the state space model from the transfer function in (4.1) using the methods described in [11], or simply Matlab's `tf2ss` command. However, the complication that arises with this is that the state model typically has a feed-forward term. The feed-forward term occurs whenever the number of zeros and the number of poles are equal, *i.e.*, the degree of the numerator and the denominator are the same in the transfer function. Such a transfer function is defined as a proper transfer function.

The reason why a proper function is not desired is that the feed-forward term in the state space model makes the output of the state space explicitly dependent on the process noise. As is evident from the transfer function in (4.1), the degree of the numerator and the denominator is equal and therefore we will have a feed-forward term present in our dynamic model. This can be remedied by choosing a strictly-proper transfer function in the z-domain, where a strictly-proper transfer function is one where the degree of the numerator is strictly less than the degree of the denominator [15],

$$\deg(D(z)) < \deg(N(z)) \quad (4.2)$$

where $D(z)$ and $N(z)$ in (4.2) correspond to the denominator and the numerator respectively.

We now need to modify the transfer function in (4.1) so that the degree of the numerator becomes less than that of the denominator. We display the pole-zero plot of the transfer function in Figure 4.1. We see that the Butterworth filter has two zeros located at $\omega = \pi$. We achieve a strictly-proper form in (4.1) by discarding one of the zeros and keeping both the poles exactly where they appear in the original transfer function as shown in Figure 4.1. However, the consequence of this is that the passband gain is not 1 anymore. To compensate for this we have to rescale the strictly-proper version of the filter.

The two frequency responses are shown in Figure 4.2 below and the strictly-proper transfer function is now

$$H(z) = 0.06 \times \frac{z + 1}{z^2 - 1.454z + 0.5741} \quad (4.3)$$

As is evident from Figure 4.2, the frequency response of the strictly-proper version of the filter matches almost perfectly to about 0.4π rad/sample which is greater than our signal cutoff frequency.

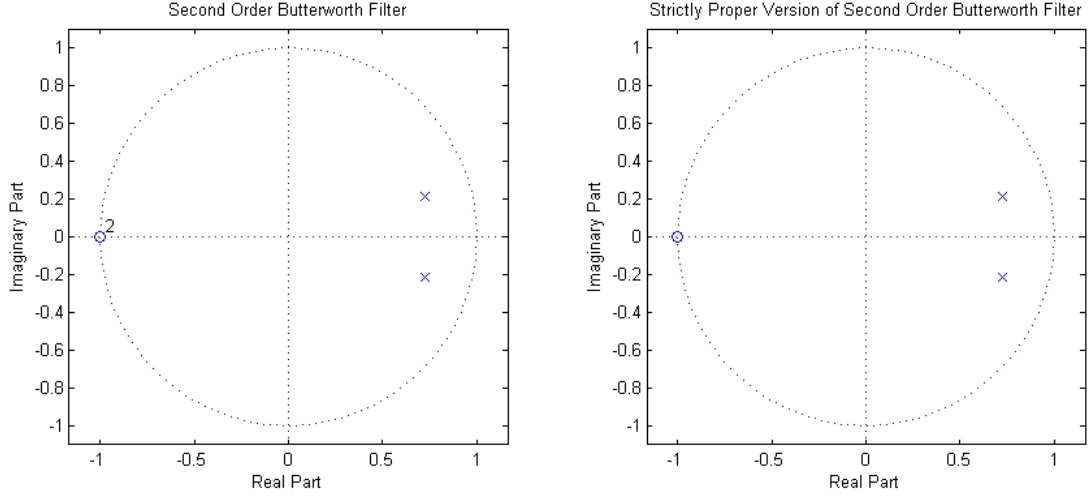


Figure 4.1: Pole-zero plots for original Butterworth filter and strictly-proper approximation

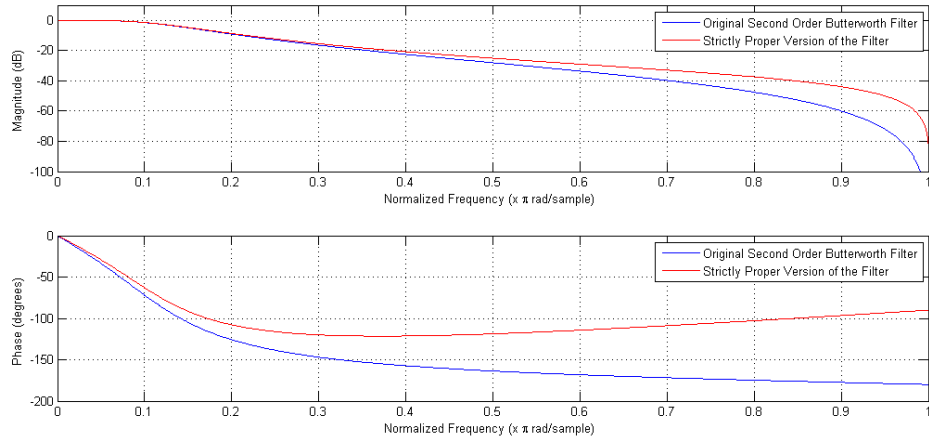


Figure 4.2: Frequency response plots for original Butterworth filter and strictly-proper approximation

We have devised an algorithm to automate this conversion process for converting to an N -th-order strictly-proper transfer function, where N corresponds to the degree of the denominator, and have included it in Appendix B.

Now that we have the transfer function in the appropriate form, the state dynamics model can be obtained using the methods mentioned in [11]. The transfer function is unique. However, the state representation is not and therefore making a good choice for the state model can help us simplify the problem and extract valuable information about the dynamics of the system. From the perspective of the communications model

we only have the observed noisy signal and we are using the state model to estimate the state of the received signal. In other words we have no influence on what the states are, however, we have the ability to peek into the state model and observe some of the states if we design our state model in observable form. For this reason, we desire that the output of our state model be such that it provide us insight about the states of the dynamic model. This is attained when we have the system represented in the observer-canonical form (OCF). This allows us to make the output of the model to be equal to one of the state which allows us to study the evolution of the states over time with ease. The conversion from CCF to OCF is also elaborated in Appendix B.

Now that we have defined the process of obtaining the state model and how to obtain the OCF, we now move on to define the algorithm of the particle filter in the next section.

4.3 Particle Filtering Algorithm

The particle filter is implemented at the receiver with the following components. The state-space model generated previously which serves as an approximate replica of the pulse shaping filter at the transmitter. At time $t = 0$ we produce a set of N_p Gaussian random variables that are independent and identically distributed (iid). The particles are propagated via the state model and by the virtue of an observable model the first state is also the output of the system.

Propagated particles are assigned a weight w_t proportional to the likelihood of the observation for a given particle. The weights of all the particles are summed and then each particle weight is divided by the sum to obtain a normalized measure of likelihood and so the weights w_t resemble probability mass. So-called *resampling* creates a new swarm of N_p particles corresponding to the observed normalized weights, or p.m.f.

For each particle the normalized weights represent the probability measure. The goal is to preserve particles with higher weights and propagate them along the state model and to suppress the progression of particles with low weights. We try to achieve this via importance sampling, thus preserving the particles with the high likelihoods and obliterating the particles with low likelihoods. So after importance sampling we may be left with a much smaller population of particles \tilde{N}_p . This leads to so called *particle impoverishment* where each time importance sampling is performed the particle population shrivels. The number of unique particles is greatly reduced below the N_p and the 2d swarm gets slimmer with each iteration. The particle filter will thus have fewer particles that are used in the a-posteriori probability estimation, and after a few iterations there might be just one surviving particle. To counter this we resample. Resampling is important because it prevents particle impoverishment [21]. The number of particles propagating at each time is equal to the initial number of particles. Each particle is assigned a weight by comparing their output to the

observed output. The particles with higher weights survive and the particles with negligible weights are stopped from propagating. However, there is one important difference that the particles that wither away in the weighting process are replaced by resampling from a uniform distribution such that the particles with significant weight are now replicated. The number of times a particle is replicated depends on the of particles discarded and also on the weight of the partial that is replicated. This ensures that the same number of particles are present at each time iteration. This is analogous and is referenced in the literature as the *survival of the fittest*.

We now present the particle filtering algorithm in the form of a flowchart in the left portion of Figure 4.3. Moreover, the pseudo-code provided below as Algorithm 1 is a modified version of pseudo-code provided in [21], and summarizes the iterative behavior of the particle filter algorithm, often known as the *bootstrap particle filter*. We also provide the slightly modified pseudo-code of the resampling algorithm as discussed in [21] as Algorithm 2.

Algorithm 1 Algorithm for Bootstrap or Sequential importance sampling (SIR) particle filter [21]

$[\{x_k^i\}_{i=1}^{N_p}] = \text{SIR}[\{x_{k-1}^i\}_{i=1}^{N_p}, z_k]$.

- FOR $i=1 : N$
 - Draw $x_k^i \sim p(x_k | x_{k-1}^i)$
 - Calculate $\tilde{w}_k^i = p(z_k | x_k^i)$
 - END FOR
 - FOR $i=1 : N$
 - Normalize $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N_p} \tilde{w}_k^i}$
 - END FOR
 - **Resample** using the resampling algorithm provided below.
-

To apply this to our problem, we create an initial swarm of particles (the actual distribution does not matter as it vanishes over time). These serve as the initial states at time $t = 0$ and are propagated using the state model to time $t = 1$. The states at time $t = 1$ are then applied the appropriate backoff to be comparable power to the sequence that was generated at the transmitter.

Meanwhile the true state process has been generated using pulse shaping, backoff as desired, and subjecting it to the Saleh model, then adding the desired amount of noise. To review, this is represented by

$$g(\tilde{Y}^i) = \frac{\alpha_a |\tilde{Y}^i|}{1 + \beta_a |\tilde{Y}^i|} \exp(j\angle \tilde{Y}^i) \quad (4.4)$$

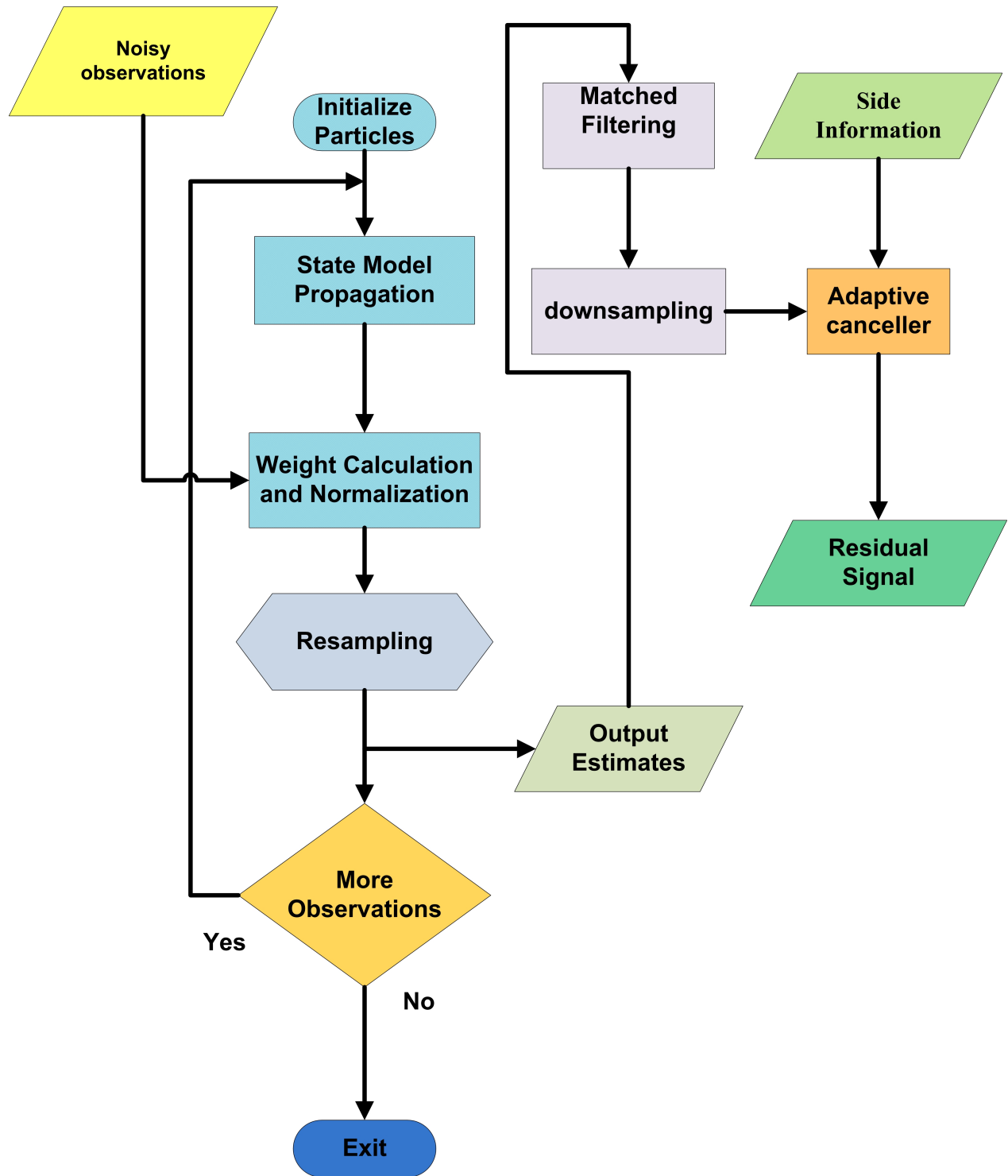


Figure 4.3: Flowchart of particle filter based inverter algorithm

Algorithm 2 Resampling Algorithm [21]

$[\{x_k^{j*}, w_k^j\}_{j=1}^{N_p}] = \text{RESAMPLE}[\{x_k^i, w_k^i\}_{i=1}^{N_p}]$.

- Initialize $c_1 = w_k^1$
 - FOR $i = 2 : N$
 - Create CDF: $c_i = c_{i-1} + w_k^i$
 - END FOR
 - Draw a random sample from $u_1 \sim \mathcal{U}[0, N_p]$
 - FOR $j = 1 : N$
 - Move along the CDF: $u_j = u_1 + \frac{j-1}{N_p}$
 - WHILE $u_j > c_i$
 - * $i = i + 1$
 - END WHILE
 - Assign sample: $x_k^{j*} = x_k^i$
 - Assign weight: $w_k^j = \frac{1}{N_p}$
 - END FOR
-

For each particle, the output of the Saleh model is generated, and its weight is found using the likelihood function based on a Gaussian distribution with appropriate noise variance. We simplify the weighting function by removing the normalizing factor in the p.d.f. The weighting procedure is shown below.

$$\tilde{w}(\tilde{Y}^i) = \exp\left(-\frac{(|Z_n - g(\tilde{Y}^i)|^2)}{\sigma^2}\right) \quad (4.5)$$

$$w(\tilde{Y}^i) = \frac{\tilde{w}(\tilde{Y}^i)}{\sum_{i=1}^N \tilde{w}(\tilde{Y}^i)} \quad (4.6)$$

With these particulars, the particle filtering algorithm proceeds as outlined above.

4.4 Linear Channel Testing

Before we tackle the nonlinear estimation problem using the particle filter, we try to get an insight into the functionality and performance of the particle filter on a linear channel, in addition to confirming the correctness of the software. There is no need for particle filtering in this situation as matched filtering followed by echo cancellation is optimum, but it serves as a good test of particle filtering.

To create a linear channel within the Saleh model framework we merely set $\beta_a = 0$ and $\alpha_a = 1$ in (2.5). This yields the simple model depicted in Figure 4.4 and also shown below mathematically.

$$\tilde{S} = S + n \quad (4.7)$$

$$A(u) = \alpha_a u \quad (4.8)$$

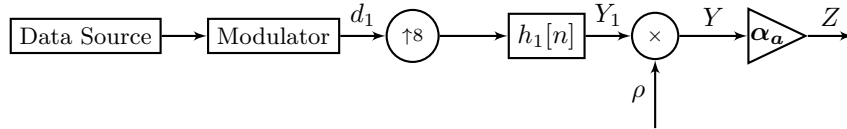


Figure 4.4: Transmitter set up for linearized channel

We initialize the filter with N_p Gaussian random particles that serve as the initial state ($t = 0$) and are used to estimate the state at time $t = 1$. The N_p particles are propagated through the dynamic model and the estimation is carried out according to the algorithm given in Section 4.3. This simple setup allows us to make several important observations regarding the particle filter's behavior. We study the behavior of the particle filter in the linearized case by varying the different parameters; namely, the SNR at the receiver, and the number of particles. The input backoff at the satellite relay is meaningless in the linearized channel as there is no notion of saturation. In addition we also briefly study the effect of having a higher-order state model to approximate the pulse shaping filter. To achieve deeper insight into the functionality of the particle

filter we start with a simple pulse amplitude modulated (PAM) data and then a simple 2-D constellation and eventually transition to the constellation of interest, 16-APSK.

4.4.1 PAM

We start with 4-level pulse amplitude modulation (PAM) as it provides real (1-D) signals. We define our alphabet $\mathcal{C} = \{-3, -1, 1, 3\}$. The data sequence is upsampled, filtered by the pulse shaping filter, and scaled. Z is corrupted by zero-mean AWGN at the receiver. Z_n is now passed onto the particle filter where the filter tries to estimate the actual signal state. The block diagram of the receiver is shown below in Figure 4.5. We now study the effects of the various parameters mentioned earlier, starting with SNR.

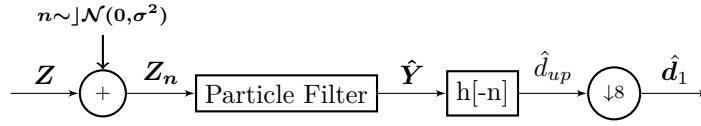


Figure 4.5: Single-user receiver with particle filter

4.4.1.1 The Effect of SNR

To understand this effect we kept N_p fixed to a modest value of 100. To reduce variation that arises from different realizations of data, we keep the same realization of the data sequence d_1 and vary the SNR by changing the power of the additive noise. In Figure 4.6 we show the tracking results as a function of SNR. It is observed that as SNR of the received signal gets higher, the tracking results get better. This is reemphasized in Figure 4.7, where the mean squared error between the true waveform Y and the estimated waveform \hat{Y} is plotted over a wider range of SNR. So we can conclude, as expected that as SNR gets higher the particle filter tracks better. In Figure 4.7 we note that tracking MSE improves as $1/\text{SNR}$, exactly as expected for a Kalman filter. However, if the noise is negligible then the particles immediately lose diversity and therefore particle filter may not be the best option. In high SNR situations it's best to use an inflated noise level in the likelihood calculation to avoid collapse of the particle swarm to one or a few particles. This provides a nice segue into our next section which analyzes the influence of particle population on tracking.

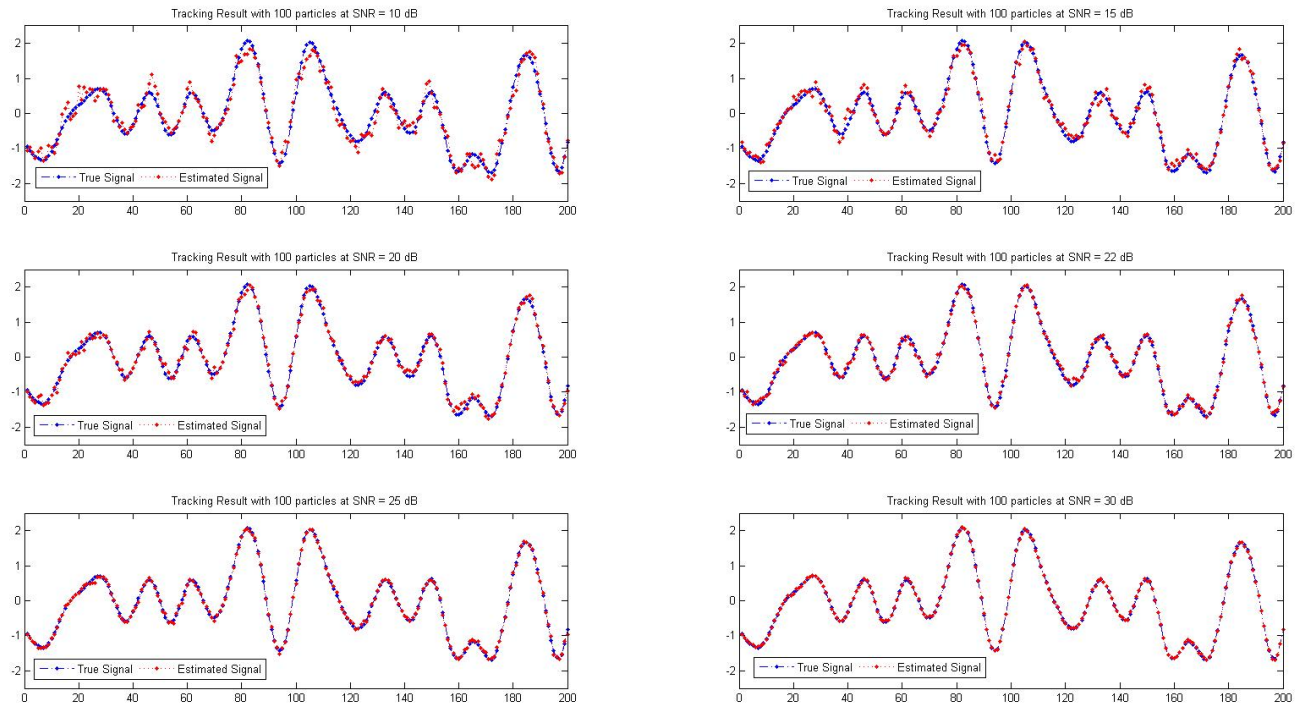


Figure 4.6: Tracking results with various SNR's with 100 particles.

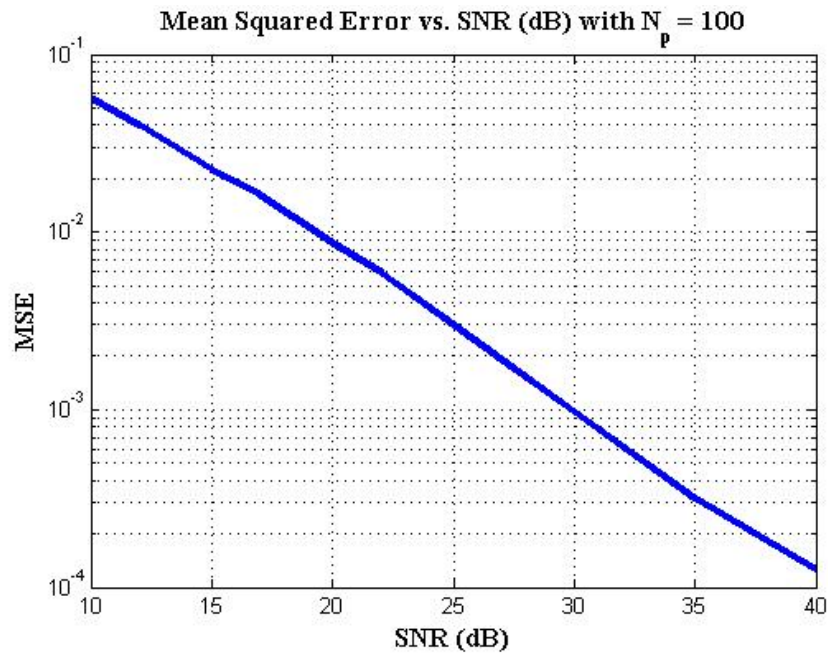


Figure 4.7: MSE vs. SNR(dB)

4.4.1.2 The Number of Particles

We now study the implications of the number of particles on the performance of the particle filter. As was claimed previously, by making the number of particles very large, the Monte Carlo approximation of the posterior p.d.f. gets closer to the true description of the posterior p.d.f., and the particle filter approaches the optimal Bayesian estimator, which in the linear case is the Kalman filter. For this analysis we fix the SNR to 15 dB and once again use the same realization of the transmitted signal to have a better understanding of the effect of number of particles. We use the following set N_p of "number of particles" to perform this analysis.

$$N_p \in \{10, 20, 50, 100, 200, 500, 1000, 2000\}$$

The results of the simulation are shown below. We can see that the plots of the estimated signal don't improve much as the number of particles increase beyond 50. Furthermore, we see that there is a performance floor, *i.e.* keeping all other parameters constant, increasing the number of particles doesn't yield an improvement in the performance.

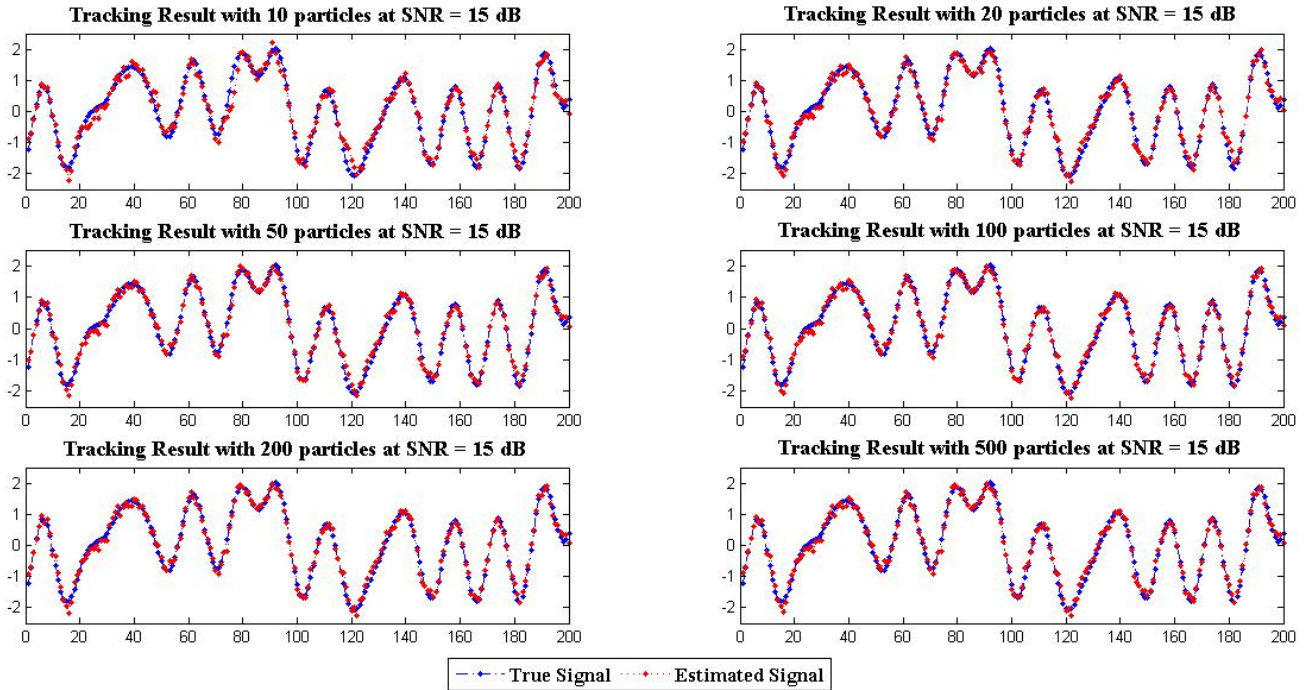


Figure 4.8: Tracking results with varying number of particles at an SNR of 15 dB.

Figure 4.9 recapitulates this phenomenon and shows MSE undergoes diminishing returns and soon the MSE encounters a floor. This floor is a consequence of the additive noise of the channel; even the optimal Kalman filter for this problem would have this same floor.

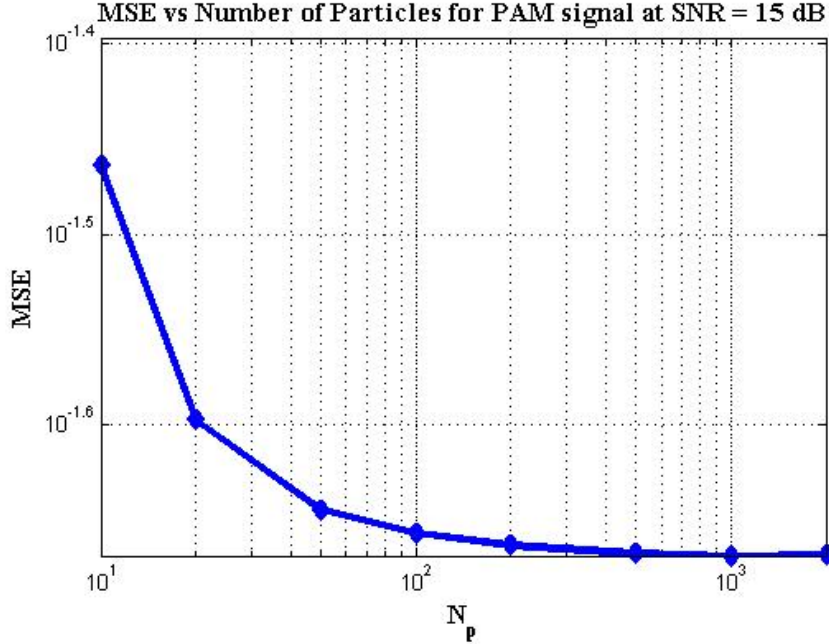


Figure 4.9: MSE vs. number of particles

4.4.1.3 Effect of Higher-Order State Models

We now repeat the experiments above for a higher-order realization of the state model. The model is still a modified Butterworth filter obtained via Algorithm B but with higher order. For simplicity we choose a fourth-order model. A higher-order model has more correlated states and a greater ‘inertia’, and generally can more closely match the dynamics of the actual FIR signal. However, intuition suggests there should be more particles required to have sample diversity and avoid particle impoverishment. The results were compared for same SNR value for both second and fourth-order filters with strictly-proper transfer functions and the results are reflected in Figure 4.10. The number of particles required are significantly higher for fourth-order to achieve the same MSE. The performance reaches a floor at around 150 particles and then the fourth-order achieves better performance for the same number of particles. However, both the systems seem to hit a floor beyond which their performance is not improved any more when the number of particles is increased. Moreover, to achieve a relatively small marginal gain reduction in MSE the additional computational cost

incurred while operating a higher-order model coupled with a large number of particles might not be worth the cost.

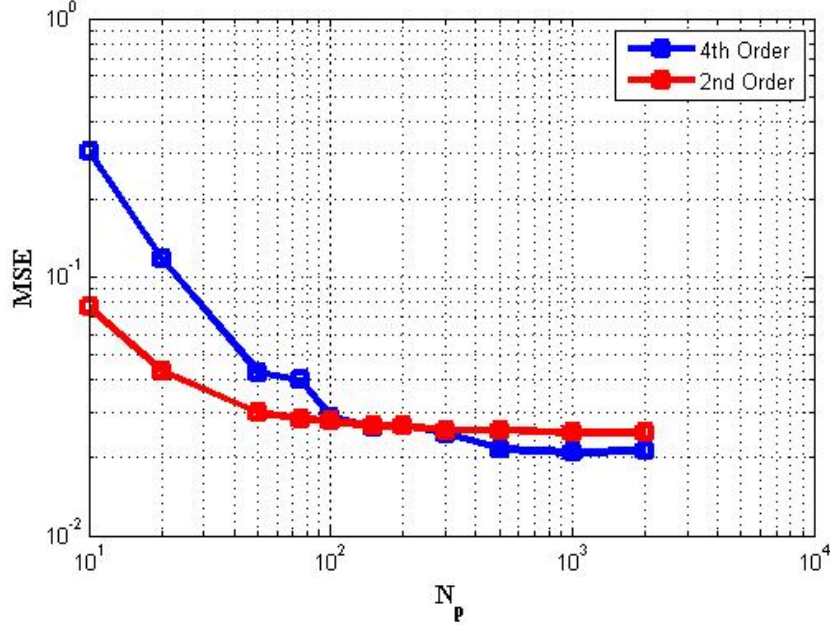


Figure 4.10: MSE vs. N_p for 2^{nd} and 4^{th} -order state models

The PAM study of the particle filter, though simple and performed on a linear channel, provides valuable insight about the functionality of the particle filter. We have empirical evidence now regarding the performance as a function of various parameters. This 4-PAM exercise confirms the correctness of our particle filter algorithm. Next, we test a constellation with complex symbols. A simple choice is QPSK because of easiness in implementation and testing.

4.4.2 QPSK

We now examine the performance of the particle filter with a complex constellation, still on a linear channel. The state model used is the second-order model in (4.3). Everything else, except for the alphabet, at the transmitter stays the same as in the linear channel with PAM described previously. At the receiver, we now have zero-mean complex AWGN. The particle filter is implemented the same way as in the case of PAM. The only change is in the weighting function which is now modified to reflect the complex sequence. The complex sequence is a proper Gaussian and we use σ^2 instead of $2\sigma^2$ in the denominator of the argument of the exponential in (4.5). We start again with the study of SNR.

4.4.2.1 The Effect of SNR

For a range of SNR's we present below, in Figures 4.11 and 4.12, the tracking results in the complex plane for a small segment of the QPSK locus in the bidirectional case. As opposed to the PAM signal, where we had 1-D noise, the corruption due to noise is much more noticeable as the two noise components are independent of each other and can deteriorate the performance of the particle filter. The tracking is not smooth at low SNRs, however, the locus of the estimate gets smoother and the tracking result improves with increasing SNR. We don't show the tracking results of single user case, but the plot provided in Figure 4.13 shows that the MSE monotonically decreases with increasing SNR, in both the single-user case and the bidirectional case. In fact, we obtain almost identical plots in Figure 4.13. This is because in the linear channel there is no notion of saturation.

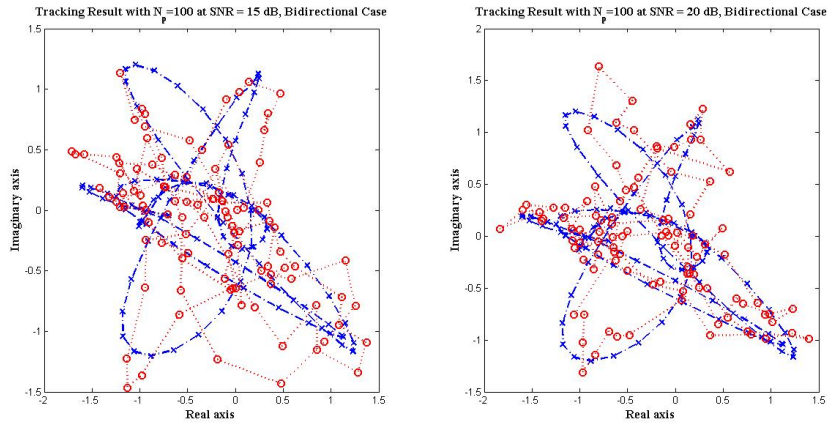


Figure 4.11: Tracking results for QPSK for SNRs 15 dB (left) and 20 dB (right), $N_p=100$, Bidirectional case

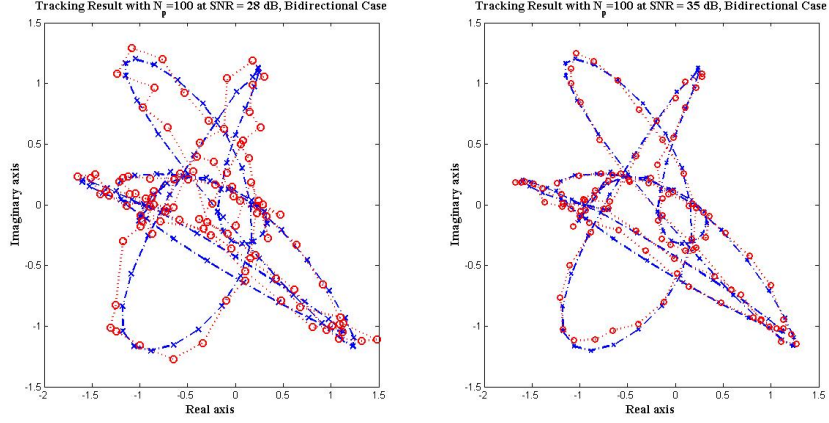


Figure 4.12: Tracking results for QPSK for SNRs 28 dB (left) and 35 dB (right), $N_p=100$, Bidirectional case

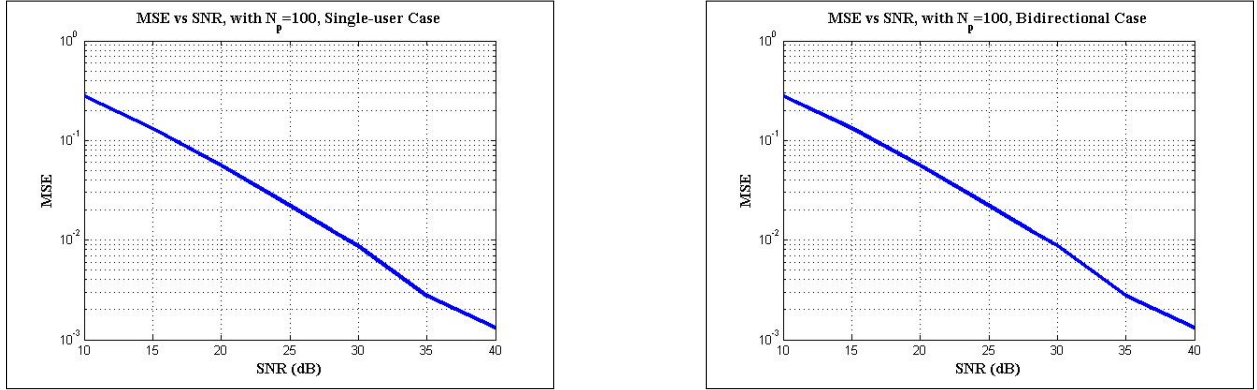


Figure 4.13: MSE vs SNR for Single-user (left) and Bidirectional (right) cases with $N_p = 100$

For the bidirectional case, we now present the constellations obtained at the output of the adaptive canceller. We observe from Figures 4.14 and 4.15 that the constellations' qualities increase with increasing SNR and the decision zones are well defined even in the low SNR case. This exercise also confirms the successful functionality of the particle filter in the case of a complex constellation. We now study the effect of number of particles in the complex constellation case.

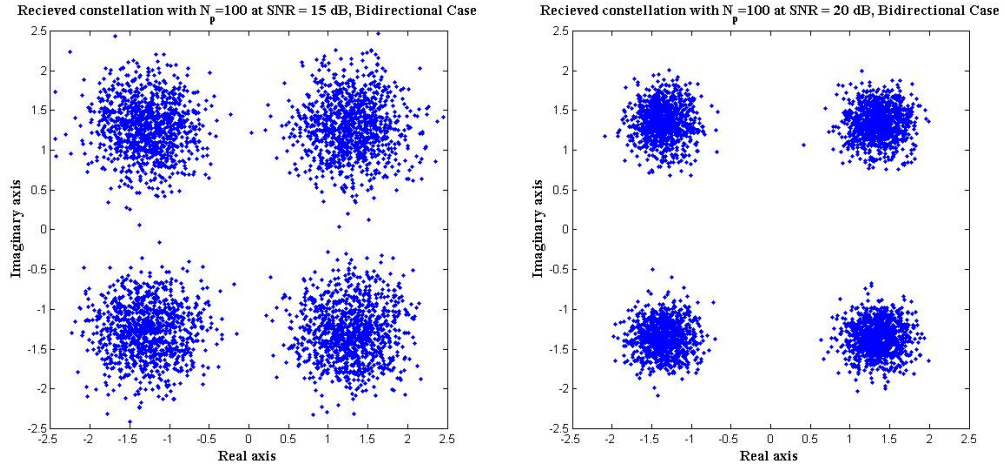


Figure 4.14: Constellations for QPSK for SNRs 15 dB (left) and 20 dB (right), $N_p=100$, Bidirectional case

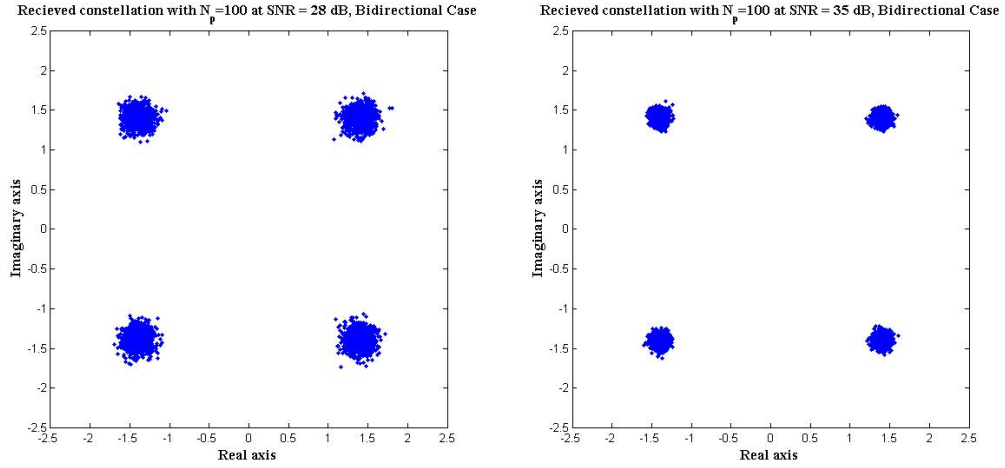


Figure 4.15: Constellations for QPSK for SNRs 28 dB (left) and 35 dB (right), $N_p=100$, Bidirectional case

4.4.2.2 The Number of Particles

Intuitively, in the two-dimensional case we need a larger particle population to maintain a similar performance as PAM. This is because with two dimensions the domain of the signal increases and therefore the number of particles must be increased in order to get a good coverage of the 2-D space. However, again we experience

the phenomenon of performance saturation where by increasing the number of particles beyond 200 there is no significant improvement in tracking error. The results for the bidirectional case are shown below in Figure 4.16.

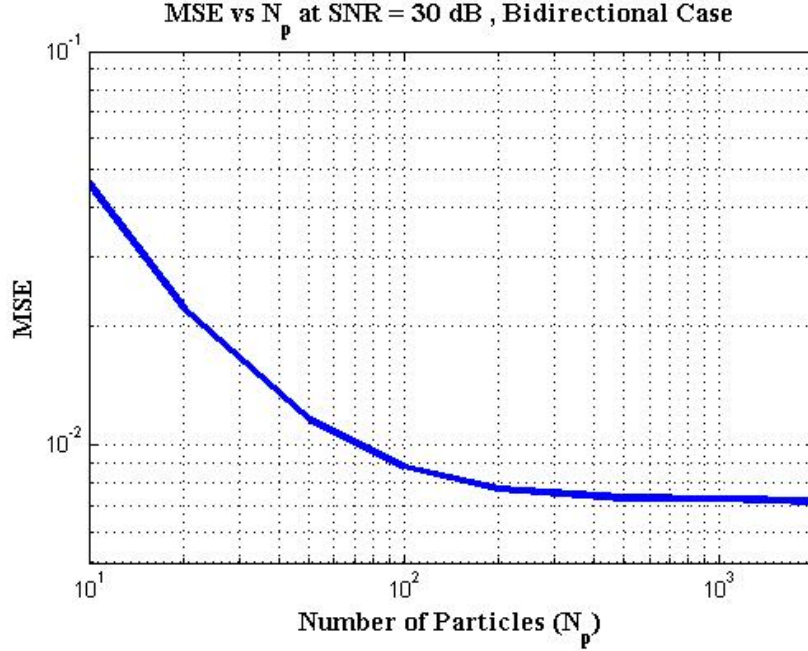


Figure 4.16: MSE vs number of particles at SNR= 30 dB, Bidirectional case

4.4.3 16-APSK

Now that we have established the performance of the particle filter for quadrature modulation, we now focus our attention on the 16-APSK constellation. We analyze the performance for the same set of parameters. The performance should be similar to QPSK in the linear case, where we encounter a monotonic decay in MSE with increasing SNR. However, we expect have an increased probability of symbol error because the symbols in the constellation are much closer to each other compared to QPSK. We first present the tracking results in Figures 4.17 and 4.18. As can be seen the tracking results get better with increased SNR. We highlight this trend in Figure 4.19 below.

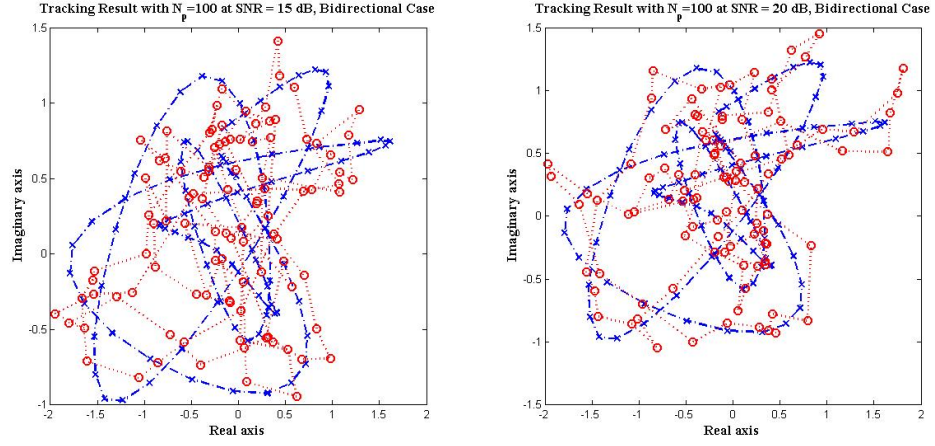


Figure 4.17: Tracking results for 16-APSK, for SNRs 15 dB (left) and 20 dB (right), $N_p = 100$, Bidirectional case

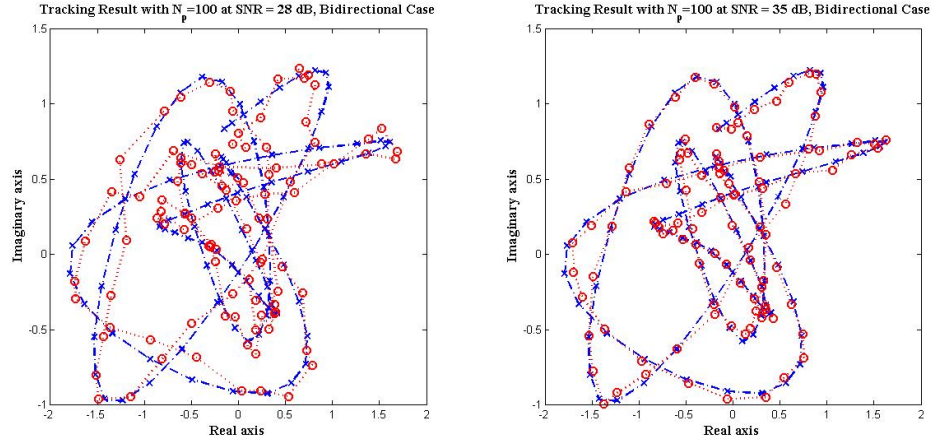


Figure 4.18: Tracking results for 16-APSK, for SNRs 28 dB (left) and 35 dB (right), $N_p = 100$, Bidirectional case

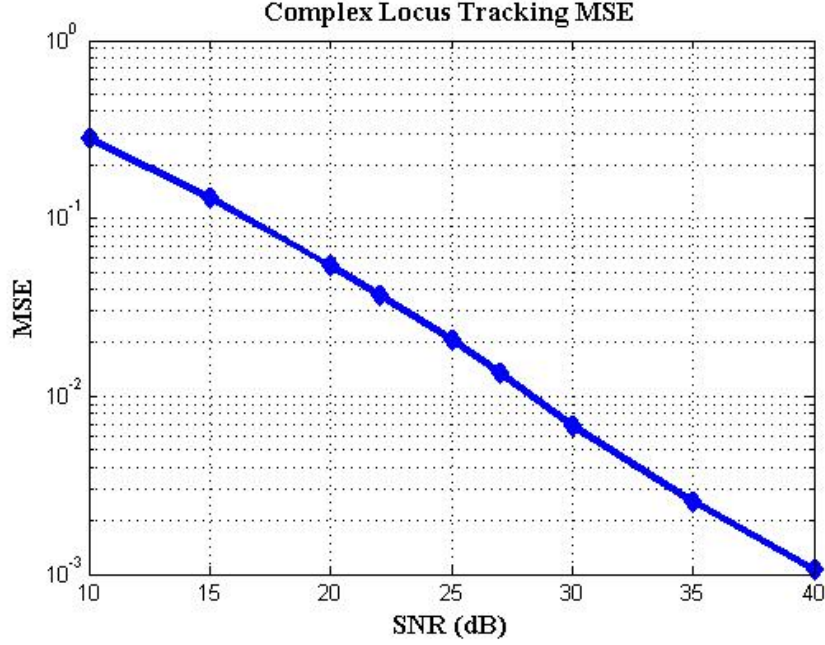


Figure 4.19: MSE vs SNR for 100 particles, 16-APSK, linear channel

Having looked at the tracking results, we now look at the constellations that are obtained at the output of the adaptive canceller. Figures 4.20 and 4.21 are in agreement with our expectation of the constellation getting deteriorated with low SNR because of the presence of more symbols and their proximity to each other. However, the constellations do get better with SNR and the probability of symbol error goes down with increasing SNR.

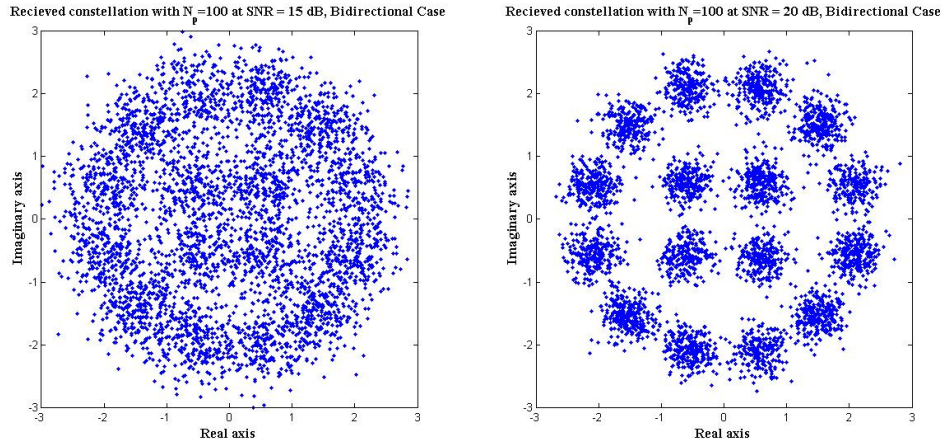


Figure 4.20: Constellations for 16-APSK, for SNRs 15 dB (left) and 20 dB (right), $N_p = 100$, Bidirectional case

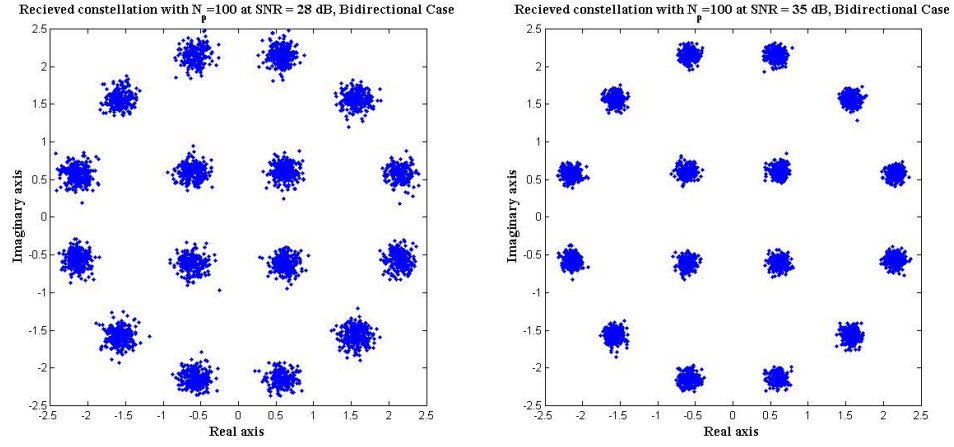


Figure 4.21: Constellations for 16-APSK, for SNRs 28 dB (left) and 35 dB (right), $N_p = 100$, Bidirectional case

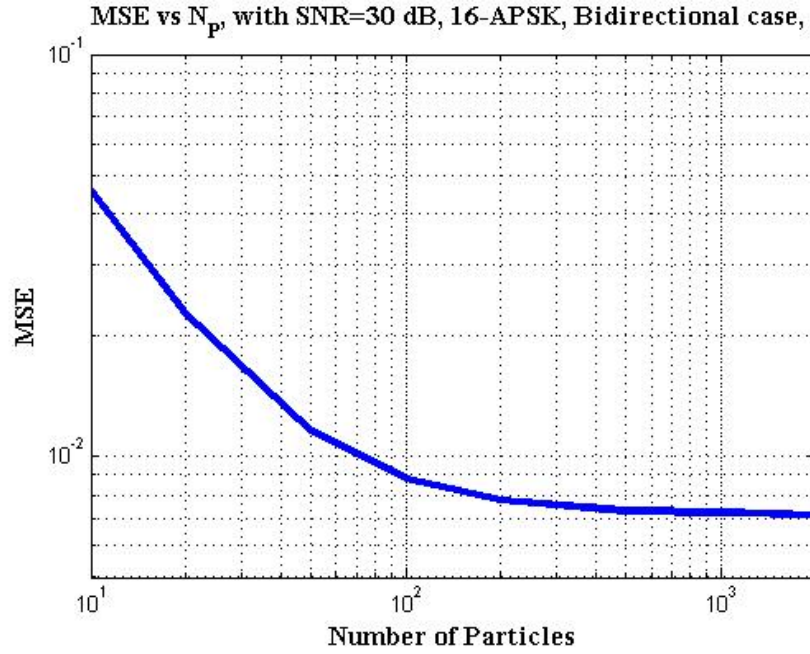


Figure 4.22: MSE vs N_p , with SNR = 30 dB, Bidirectional case

In the 16-APSK case, the number of particles has a similar effect on the tracking MSE as was observed in the case of QPSK and 4-level PAM. This is illustrated graphically in Figure 4.22. In summary, for all cases

on a linear channel, particle filtering with 50-100 particles seems a good design choice.

4.5 Nonlinear Channel

We now return to the core of the problem with having a better understanding of the behavior of the particle filter. The analysis that we have done so far presents valuable insight in using the particle filter to invert the actual Saleh nonlinearities introduced at the satellite relay. We first treat the special case of a single-user transmitting and a single-user receiving in a nonlinear setup. We now restore the original Saleh parameters given as following

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha_a & \alpha_\phi \\ \beta_a & \beta_\phi \end{bmatrix} = \begin{bmatrix} 2 & 4.0033 \\ 1 & 9.1 \end{bmatrix}$$

We again study the effect of the SNR, number of particles, and also another important parameter that was absent in the case of the linearized channel, namely the backoff. Based on the analysis so far, we have a fairly good understanding of the effects of SNR and the size of particle swarm. However, we need to see the influence of backoff on the performance of particle filter as it is the parameter that dictates the severity of the nonlinearity.

4.5.1 Effect of backoff

Typically the signals are backed off significantly to avoid the nonlinear distortion introduced by the HPA, however, a significant loss of power occurs when the signals are backed off. The more the signal is backed off the closer we are to the linear channel but our goal is to see if we can be more power efficient by operating in the nonlinear zone of the HPA. For this part of the analysis we fix the SNR to 35 dB, a fairly high number, to minimize the artifacts of additive noise. Furthermore, we also start with a fairly high number of particles as we are trying to minimize the consequences introduced by the dearth of particles. We vary the backoff from quasi-linear operation to highly nonlinear. We start with a high backoff to see if the particle filter is performing as it would in a linear setup and there are no unexpected observations made. This will allow us to focus on the effect of SNR. So we start with a backoff of -12 dB and slowly work our way into the nonlinear zone.

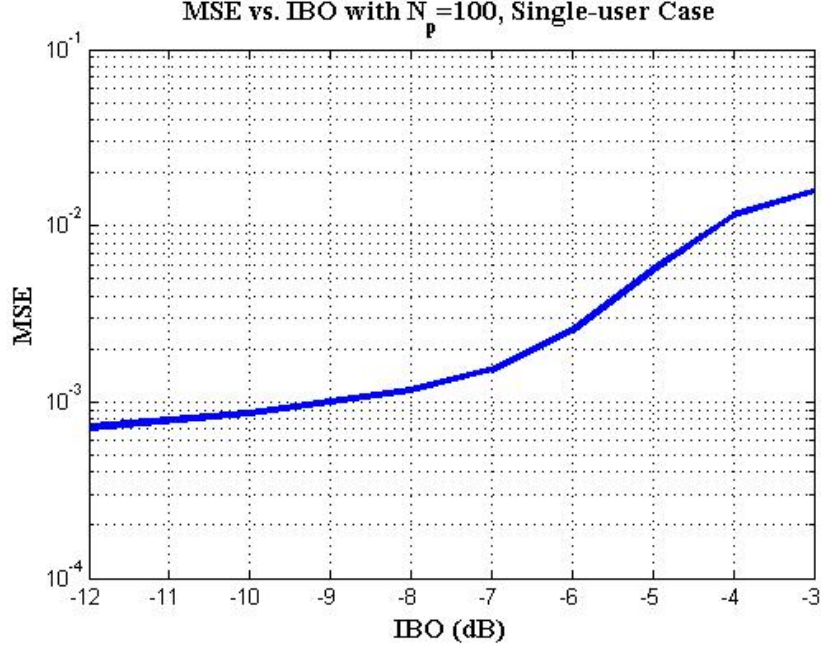


Figure 4.23: MSE vs. IBO with $N_p = 100$ and SNR = 35 dB, Single-user case

After going through the particle filter inverter, the signal is quasi-linear and therefore is passed through the matched filter. The matched filter output is downsampled and some of the constellations recovered at SNR = 35 dB with 100 particles at various outputs are presented below. The constellations do not show any nonlinear distortion for larger backoffs; however, when backoff is gradually increased we see that the outer ring of the constellation undergoes distortion which is radial in nature. This is a consequence of gain compression which becomes more substantial as the input backoff gets smaller. This leads to erroneous state estimation and thus we see an outward radial drift in the outer ring's constellation points.

For the backoffs $\in \{-12, -10, -8, -7, -6, -5, -4, -3\}$, we plot their corresponding SER. We again use the centroid method explained earlier to calculate the SER. Note the scale difference in Figures 4.24 and 4.25. Also note that each constellation in figure has a different energy which is inversely proportional to the backoff level. This figure demonstrates the trade off between power and distortion. The signals in Figure 4.25 undergo a more severe distortion, however, the decision boundaries seem less vague. To quantify this observation we use the centroid based approach to attain the SER for all the eight cases and the SER plot is shown in Figure 4.26.

The constellation plots show interesting radially-distributed behavior, with largest scatter for the outer ring points, as should be expected. Decisions based on this constellation however would be high quality, especially given the nature of the scatter. The Voronoi zones previously depicted in Figure 2.14 are still going

to be valid despite the distortion as the constellations points can be still fairly easily classified into their respective zones. Even in the case when backoff goes as low as -3 dB, we see that the Voronoi decision boundaries as discussed in Chapter 2 are still valid.

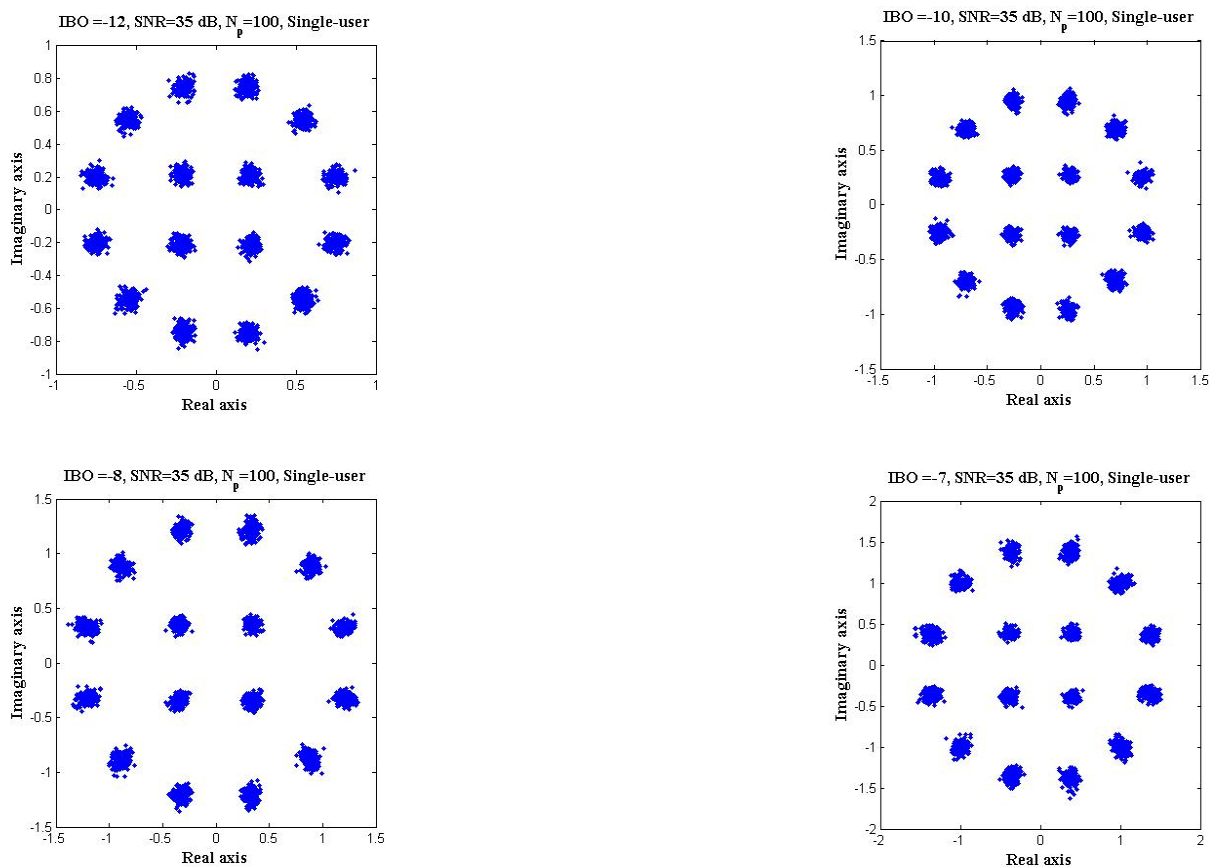


Figure 4.24: Constellation plots for SNR = 35 dB, $N_p=100$
 IBO = -12 dB (top-left), -10 dB (top-right), -8 dB (bottom-left), -7 dB (bottom-right)

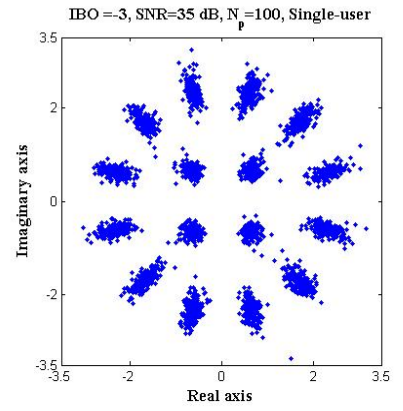
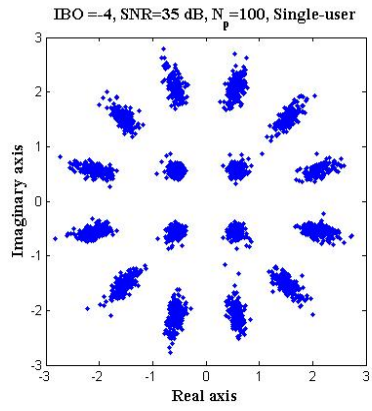
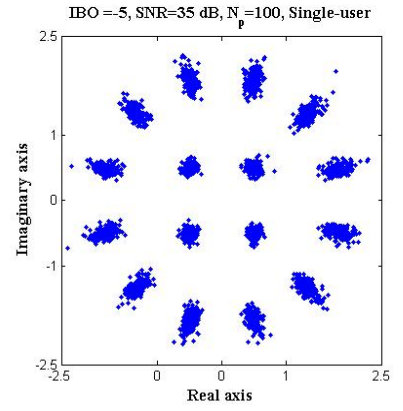
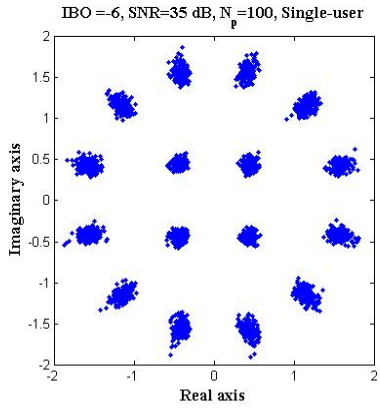


Figure 4.25: Constellation plots for $\text{SNR} = 35$ dB, $N_p=100$
 $\text{IBO} = -6$ dB (top-left), -5 dB (top-right), -4 dB (bottom-left), -3 dB (bottom-right)

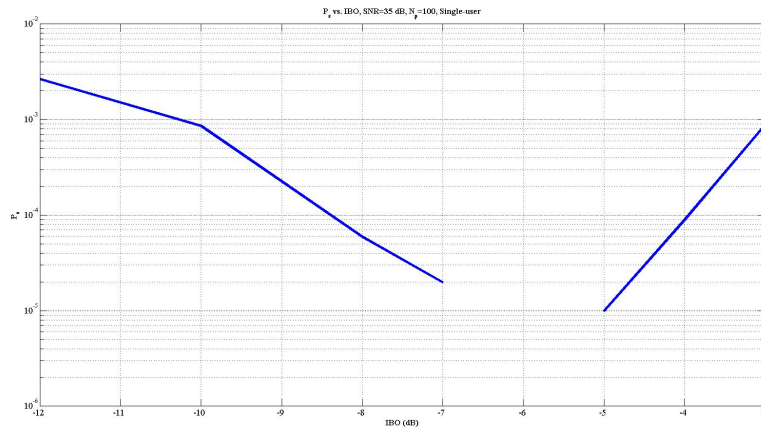


Figure 4.26: P_s vs. IBO with $N_p = 100$ and $\text{SNR} = 35$ dB, Single-user case.

4.5.2 Effect of SNR

Next we study the effect of SNR, still for a single user to better understand the performance and behavior of the particle filter over the nonlinear channel. We now check the robustness of the particle filter by subjecting it to perform under various levels of AWGN. The backoff is also kept constant at 4 dB which is about 40% of the saturated signal power level. We show some tracking results which are followed by the constellations of the data produced at the receiver. We expect the performance to increase with the SNR as there is more certainty associated with certain particles. Although high SNR has the advantage of giving more confidence in the measurement, it can also lead to the loss of diversity among the particles as all the particles in a very focused region tend to carry all the weight. This makes the particle filter susceptible to losing track. This can be countered by artificially inflating the measurement noise by scaling up the respective variance. We now demonstrate the tracking results and follow them by plotting the MSE as a function of SNR. It seems by inspection of Figure 4.27 that there is a performance floor. This is most likely a consequence of the number of particles being fixed and therefore we claim that the performance can be improved by increasing the number of particles.

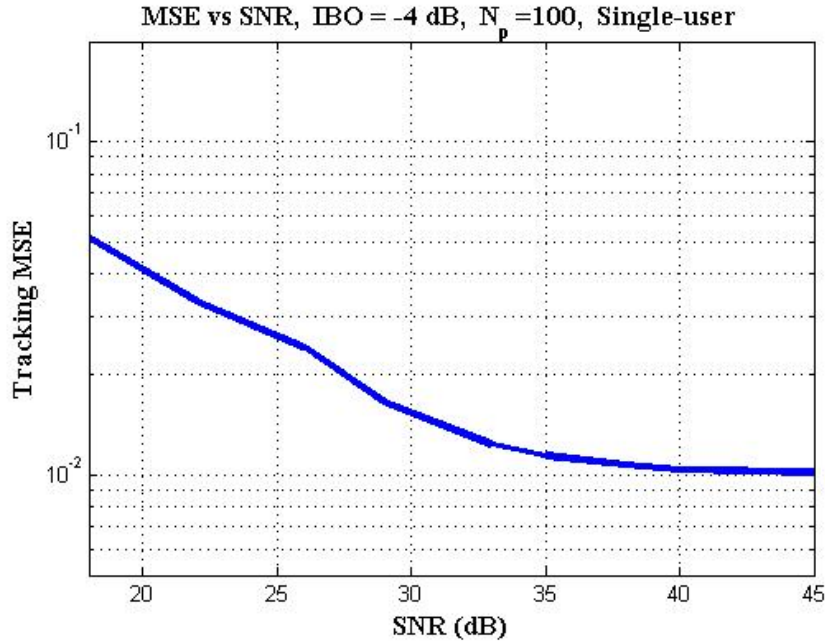


Figure 4.27: MSE vs. SNR for 16-APSK, IBO = -4 dB, $N_p = 100$

4.5.3 Effect of number of particles

In general an increased number of particles lead to a better estimation of the posterior density. However, as we saw in the linear case that the MSE experiences diminishing returns with respect to the number of particles. In the linear case, the MSE decays by increasing the number of particles, however, after the number of particles have reached about 50 we don't see any major improvement in tracking resulting from increasing the number of particles. This was because the posterior pdf was unimodal and it was completely characterized with about 50 particles and increasing the number of particles beyond that did give any improvement in the posterior pdf of the state. However, with increased drive level we can have a multimodal distribution. In our particular case the Saleh model is a non-invertible nonlinearity and therefore for a possible value of output there are two possible input values. This fact was discussed in great detail in section 4.4.1.2 and also in the MAP based inversion algorithm discussed in Chapter 3. The distribution demonstrates more strong bimodal behavior at low backoff levels as the signal tends to evolve more frequently into the saturation zone as shown in Figure.3.2. Therefore intuitively we need more particles to get a more accurate posteriori distribution. To test this hypothesis we give the particle filter a specific SNR and an input backoff and run the algorithm for different number of particles. The results obtained for tracking mean-squared-error by varying the number of particles is presented in figure are reflected in Figure 4.28. We notice that the MSE again observes a floor between 100 and 200 particles. This time, *i.e.* in the case of nonlinear waveform, the floor is reached at a higher number of particles since more particles are needed to represent the a-posteriori pdf.

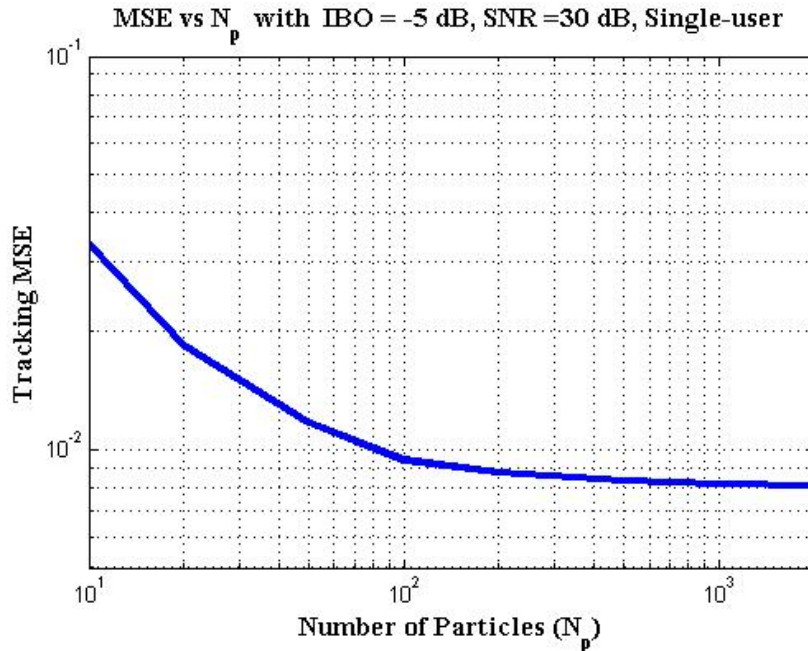


Figure 4.28: MSE vs. N_p for 16-APSK, with SNR= 30 dB, IBO= -5 dB, Single-user case

We also present the symbol error rate as a function of the particle population in Figure 4.29. The SER initially decreases with the number of particles but after a 100 particles the SER becomes zero.

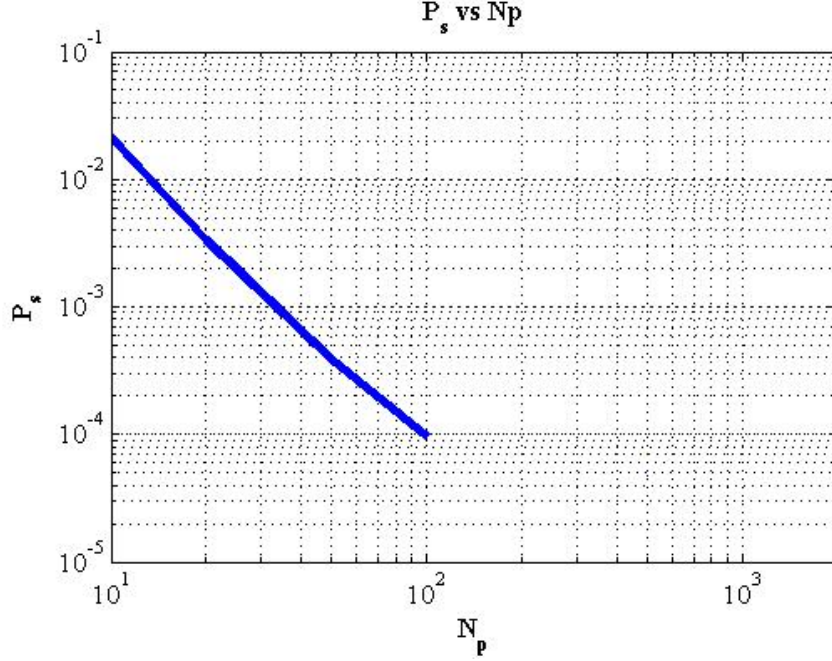


Figure 4.29: P_s vs. N_p for 16-APSK, with SNR =30 dB, IBO =-5 dB, Single-user case

4.5.4 Bidirectional transmission

Up to this point we have built the discussion of the particle filter in order to justify its use and get intuition and assurance that it has the potential to work as we have proposed. We have shown previously the performance of single user results over linear and nonlinear channels and inferred important information about the performance in tracking the transmitted waveform. Our results of waveform tracking in the case of nonlinear channel study for one user were encouraging. We now take the next step of having two users transmit information to the relay simultaneously as per Figure. 2.1. At the relay we put the aggregate signal into the HPA and then transmit the aggregate amplified signal back to each terminal.

We now study the tracking results using the particle filter for this aggregate signal. The difference from the single user case is that the signal now has a higher peak to average power and thus stays in the saturation or ambiguity zone, as shown in Figure 3.2, longer relative to the single-user problem addressed earlier. The amplifier for the same input backoff is therefore going to be driven harder and the output will undergo

a relatively more severe compression and distortion as compared to the single user case. Therefore the challenge for the particle filter is more difficult in the sense that it has to be able to track a more severely distorted signal. We propose that the number of particles required for tracking the aggregate signal is going to be higher. Moreover, we expect that we might have to backoff to a larger backoff in order to achieve the same performance as the single user case. We begin the analysis by studying the effect of SNR, backoff, and the number of particles propagated in that order. Then we look at the implications of these parameters on the error rate.

4.5.5 Effect of SNR

As we saw in the implementation of particle filter for the linear channel that as noise power gets smaller and smaller, the estimate of the particle filter gets better and better as the next observation is actually closer to where the particle filter predicts its presence. However, when we have too high an SNR the particles experience impoverishment as was explained earlier. This can cause loss of track as the population density of the particles get concentrated and they don't provide a very good coverage. Also in case of lost track the filter will take longer to regain track because the noise variance used in the weighting function is small and therefore might take several samples before the particle filter gets back on track. Moreover, this loss of track could happen again and therefore the overall tracking performance deteriorates. We therefore artificially inflate the noise variance used in the weighting function in (4.5) to calculate the likelihood of each new state. This dispersion of particles leads to better tracking. In Figure 4.30 we present tracking results for a segment of locus at different SNRs. It conforms with our previous discussion that increased SNR implies low MSE.

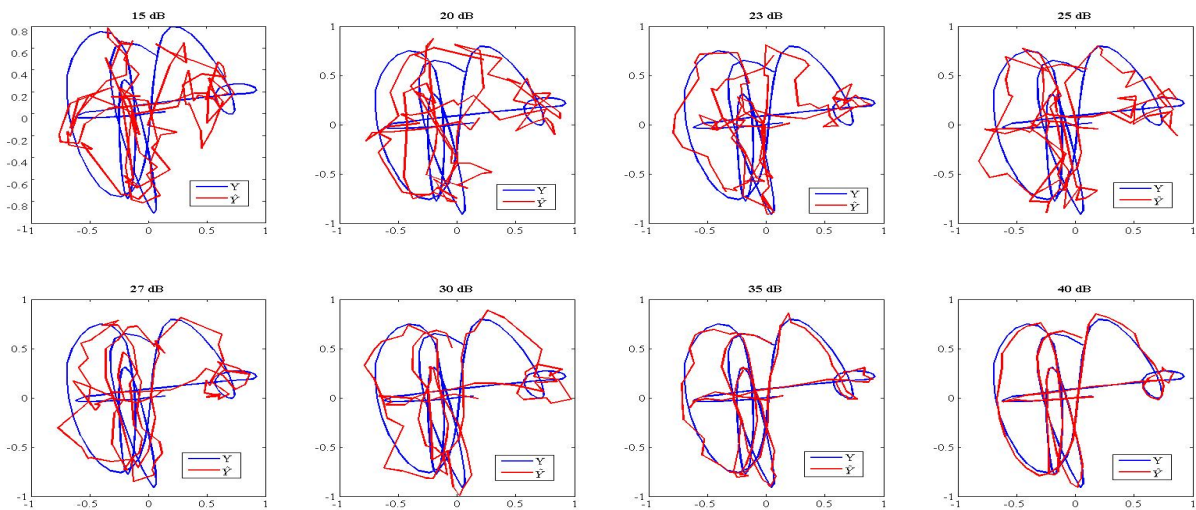


Figure 4.30: Tracking for varying SNR at IBO = -6dB and $N_p = 500$

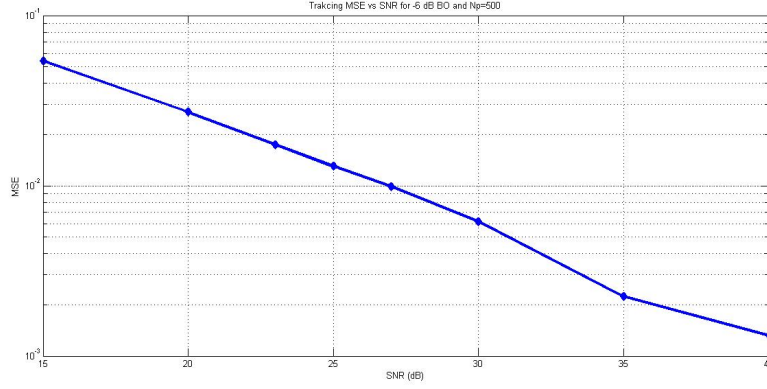


Figure 4.31: MSE vs SNR, IBO= -6 dB and $N_p = 500$

4.5.6 Effect of number of particles

Over the single source nonlinear channel, we saw that the number of particles have to be increased significantly from the linear to nonlinear channel. Now we believe because of the presence of second signal the PAPR increases even further, and therefore the signal is going to be in the saturation zone for longer time, hence making the distribution more bimodal. The bimodality of the distribution creates ambiguity between the choice of correct state and therefore a greater number of particles would lessen the problem to a certain degree.

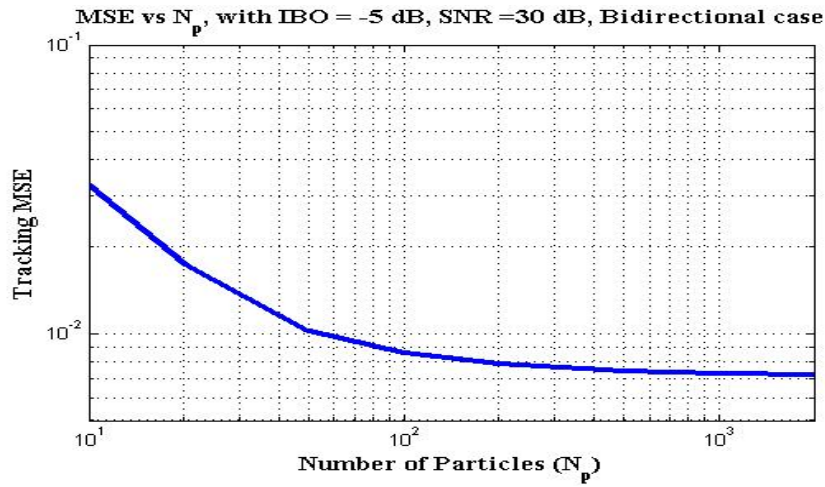


Figure 4.32: MSE vs. N_p for 16-APSK, with SNR= 30 dB, IBO= -5 dB, Bidirectional case

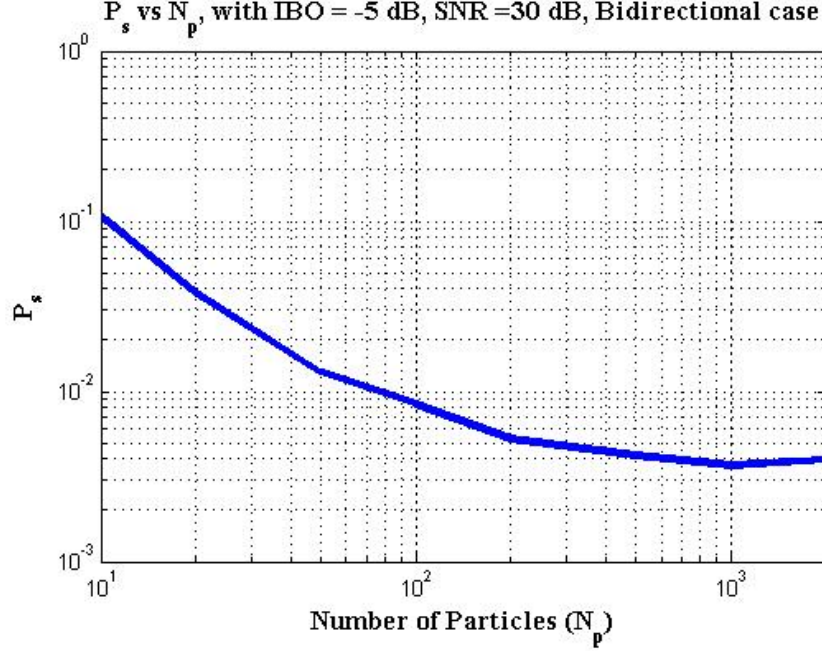
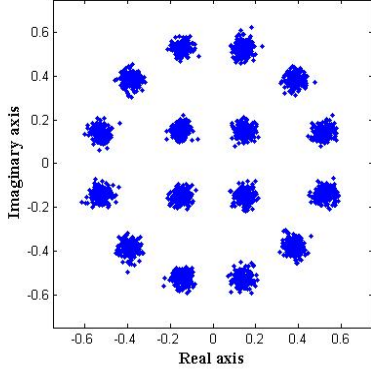


Figure 4.33: P_s vs. N_p for 16-APSK, with SNR = 30 dB, IBO = -5 dB, Bidirectional case

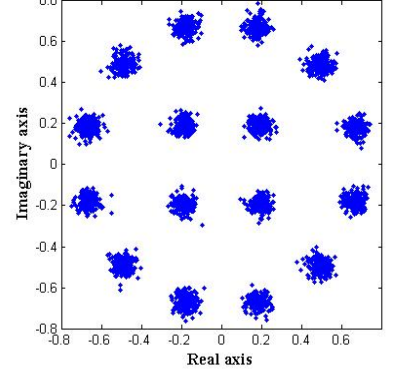
4.5.7 Effect of backoff

As mentioned previously the number of episodes where the signal travels into the saturation zone is going to be higher and therefore we most likely won't be able to operate in low backoffs while maintaining reliable communication. Therefore the effect of lower backoffs might be more severe in this case. With the addition of the second signal the average peak to average power ratio gets even higher than in the single user case and therefore a larger backoff might be required for the dual user to attain a similar performance. We present the MSE vs input backoff curve below in Figure 4.37. The plot confirms our intuition that with larger backoffs there is significant power lost in transmission and the effective SNR at the receiver gets lower. Therefore when we are backed off more we gain linearity but at the expense of losing signal power and reducing the SNR at the receiver. However, when we operate at smaller backoffs we don't lose much power but we introduce the nonlinear distortion that is a consequence of driving the amplifier into saturation.

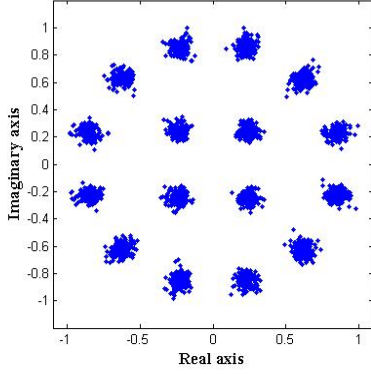
IBO = -12 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -10 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -8 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -7 dB, SNR=35 dB, $N_p=200$, Bidirectional Case

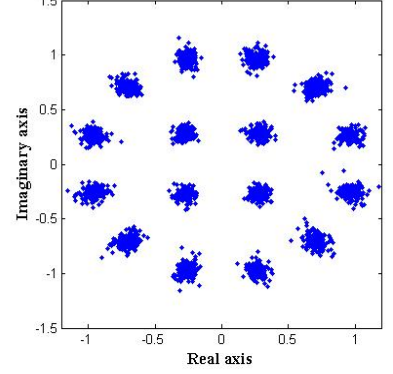
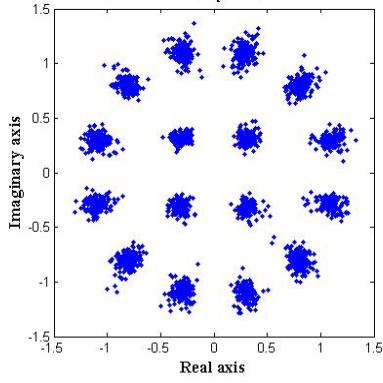
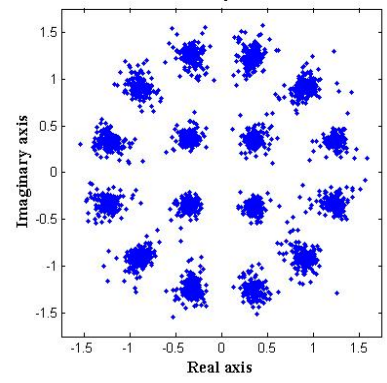


Figure 4.34: Constellation plots for SNR= 35 dB, $N_p = 100$
IBO= -12 dB (top-left), -10 dB (top-right), -8 dB (bottom-left), -7 dB (bottom-right)

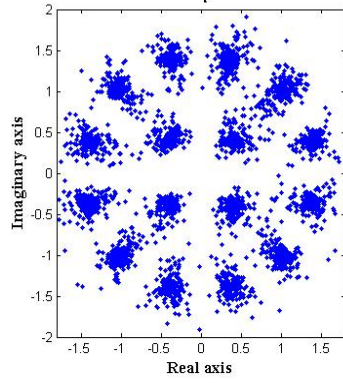
IBO = -6 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -5 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -4 dB, SNR=35 dB, $N_p=200$, Bidirectional Case



IBO = -3 dB, SNR=35 dB, $N_p=200$, Bidirectional Case

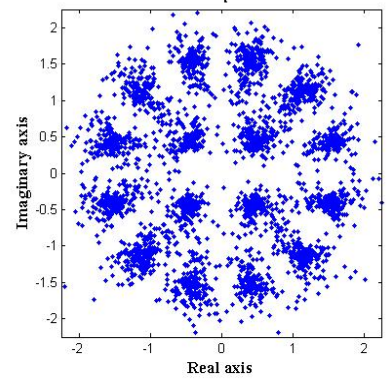


Figure 4.35: Constellation plots for SNR= 35 dB, $N_p = 100$
IBO = -6 dB (top-left), -5 dB (top-right), -4 dB (bottom-left), -3 dB (bottom-right)

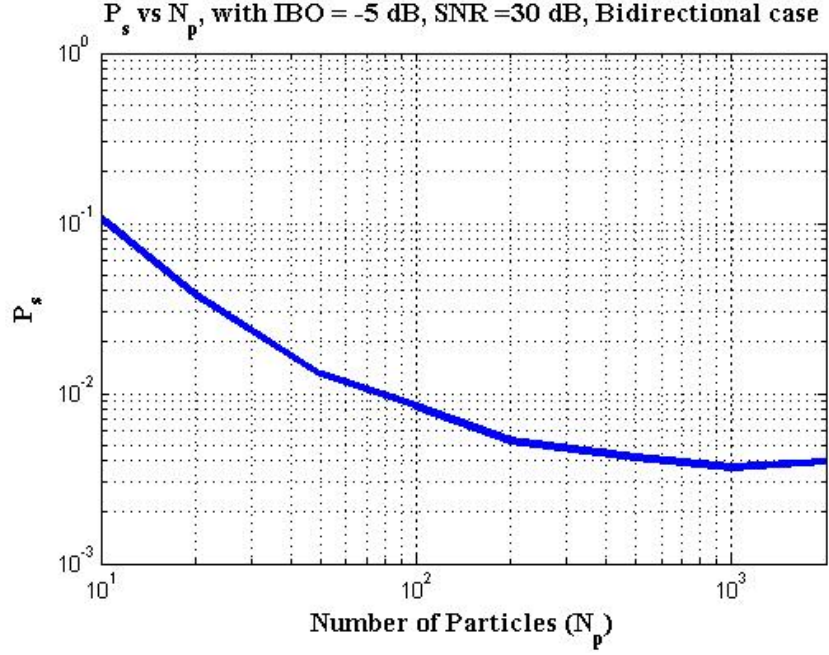


Figure 4.36: P_s vs. IBO with $N_p = 200$ and SNR = 35 dB, Bidirectional case

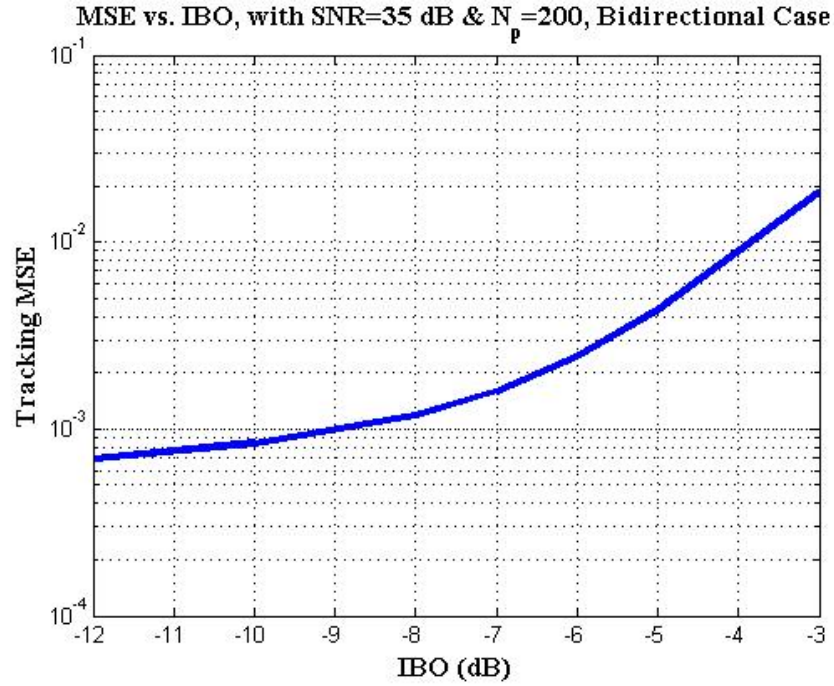


Figure 4.37: MSE vs. IBO with $N_p = 200$ and SNR= 35 dB, Bidirectional case

4.6 Performance Evaluation

We try to recreate the same evaluation criteria that we used for the MAP inverter. We start by looking at the MSE encountered in tracking the waveform. The plot in Figure 4.38 depicts the MSE as a function of SNR for various backoffs. The MSE encounters a floor at higher SNR because of the loss in diversity and therefore we inflate the measurement noise in weight calculation to appease the problem. This is specially helpful when there is only one particle that survives to be propagated along ,*i.e.*, all the other particles in the following time step are just its descendants. Also in the ambiguity zone the state distribution is bimodal, and since our estimate is based on the mean of output of the state model, we are always underestimating when the true state is greater than one or overestimating when it is smaller than one(akin to the concept of lower and higher roots discussed in Chapter 3). This produces some tracking error which is reflected in the plots below.

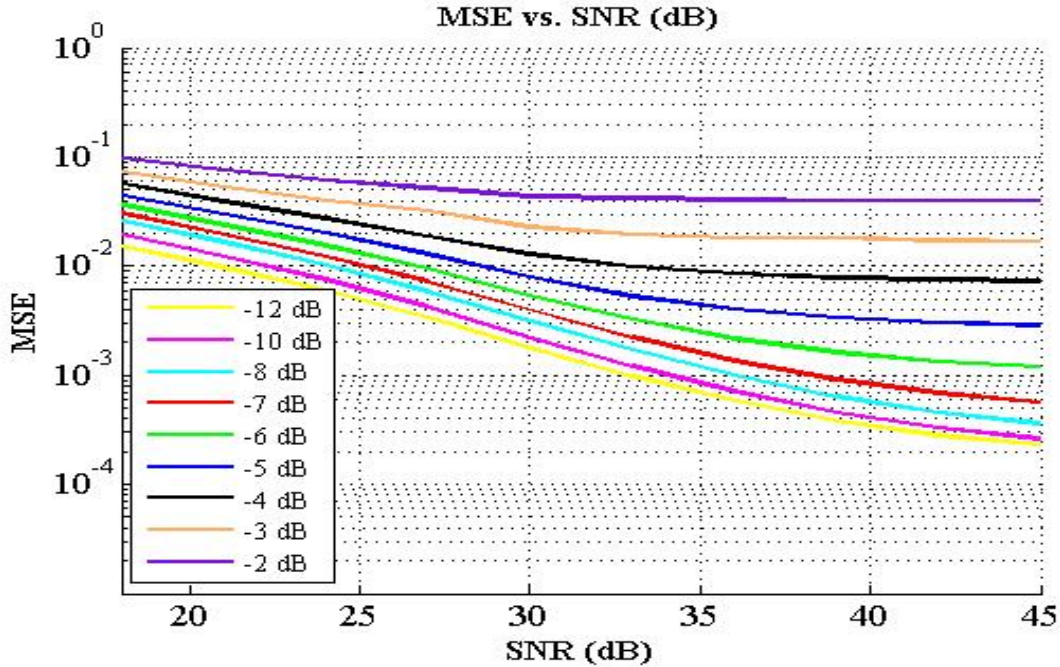


Figure 4.38: Tracking MSE of particle filter

We now look at the final criterion for evaluating the performance of the particle filter, namely the symbol error curves. We evaluated the effects of backoff and SNR in detail previously. We see the manifestation of those results in the Figure 4.39. As the SNR gets higher we have more confidence in our measurements so the error rate is non-increasing. Large backoffs imply operation at a level much smaller than saturation. This means very limited nonlinear distortion results but at the same time we have a much smaller transmission power. By backing off more, we are lowering the effective SNR at the receiver and therefore we see a higher

error rate as exemplified by the -12 dB curve in Figure 4.39. At -7 dB we are able to achieve an error rate of about 10^{-4} and we are still operating in the nonlinear regime of the satellite amplifier. This result concludes our discussion of particle filter and in the next chapter we summarize our results and compare the two algorithms.

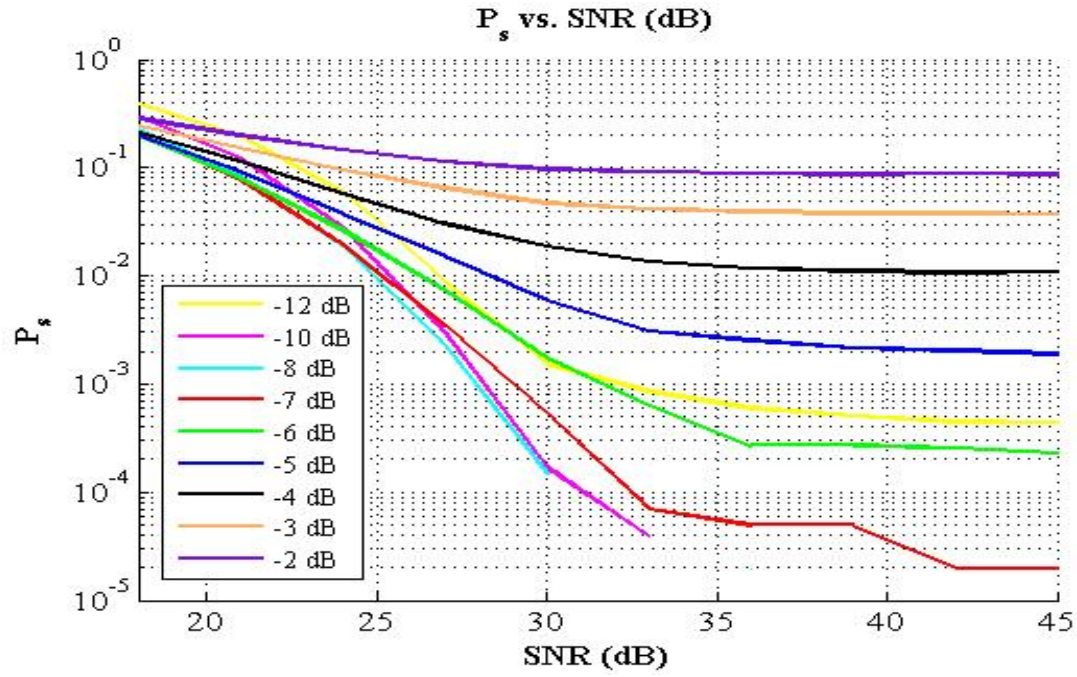


Figure 4.39: P_s vs SNR for particle filter

Chapter 5

Summary and Future Work

5.1 Summary of Results

In this research, we focus on receiver-based techniques for combatting nonlinearity in satellite relay channels. Our particular interest is in bidirectional (same-frequency) communication, where we need to combat self-interference using known side-information. This is most easily done with an echo cancellation approach, but this relies on the channel model being approximately linear. So, we present the two approaches and their performance for inverting the nonlinearity introduced by nonlinear satellite channels. In the case of bidirectional communication, this is followed by echo cancellation.

For this work, we focused on the Saleh model which characterizes a noninvertible nonlinearity. The noninvertibility of the Saleh model stems from the fact that there is two-to-one mapping and therefore there is ambiguity in choosing the correct input for an observed output. One traditional approach is to ‘backoff’ the amplifier into a more linear regime of operation, but this has performance impacts (loss of output power and amplifier efficiency).

We attempt to create an overall artificial linear channel by countering the nonlinearity. We devised two stochastic algorithms that attempt to invert the Saleh nonlinear channel. These algorithms focus on correct estimation of the waveforms or locus generated at the input of the nonlinear relay. The first algorithm is a MAP-based decision method. The mathematical inversion of the Saleh model yields two possible roots that correspond to the input for each received output sample. And the sequence of roots that maximized the probability of received sequence or a-posteriori probability was chosen. We saw that the algorithm estimated the input locus with small error at high SNR. The error increased, of course, as we decreased the received SNR. The results are discussed in detail in chapter 3.

This low-SNR performance degradation motivated us to look for another algorithm that is more robust to additive noise. Our next approach was the particle filter which is a Bayesian estimator of the input waveform to the HPA based on the noisy measurements available at the receiver. The discussion and results for particle filter are documented in chapter 4.

Based on analysis performed in Chapters 3 and 4, we now present a brief comparison between the two algorithms in terms of performance. For the MAP algorithm we choose a parsing length of 8 and for the particle filter we use 200 particles. Moreover, we fix the data set that would be processed by both the algorithms. We start by first comparing the tracking MSE of each algorithm. In the case of each algorithm we calculate the MSE in waveform tracking as a function of SNR and input backoff, and the results are displayed in Figure 5.1.

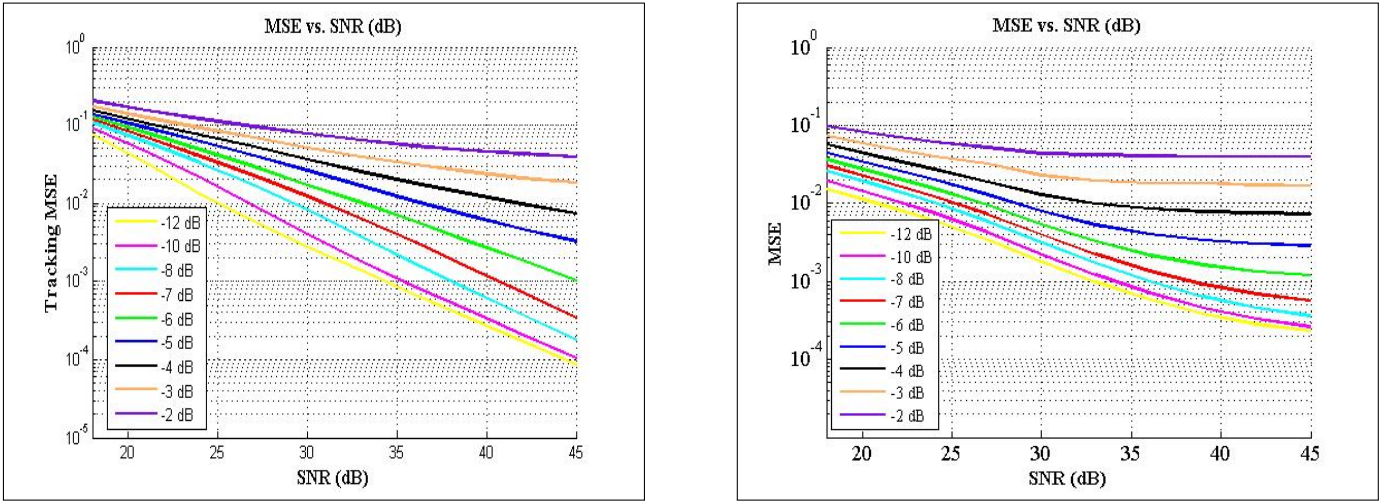


Figure 5.1: Comparison of tracking MSE of MAP inverter (left) and particle filter (right)

From Figure 5.1 we notice that at the low end of SNRs (say < 25 dB), the particle filter has lower MSE and does better for almost all combinations of SNR and backoff. However, as we increase SNR we see that the MAP inverter catches up and the MSEs in each case are very similar. In fact, towards the higher end of the SNR range presented in the figure, the particle filter experiences a floor on MSE which is a consequence of the fact that we are using a fixed particle population size, namely 200. If we use more particles we expect to lower this floor. On the other hand, the MAP detector does not encounter a floor; infact the MSE seems to decrease with the same slope over the range of SNRs presented in the figure.

Next we compare the symbol error rates of the two algorithms. Again the plots are reproduced in Figure 5.2 from their respective chapters. The nonlinear distortion at backoffs greater than 4 dB introduces severe distortion which is reflected in the SER curves, which makes these IBO's undesirable operating points. The

two algorithms attain a P_s of 10^{-3} around an SNR of 30 dB for various backoffs *e.g.*, -6,-7, and -8. It seems that the particle filter achieves a SER of 10^{-3} at a slightly lower SNR, but it encounters a performance floor soon after, due to population size of particles. For this particular simulation the MAP inverter achieved an SER of 0 for a run of 10^5 symbols for backoffs of -7,-8, and -10 dB for SNR's beyond 39, 33, and 36 dB respectively.

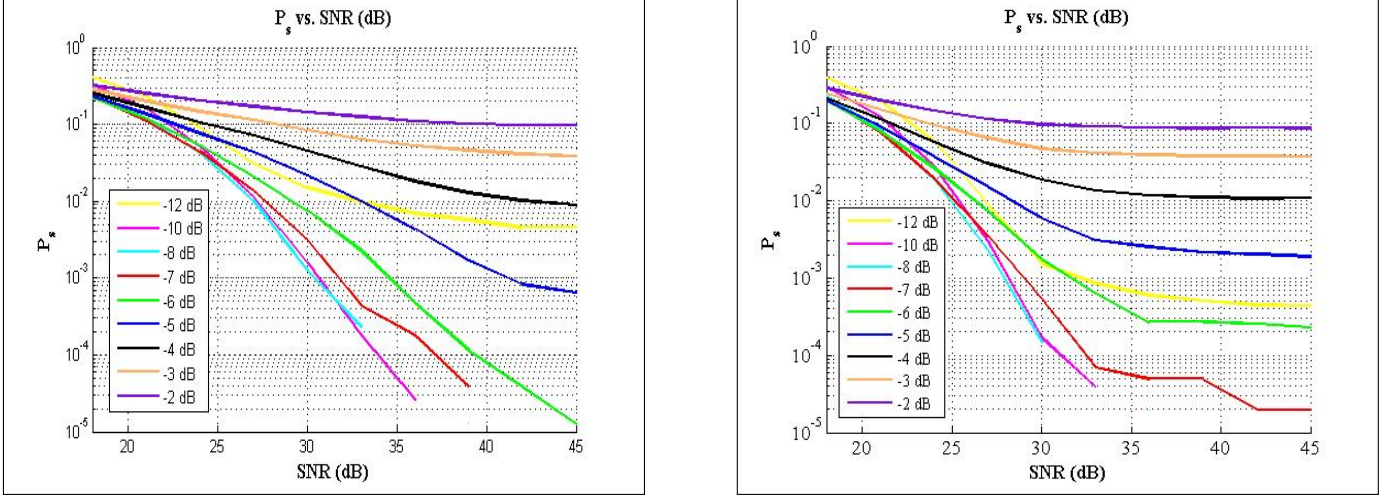


Figure 5.2: Comparison of SER plots for MAP inverter(left) and particle filter(right)

We next discuss the computational times associated with each algorithm. As it stands, the particle filter operates continually, rather than over ambiguous spans in the data. This is an inherent complexity penalty on the particle filter. On the other hand, this complexity is invariant to IBO, as long as particle population size is fixed, whereas the MAP detector's complexity depends strongly on IBO.¹

On the other hand, if we fix the parsing length deployed in the MAP inverter, we get different run times based on the amount of time the signal spends in the ambiguity zone. For larger backoffs there is little ambiguity and therefore the MAP inverter is not invoked and the computational time is smaller. Whereas in the case of smaller backoffs the time spent in saturation is considerably higher and the MAP inverter would be a lot more occupied trying to estimate the ambiguous sequences. So the time complexity experienced by the MAP inverter is a function of backoff and to some extent when the noise power is significant enough to push the signal into the Run time on a single PC for 100000 symbols (800000 samples) is shown in Figure 5.3.

¹This is not to be confused with the previous discussion where it was suggested that increasing the population of particles can appease the floors in performance. Here we are comparing complexity with the number of particles kept fixed.

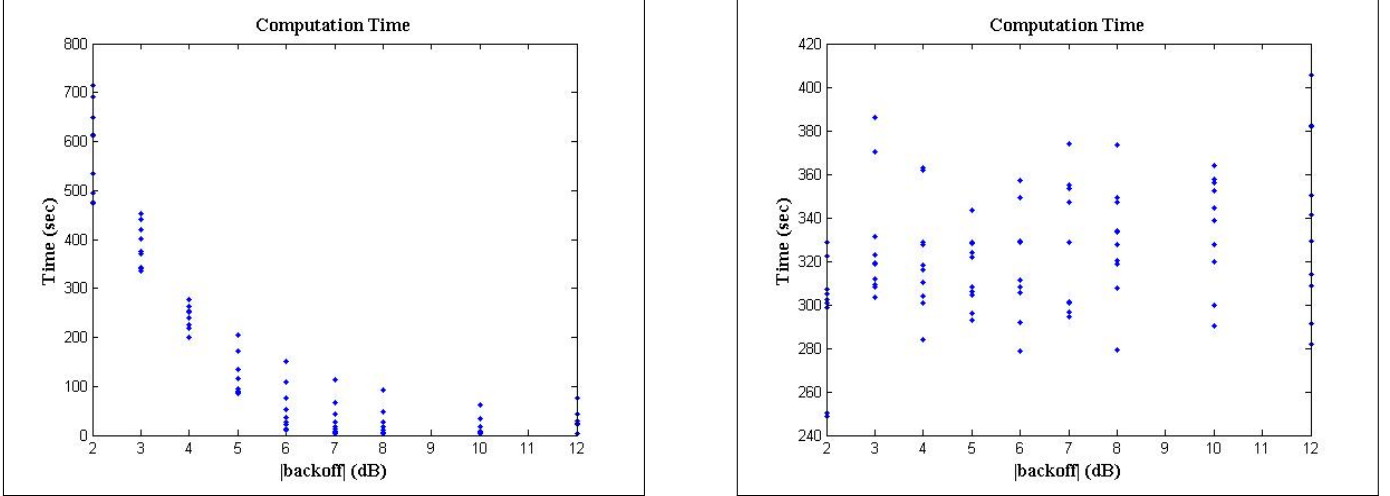


Figure 5.3: Computational time comparison of MAP inverter(left) and particle filter(right)

5.2 Future work

5.2.1 MAP detector

Two things seem worth study. First, further noise reduction prior to root-finding could be helpful. We tried simple linear filtering to pass the signal band while rejecting out-of-band noise, but this did not prove fruitful. It seems that the intermodulation signal components rejected by the linear filter are fairly important at waveform inversion of the nonlinearity. We have experimented some with wavelet de-noising methods that show some promise. However, in the end we can probably only improve the effective SNR seen by the root-finder by a few dB.

A bigger issue is computational. We are already forced to parse long ambiguous spans into shorter segments of say length $N = 8$ for MAP processing. This still requires evaluation of 256 candidate root sequences, which helps on computation load. However, the tail estimates from one string become the anchor points for the next segment, and these are occasionally wrong. If the entire ambiguous span can be processed, experimental evidence is that much better MSE can be achieved. So, signal processing schemes that find good approximations to the MAP sequence, if not perhaps the best, are needed for longer spans.

5.2.2 Particle filtering

We incorporated the AM-PM modulation into the MAP inverter, however, the particle filter based inverter in its current state just focuses on countering the AM-AM distortion. We plan on incorporating the AM-PM inverter soon in the particle filter based inversion algorithm and repeat the simulations.

Also, smoothing, in conjunction with particle-based methods, rather than filtering, should provide smaller MSE by allowing the estimate of the waveform to emerge after a short delay, [7]. This is analogous to the improved performance of Kalman smoothers over Kalman filters. The complexity increase seems quite reasonable, and it could be that smoothing allows a reduction in the required number of particles.

Regarding computation, particle filtering is unfortunately not amenable to parallel processing, in for example FPGA platforms. Particle propagation and likelihood calculations can be done in parallel, but resampling is not parallelizable. Further investigation of this aspect is probably necessary for implementation in high-speed satellite communication applications.

Appendix A

Algorithm For Reduction to Strictly-Proper From

Consider a transfer function of the form

$$H(z) = \frac{B(z)}{A(z)}.$$

In the special case that $H(z)$ is either a Butterworth or a Chebyshev filter there are zeros at $\omega = \pi$, or $z=-1$. The numerator, $B(z)$ can be expressed as following

$$H(z) = k_{H(z)} \frac{(z+1)^N}{A(z)} \tag{A.1}$$

Another piece of information that is used in the algorithm is making the passband gain of the strictly-proper transfer function equal to that of the original transfer function. At $\omega = 0$ we have $z = 1$ and thus from the numerator in (A.1), $(z+1)^N$ becomes 2^N . If m zeros are removed from $B(z)$, the new polynomial \tilde{B} has to be scaled by 2^m to obtain the same bandpass gain. Since we are only using the trivial case of removing one zero, the new passband gain becomes 2^{N-1} and therefore we just scale by 2 in (A.2) in the Algorithm below.

We now present an algorithm to obtain a strictly proper transfer function with the same passband gain.

Algorithm 3 Algorithm for finding strictly-proper transfer function from Chebyshev/Butterworth filter

- Find the degree of the Numerator $B(z)$. Call it N .
- Create a polynomial $\tilde{B}(z)$ of degree $(N - 1)$ in z as follows

$$\tilde{B}(z) = (z + 1)^{N-1}$$

- Find $k_{H(z)}$, the gain of $H(z)$.
- Calculate the gain of strictly-proper function

$$k_p = 2k_{H(z)} \tag{A.2}$$

- Calculate the new numerator $B_p(z)$ that produces a strictly proper transfer function

$$B_p(z) = 2\tilde{B}(z)$$

- Create proper transfer function $H_p(z)$ as follows

$$H_p(z) = \frac{B_p(z)}{A(z)}$$

Appendix B

Obtaining Observer-Canonical Form

The most common canonical forms; the controller canonical form (CCF) and the observer canonical form (OCF) can be obtained using the methods described in [11]. For this research we used *Matlab*'s built-in command '*tf2ss*' which accepts the numerator and the denominator of a transfer function and generates the corresponding state model in the form of the matrices A, B, C and D .

$[A, B, C, D] = \text{tf2ss}(B(z), A(z))$

By default the model is in CCF, but for reasons mentioned in section 4.2 we want to convert this into OCF. To attain the corresponding OCF we need to change A, B, C , and D such that transition matrix in OCF is the transpose of the transition matrix in CCF, and the input and output matrices in OCF are obtained via transposing the output and input matrices in CCF respectively. The feed-forward term D stays the same for both canonical forms.

$$A_o = A_c^T$$

$$B_o = C_c^T$$

$$C_o = C_c^T$$

$$D_o = D_c$$

Here the subscripts **c** and **o** correspond to the canonical and observer forms respectively, and the superscript **T** corresponds to the transpose. As both the CCF and OCF are state models that emerge from strictly-proper transfer functions we have the feed-forward term $D_o = D_c = 0$.

Bibliography

- [1] Optimizing satellite communications using doubletalk[®] Carrier-in-Carrier[®] & cdm-625 advanced satellite modem, Comtech EF Data, 2010.
- [2] J.B. Anderson. *Digital transmission engineering*. IEEE Series on Digital & Mobile Communication. Wiley, 2006.
- [3] J. S. Bay. *Fundamentals of linear state space systems*. McGraw-Hill College, 1999.
- [4] S. Benedetto and E. Biglieri. *Principles of digital transmission, with wireless applications*. Information Technology Series. Springer US, 1999.
- [5] J. V Candy. *Bayesian signal processing, classical, modern and particle filtering*. John Wiley/IEEE Press, 2009.
- [6] C.T. Chen. *Linear system theory and design*. Oxford University Press, 1999.
- [7] T.C. Clapp and S.J. Godsill. Fixed-lag smoothing using sequential importance sampling. In *Bayesian Statistics 6: Proceedings of the sixth Valencia international meeting*, volume 6, pages 743–752. Oxford University Press, 1999.
- [8] G.E. Corazza. *Digital satellite communications*. Information Technology: Transmission, Processing and Storage. Springer, 2007.
- [9] A. Doucet. *Sequential Monte Carlo methods*. Wiley Online Library, 2001.
- [10] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.
- [11] G.F. Franklin, J.D. Powell, and A. Emami-Naeini. *Feedback control of dynamic systems*. Pearson Education, 2011.

- [12] M. Gomes, V. Silva, F. Cercas, and M. Tomlinson. Low back-off 16-apsk transmission using magnitude modulation and symbol quantization. In *Satellite and Space Communications, 2008. IWSSC 2008. IEEE International Workshop on*, pages 229–233. IEEE, 2008.
- [13] F. Gustafsson. Particle filter theory and practice with positioning applications. *Aerospace and Electronic Systems Magazine, IEEE*, 25(7):53–82, July 2010.
- [14] S. Haykin. *Adaptive filter theory*. Prentice Hall, Upper Saddle River, N.J, 2002.
- [15] A.A. Kumar. *Control systems*. PHI Learning, 2007.
- [16] M. Mallick, V. Krishnamurthy, and B. Vo. *Integrated tracking, classification, and sensor management: theory and applications*. John Wiley & Sons, 2013.
- [17] S. Maskell and N. Gordon. A tutorial on particle filters for on-line nonlinear/non-Gaussian Bayesian tracking. In *Target Tracking: Algorithms and applications (Ref. No. 2001/174), IEE*, volume Workshop, pages 2/1–2/15 vol.2, Oct 2001.
- [18] J. Peng, C. Xu, S.G. Wilson, and T. Xie. Two-way (same-frequency) relaying: Information rates and DVB-S2 performance. 2013.
- [19] H.V. Poor. *An Introduction to Signal Detection and Estimation*. A Dowden & Culver book. Springer, 1994.
- [20] B. Ristic. *Particle filters for random set models*. Springer, 2013.
- [21] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.
- [22] A.A.M. Saleh. Frequency-independent and frequency-dependent nonlinear models of twt amplifiers. *Communications, IEEE Transactions on*, 29(11):1715–1720, November 1981.
- [23] M. Salehi and J.G. Proakis. *Digital communications*. McGraw–Hill, New York, 2007.
- [24] S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.
- [25] A. Sayed. *Adaptive filters*. Wiley-Interscience IEEE Press, Hoboken, N.J, 2008.
- [26] M. Schetzen. *The Volterra and Wiener theories of nonlinear systems*. Krieger Pub., 2006.
- [27] R. Srinivasan. *Importance sampling: Applications in communications and detection*. Springer, 2002.
- [28] S.G. Wilson. *Digital modulation and coding*. Prentice Hall, 1996.

- [29] C. Xu and S.G. Wilson. Minimizing transponder cost for two-way data exchange in satellite networking. In *Communications (ICC), 2012 IEEE International Conference on*, pages 3250–3254, June 2012.