Decision–Feedback Equalisation with Fixed–Lag Smoothing in Nonlinear Channels

by

David Wayne Bartel

Bachelor of Applied Science (Exploration Geophysics),
Curtin University of Technology, 1982
Graduate Diploma of Applicable Mathematics,
Murdoch University, 1988
Master of Mathematical Sciences (Signal and Information Processing),
University of Adelaide, 1999

Thesis submitted for the degree of

Doctor of Philosophy

in

Electrical and Electronic Engineering
University of Adelaide

2006

Chapter 4

Results

4.1 Introduction

In chapter 2 we introduced the FLSDFE algorithm of Perreau *et al.* [52] and illustrated how the algorithm could be extended for use on channels whose baseband envelope input–output relationship took the form of a finite Volterra series. Chapter 3 subsequently extended the static analysis of chapter 2 to cover the dynamics of error propagation, using state space models. In the current chapter, we continue the analysis of the FLSDFE algorithm, using the single–distinct–errors state space model of Choy and Beaulieu [24], as introduced in section 3.4.6. A 'resonance' phenomenon was observed for various simulated communication channels, using the Choy and Beaulieu model. This phenomenon appears not to have been discussed in the previous literature on DFE error propagation.

Section 4.2 first discusses a methodology for the physics–based simulation of an underwater acoustic communication channel, comparing transmission loss calculations done with a rigorous wave–theoretic approach against a geometrical–acoustics (ray–tracing) approach. It is shown that, at the frequencies typically used in underwater acoustic communications, a ray method is a satisfactory and convenient approximate method for the physical part of the overall communications system simulation. Using the ray approach, physics–based simulations of the FLSDFE algorithm are conducted for several underwater acoustic channel models, and a kind of resonance phenomenon is observed with the Choy and Beaulieu single–distinct–errors model. This phenomenon is subsequently investigated further, in section 4.3, using simplified channel models. Four channel models are used, both linear and nonlinear, minimum–phase and nonminimum–phase, to quantify the effects of these aspects on the FLSDFE algorithm's performance.

As discussed in chapter 2 and illustrated in figure 2.3, the FLSDFE algorithm does not feed back smoothed estimates $\hat{x}_{t-n|t}$ at nonzero smoothing lags n. Filtered outputs

$\hat{x}_{t|t}$, however, are fed back. Moreover, the smoothed output $\hat{x}_{t-n|t}$ is functionally dependent upon these filtered outputs $\hat{x}_{t|t}$, as demonstrated via example in (2.26) and (2.53). To understand the complicated stochastic dynamics of the FLSDFE output process $\{\hat{x}_{t-n|t}\}$, we therefore first need to understand the dynamics of the filtered process $\{\hat{x}_{t|t}\}$, whose behaviour is complex enough.

4.2 Underwater acoustic channel models

To investigate the potential benefits of the FLSDFE algorithm in practical channels, we constructed simple approximations of underwater acoustic communication channels. There is a well–developed literature on ocean–acoustic modelling, and we draw in particular on the theory of propagation of sound within a homogeneous fluid layer bounded above by a pressure release surface and below by a higher velocity, homogeneous solid half–space [15, 32].

4.2.1 Computing pressure fields rigorously

The model HANKEL was written by the author to rigorously compute the complex acoustic pressure field in a homogeneous fluid layer bounded above by a pressure release surface and below by a higher velocity, homogeneous solid half–space. Appendix J discusses the HANKEL model in more detail. Figure 4.1 gives a plot of the transmission loss from a monochromatic point source of frequency 30 Hz. The computed result reproduces that of Schmidt and Jensen [56]. This problem consists of a shallow ocean, of depth 100 m, with uniform speed of sound 1500 $\mathrm{m\,s^{-1}}$ and density 1000 $\mathrm{kg\,m^{-3}}$. The bottom is a solid halfspace, with density 2000 $\mathrm{kg\,m^{-3}}$, compressional speed 1800 $\mathrm{m\,s^{-1}}$, shear speed 600 $\mathrm{m\,s^{-1}}$, compressional attenuation 0.1 dB/λ and shear attenuation 0.2 dB/λ. At a depth of 95 m is a monopole point source emitting a tone of frequency 30 Hz. The problem of Schmidt and Jensen is that of computing the transmission loss at the ocean floor, out to 5 km.

The blue curve gives the loss as computed from classical ray tracing, which is an approximate method based on geometrical acoustics. The red curve was produced from an exact form of the solution for the complex pressure field (see appendix J). This solution expresses the field as an infinite series of oscillatory integrals. HANKEL estimates the field by numerical quadrature over a finite set of the most significant integrals.

NOTE:
This figure is included on page 103
of the print copy of the thesis held in
the University of Adelaide Library.

Figure 4.1. Transmission loss curves at 30 Hz. The HANKEL result is in excellent agreement with Schmidt and Jensen's figure 5 [56].

Although the ray method is orders of magnitude faster than HANKEL, it is clearly inaccurate for this problem. This is due to the fact that the ray method is exact only in the asymptotic limit of infinitely high source frequency, but the frequency in figure 4.1 is very low, 30 Hz. At practical underwater communication frequencies we show that ray theory is adequate [30, 32, 36].

Figures 4.2 and 4.3 extend the benchmark problem of Schmidt and Jensen to the whole water column. We clearly see that the simple ray solution does not match the exact solution with sufficient accuracy in this case. HANKEL was about 10,000 times slower than ray tracing, taking about one week to compute the field out to 3 km. The rigorous solution displays fine ripples that are an interference pattern arising from the superposition of the single waterborne mode and an interface wave known as a *Scholte* wave [56]. Many other models do not take such interface waves into account, as these waves generally do not contribute significantly to the acoustic pressure field.

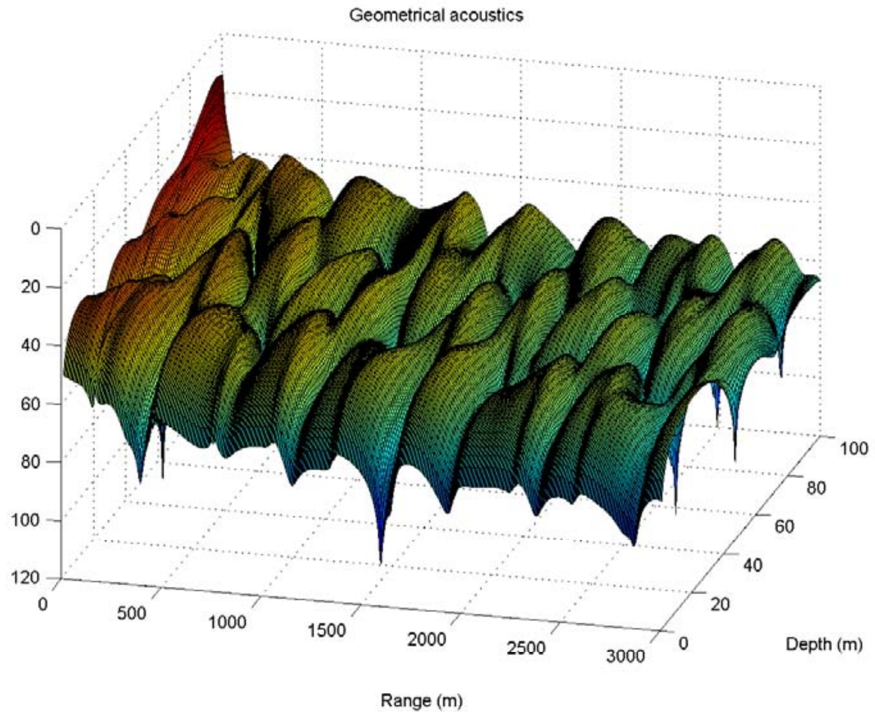

Figure 4.2. Transmission loss surface at 30 Hz computed by simple ray tracing.

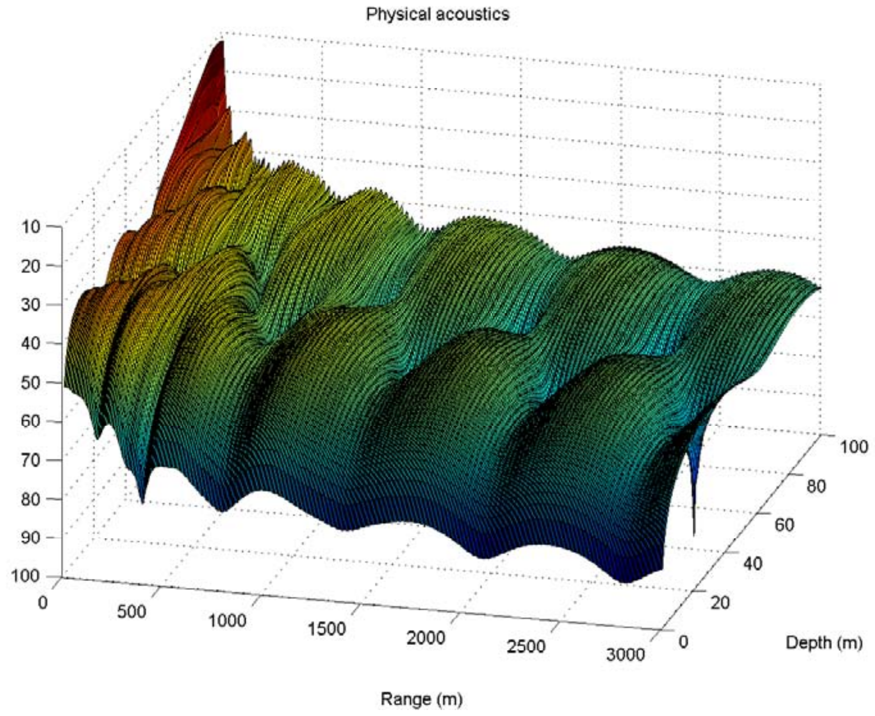

Figure 4.3. Transmission loss surface at 30 Hz computed by wavenumber integration.

4.2.2 Computing pressure fields with rays

Although HANKEL produces exact pressure fields, it is not suitable for general use within a simulation of underwater acoustic communication. At higher frequencies, the underlying integrals rapidly become ever more highly oscillatory, and it becomes necessary to compute ever more of these integrals to obtain a pressure value of sufficient accuracy. Since HANKEL takes no shortcuts beyond those of numerical quadrature, the run times become very long for many problems of interest. This is true of other modelling approaches, as well, though to a lesser extent. Hence we turn to a much faster, simpler and more versatile technique, that of classical ray tracing. This is amply discussed in the ocean–acoustics literature, and will not be described here, other than to note that it derives from an asymptotic approximation of the integrals that are treated exactly by HANKEL [15, 30, 32, 36].

Practical underwater acoustic communication systems typically have frequencies of several kHz or more [31]. To show the relative accuracy of ray tracing at higher frequencies, we computed the transmission losses from a monochromatic point source of frequency 1 kHz for two different ocean–acoustic environments. Figures 4.4 and 4.5 show transmission losses for two bottom types. In each figure, we compare transmission loss calculations obtained from HANKEL with those produced by classical ray tracing, and with two state–of–the–art models, ORCA, a *normal modes* model [63], and RAMS, a *parabolic equation* model [25].

Together, these four models cover the gamut of common approaches to solving the Helmholtz equation [32]. (The Helmholtz equation is a linear second–order partial differential equation that plays a central rôle in computational ocean acoustics, akin to the rôle that the Schrödinger equation plays in quantum chemistry.) By using separate computer codes, based on widely differing mathematical approaches, a consensus of solutions can be taken to mean that all solutions are close to the true solution. We use this philosophy to provide a benchmark to the accuracy of the method of classical ray tracing, which is of interest in practical simulations of underwater acoustic communication systems, due to the versatility and speed of ray tracing.

Table 4.1 gives the ocean–acoustic material properties of eight canonical seabed types, as suggested by Jensen and Kuperman [35]. Jensen and Kuperman considered this set of seabed types as being representative of the spectrum of types likely to be encountered in the world's oceans. The data are arranged in order of increasing density ρ_b

of the bottom, in kg m^{-3}. The columns c_{2p} and c_{2s} are the speed of sound of P (compressional) and S (shear) waves, respectively, in m s^{-1}. Decibel attenuation rates α_{2p} and α_{2s} are in the units dB/λ_{2p} and dB/λ_{2s}, respectively, where λ_{2p} and λ_{2s} are the wavelengths of P and S waves, respectively:

$$\lambda_{2p} = \frac{\omega}{c_{2p}}, \quad \text{and} \tag{4.1}$$

$$\lambda_{2s} = \frac{\omega}{c_{2s}}, \tag{4.2}$$

where $\omega = 2\pi f$, with f being the source frequency, in Hertz (Hz).

Table 4.1. Jensen and Kuperman's representative spectrum of seabed types [35]. Shown highlighted in grey are the two seabed types used in the FLSDFE simulations.

NOTE:
This table is included on page 106
of the print copy of the thesis held in
the University of Adelaide Library.

A shallow uniform ocean of depth 50 m, density $\rho_1 = 1000$ kg m^{-3} and speed of sound $c_1 = 1500$ m s^{-1} was used with each of the bottom types highlighted in the table above. A point monochromatic source of frequency 1 kHz was placed at depth 25 m below the surface of the sea. The transmission loss from this source was computed at depth 5 m, out to a range of 500 m, and the results are presented as figures 4.4 and 4.5 below. Note that the HANKEL data were computed at a coarse range interval, owing to the much slower run times of this model.

As the figures show, there is excellent overall agreement of each model with the other models for each of the environments. This shows that classical ray tracing is adequate for the given problem set. By extension, ray tracing should be adequate for frequencies above 1 kHz, as the ray method is asymptotically exact in the limit of infinite frequency [32]. Classical ray tracing is by far the quickest of the four methods for synthesizing the complex acoustic pressure field at a receiver.

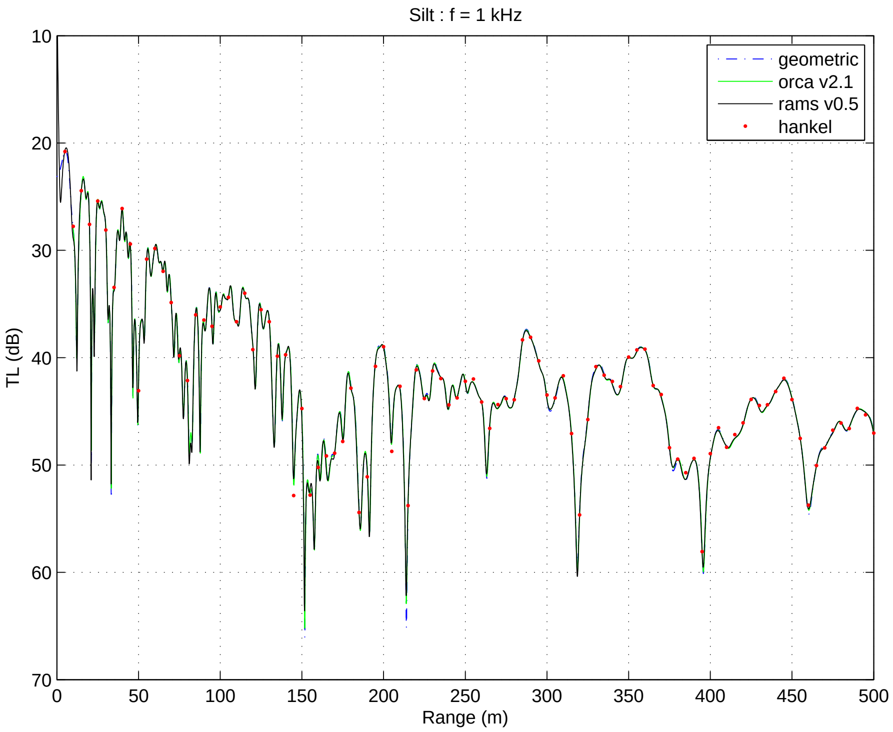

Figure 4.4. Transmission loss curves for Silt bottom.

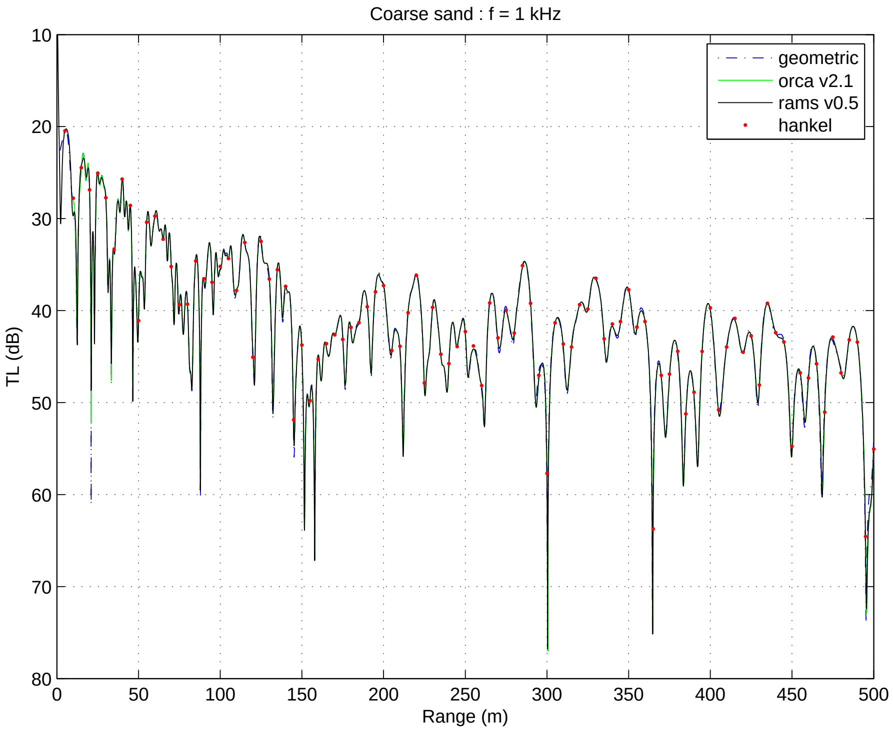

Figure 4.5. Transmission loss curves for Coarse Sand bottom.

4.2.3 Ray–based channel models

In this section, we focus on the ray solutions at a range of 500 m for the silt and coarse sand bottom types. We model the channels as having a linear finite–impulse response (FIR). The linearity is appropriate given that the acoustic wave equation is linear at typical underwater communication sound pressure levels [15, 30–32, 36, 56, 63]. Moreover, a finite–impulse response is what is observed in real underwater acoustic communication channels [32, §7.6.3].

The real part of the impulse response in each case was quantized to the nearest 0.001 second, producing the simplified channel models in figures 4.6 and 4.7. The period between samples was chosen as the reciprocal of the source frequency, 1 kHz. The impulse response data is normalized so that the sum of the squares of the response values is unity. Note that a bulk delay has been removed, corresponding to the time of first arrival of the earliest ray to reach the receiver. Positive and negative signs account for an even and odd number of surface reflections, respectively, with each surface reflection being equivalent to multiplication of the complex pressure by -1 (the surface is a pressure–release interface). The relative magnitudes of the impulse response data arise from the magnitude response of the bottom reflection coefficient, and the number of bottom interactions encountered by each ray between source and receiver.

The data in figures 4.6 and 4.7 thus provide an approximate normalized FIR representation of the ocean–acoustic channel's impulse response. It is these abstracted channel models that will be used in the subsequent FLSDFE simulations. With BPSK signalling, the quantities $h(k)$ in figures 4.6 and 4.7 are the (sparse) kernel coefficients $h_1(k_1), k_1 \in \{0, \ldots, N\}$, of the linear part of the SISO channel model (2.50). The symbol lag (k) in the figures is interpreted as discrete time k_1 in the channel model.

From the normalized FIR models in 4.6 and 4.7, we can formulate corresponding z transform models for the channels [43]. Figures 4.8 and 4.9 show the locations of the poles and zeros for each channel, as computed with MATLAB$^{\text{TM}}$. The poles are all at the origin, and the number next to the cross indicates their multiplicity. Zeros are indicated by open circles, and the dashed curve is the unit circle.

We note immediately that neither channel is strictly minimum phase, since each has at least one zero outside the unit circle. The coarse sand model, however, seems to be worse, in the sense that there are more zeros outside the unit circle, and many of the

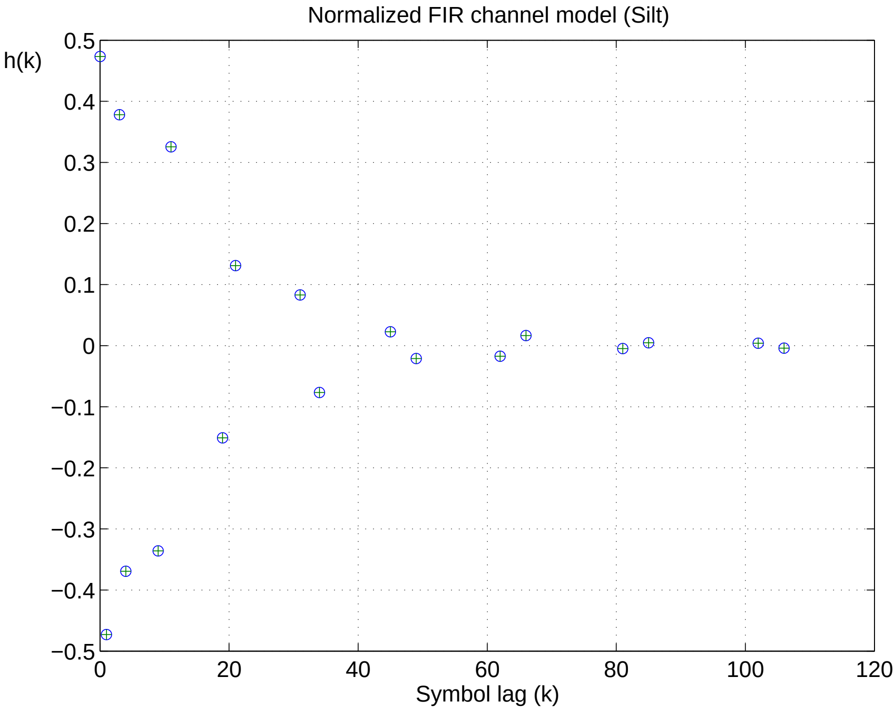

Figure 4.6. Channel model for Silt bottom.

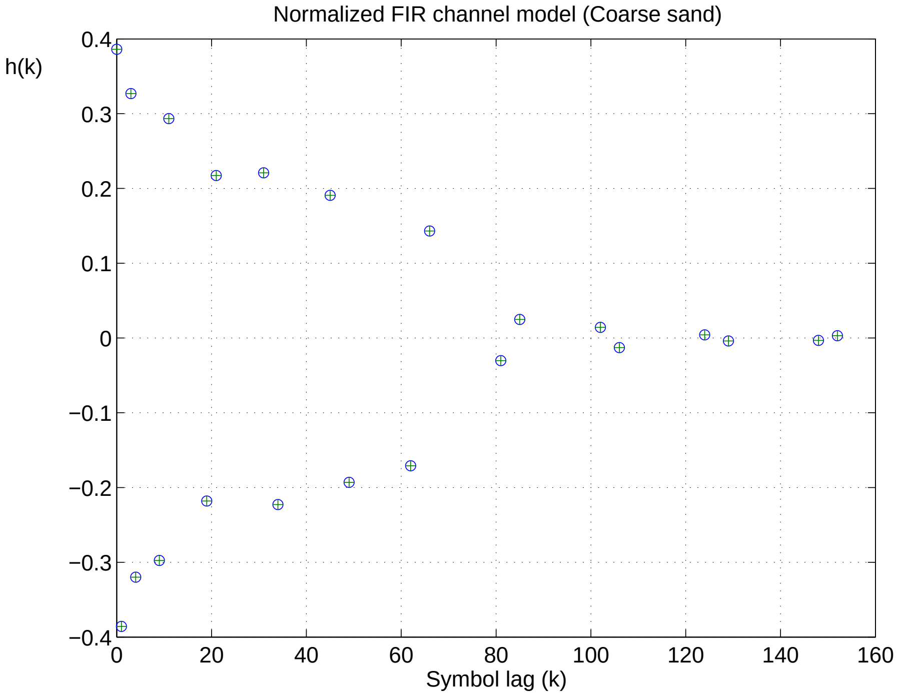

Figure 4.7. Channel model for Coarse Sand bottom.

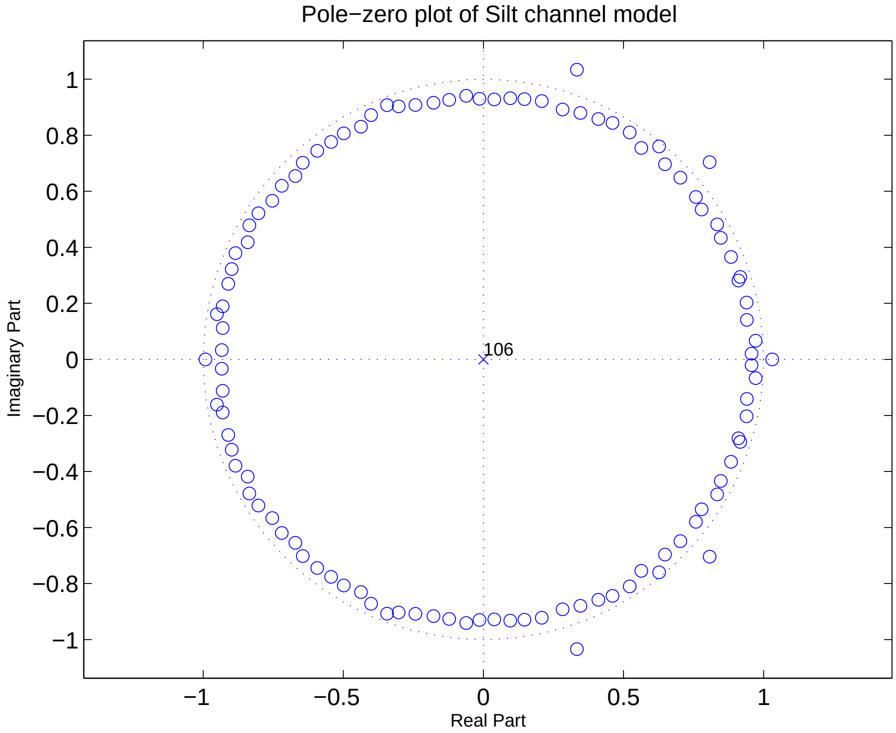

Figure 4.8. Pole–zero plot for silt bottom.

zeros are closer to the unit circle. We may expect worse equalisation performance for the coarse sand bottom model than for the silt bottom model.

Based upon the normalized FIR channel models in figures 4.6 and 4.7, BPSK transmission was simulated and BER versus input SNR curves for the FLSDFE algorithm were constructed. Figures 4.10 and 4.11 provide the empirical BER curves for the silt and coarse sand bottom models, respectively. At each SNR value the FLSDFE algorithm was run at zero, minimum and maximum smoothing lags. For each combination of SNR and lag, the same number of bits were transmitted.

The BER curves show that in each case smoothing improves performance, even though the channels were not strictly of minimum phase. The greatest BER reduction came about from smoothing at maximum lag.

To investigate the performance of the FLSDFE algorithm further, use was made of the single–distinct–errors state space model of Choy and Beaulieu, introduced in section 3.4.6. The steady–state state transition probabilities α_k and β_k from (3.270) and (3.271), respectively, were estimated via simulation, and these are plotted in figures 4.12 and 4.13 for the silt and coarse sand bottom models.

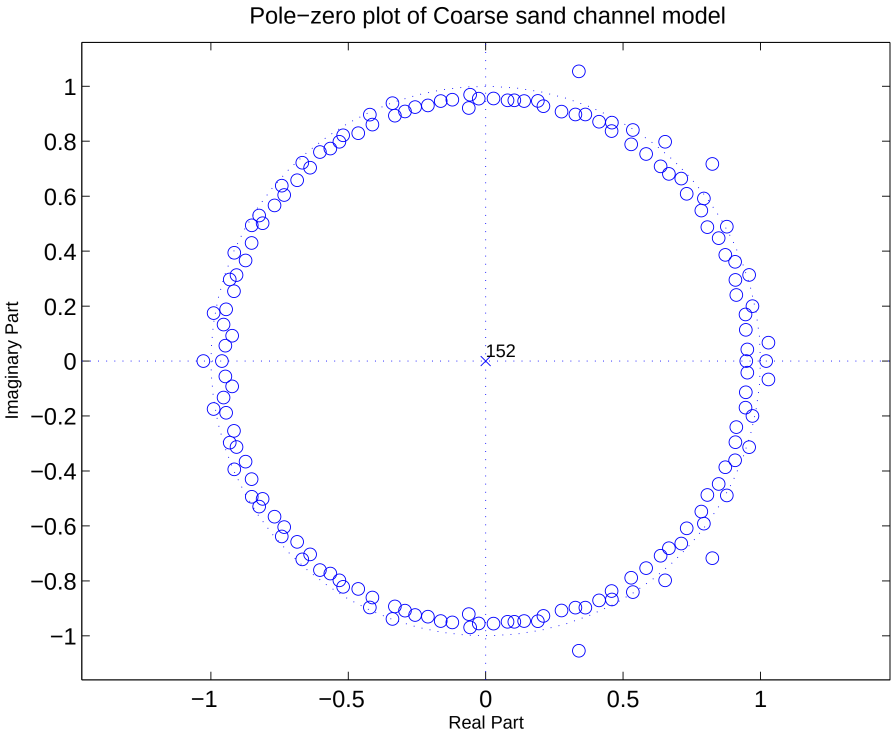

Figure 4.9. Pole–zero plot for coarse sand bottom.

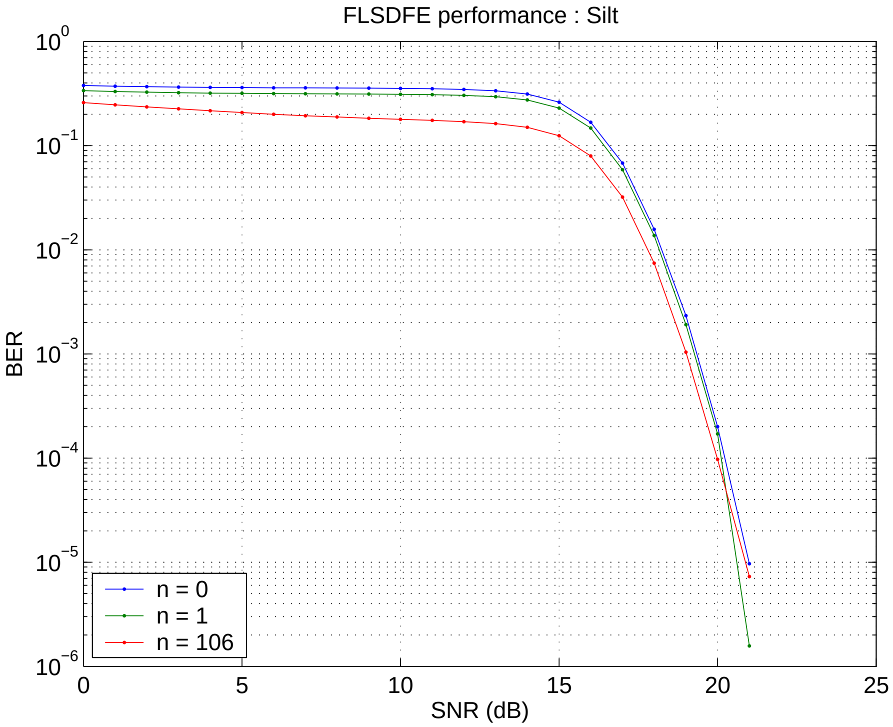

Figure 4.10. FLSDFE performance for silt bottom.

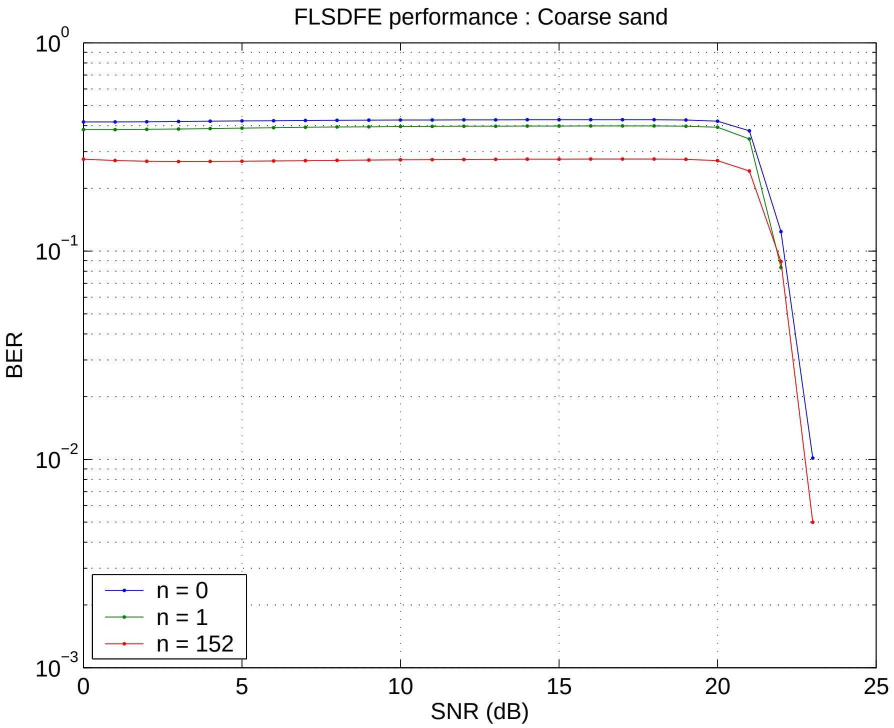

Figure 4.11. FLSDFE performance for coarse sand bottom.

The upper two curves in each figure consist of a set of overlaid plots of transition probabilities β_k (upper left curves) and α_k (upper right curves), where k runs over the set of states from the Choy and Beaulieu models (see figure 3.11). It is not possible to distinguish individual α_k and β_k data sets in the upper curves of figures 4.12 and 4.13, so they are presented in three–dimensional form in the matching lower plots. The colour–coding of the lower figures is the z direction, replacing the y direction of the upper plots.

Of immediate note is the presence of humps in some of the β_k curves for both channel models. That is, for some of the β_k, there is not a monotonic rise in transition probability with increasing SNR. Rather, the probability rises, reaches a peak, and then falls again at still higher input SNR. Referring to figure 3.11, it appears there are barriers to error recovery at both low and high SNR, with a partial window at intermediate SNR values. It is as though there were two modes for some of the β_k transitions, with peaks at different SNR values for different β_k. These barriers can be seen in the lower left plots of figures 4.12 and 4.13 as horizontal stripes of low state transition probability at high SNR.

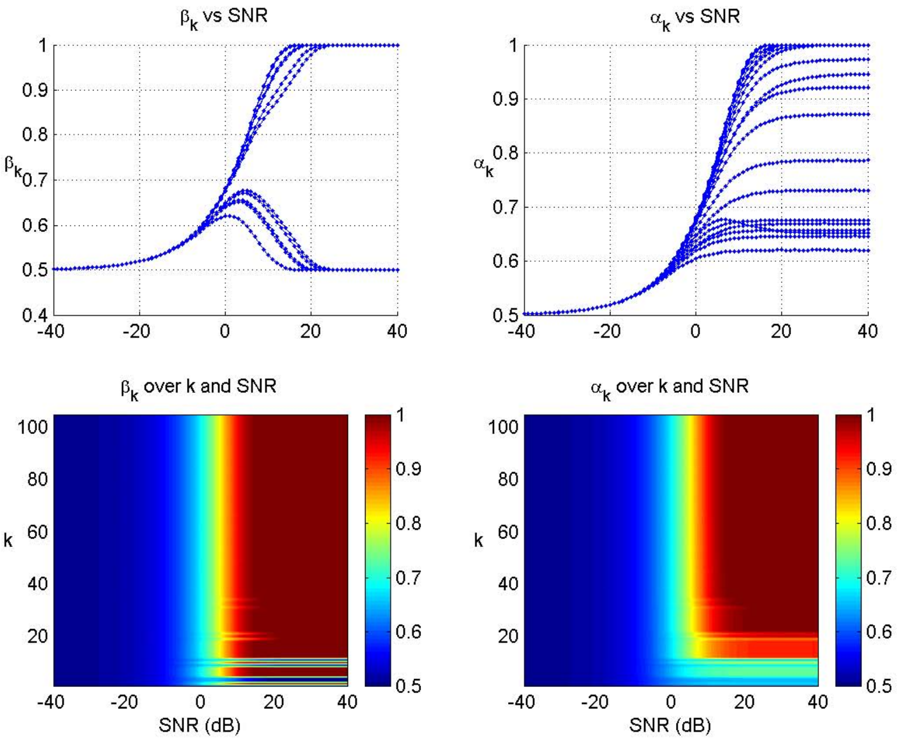

Figure 4.12. State transition probabilities for silt bottom.

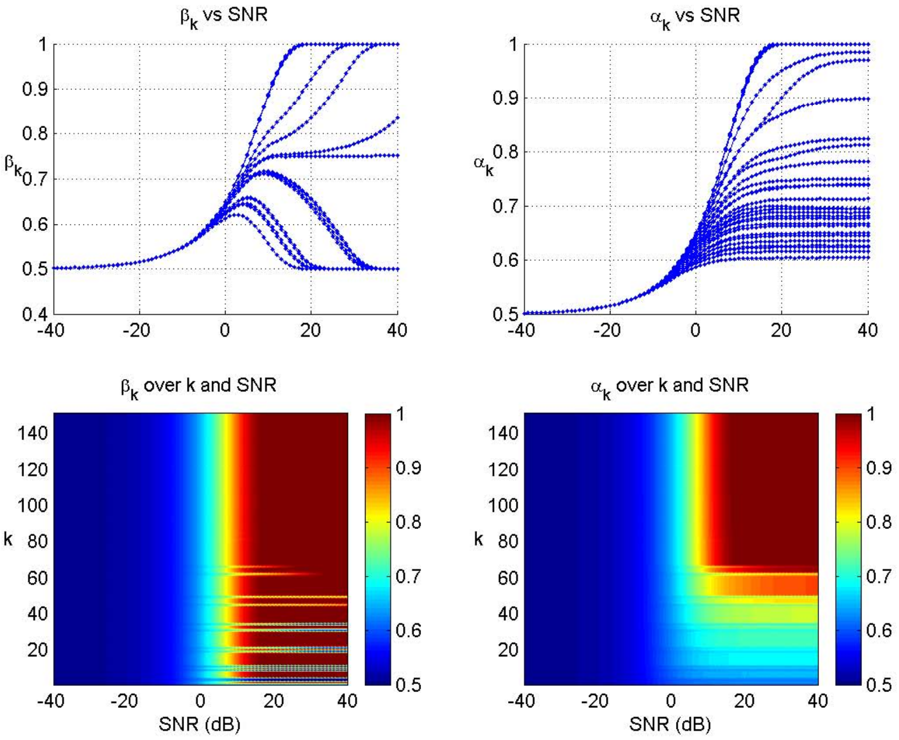

Figure 4.13. State transition probabilities for coarse sand bottom.

We note that the stripes occur at lower values of k, implying that there is more chance of getting stopped by a barrier while further from error recovery than there is while closer to error recovery. In the next section, we explore these 'resonance' phenomena further, with simple generic channel models.

4.3 Generic BPSK channel models

Section 4.2 highlighted some potential performance issues with the application of the FLSDFE algorithm to the equalisation of underwater acoustic communication channels. To better understand the complex error propagation dynamics, channel models with shorter memory N were constructed and analyzed. Specifically, we look at two purely linear channels, and two purely cubic channels, both of memory $N = 6$. Cubic channels arise in satellite communication, for example, where amplifiers are driven to saturation to maximize output power [11–14].

Section 4.3.1 introduces the baseband SISO channel models that will be used; bit error rate (BER) curves are presented in section 4.3.2; section 4.3.4 discusses error recovery times; section 4.3.3 provides curves of state transition probabilities for the single–distinct–errors model of Choy and Beaulieu [24]; and the mean and standard deviations of the lengths of error bursts are estimated in section 4.3.5.

4.3.1 Model description

We use a linear family of models,

$$y_t^{(1,m)} = \sum_{k_1=0}^{N} h_1^{(m)}(k_1) x_{t-k_1} + v_t, \tag{4.3}$$

and a cubic family,

$$y_t^{(3,m)} = \sum_{k_1=0}^{N} \sum_{k_2=k_1}^{N} \sum_{k_3=k_2}^{N} h_3^{(m)}(k_1, k_2, k_3) x_{t-k_1} x_{t-k_2} x_{t-k_3} + v_t, \tag{4.4}$$

with model index $m \in \{0, \ldots, N\}$ and Volterra kernels

$$h_1^{(m)}(k_1) = a_m \left(|k_1 - m| + 1\right)^{-2}, \quad \text{and} \tag{4.5}$$

$$h_3^{(m)}(k_1, k_2, k_3) = b_m \left(\left[\sum_{l=1}^{3} (k_l - m)^2 \right]^{\frac{1}{2}} + 1 \right)^{-2}. \tag{4.6}$$

Coefficients $a_m > 0$ and $b_m > 0$ were chosen such that

$$\sum_{k_1=0}^{N} \left(h_1^{(m)}(k_1)\right)^2 = 1, \quad \text{and} \tag{4.7}$$

$$\sum_{k_1=0}^{N} \sum_{k_2=k_1}^{N} \sum_{k_3=k_2}^{N} \left(h_3^{(m)}(k_1, k_2, k_3)\right)^2 = 1. \tag{4.8}$$

The additive noise sequence $\{v_t\}$ in each model is independent and identically distributed, zero-mean and Gaussian, with variance σ_v^2. For a random BPSK symbol sequence $\{x_t\}$ drawn from $\mathcal{A}_2 = \{-1, 1\}$, the signal–to–noise ratio for each model is given simply by

$$\text{SNR}_{\text{dB}} = -10 \log_{10} \sigma_v^2. \tag{4.9}$$

The linear models in (4.3) were chosen to give a quadratic decay in kernel coefficient $h_1^{(m)}(k_1)$ on either side of the peak at $k_1 = m$. Model 0 of family (4.3) is a minimum phase, linear FIR channel [43]. For such channels, it is possible to implement a stable and causal inverse filter to compensate for channel distortion. This is not the case for the other models in the linear family (4.3), indexed by $m \in \{1, \ldots, N\}$, since those channels are *non*minimum phase. For the family (4.3), therefore, we expect worse equalisation performance for the models indexed by $m \in \{1, \ldots, N\}$ than for the model indexed by $m = 0$. The cubic family (4.4) provides a 3–dimensional analogue of (4.3), and was included as a simple example of a nonlinear channel.

We restrict our attention in the remainder of this chapter to the specific models with indexes $m \in \{0, 3\}$. When $m = 0$, we get the minimum phase, linear FIR channel and its cubic cousin; and with $m = 3$, for the linear model we get a nonminimum phase channel with a peak kernel value at $h_1(3)$ instead of at $h_1(0)$, which means there is a delay of 3 samples before the maximum signal energy. This makes for an interesting contrast to the underwater acoustic channels of section 4.2.3, which had peak impulse response at zero lag.

4.3.2 BER versus SNR

Figures 4.14–4.17 give curves of bit–error–rate (BER) versus input signal–to–noise ratio (SNR), obtained by simulation. Figure 4.14 is for the linear, minimum phase channels ($m = 0, p = 1$); figure 4.15 is for the linear, nonminimum phase channels ($m = 3, p =$

1); figure 4.16 is for the cubic, minimum phase channels ($m = 0, p = 3$); and figure 4.17 is for the cubic, nonminimum phase channels ($m = 3, p = 3$). Note that there are seven curves in each figure, one for each lag $n \in \{0, \ldots, N\}$. The colour–coding follows that of the `plot` command of `MATLAB`$^{\text{TM}}$, namely blue, green, red, cyan, magenta, yellow and black, for lags $n = 0, \ldots, 6$, respectively.

Figures 4.14 and 4.15 show the data for the linear pair of models, those with Volterra series order $p = 1$. We note that, for both channels, the FLSDFE does no better at low SNR values than flipping a coin, with BER values of 0.5. At higher SNRs, however, two different pictures emerge.

In figure 4.14, at lag $n = 0$ (the blue curve) we get the rapid falloff of BER with SNR that is typical of a DFE [57]. We note from (4.5) with $m = 0$ that the kernels $h_1^{(0)}(k_1)$ fall off rapidly from their peak value $h_1^{(0)}(0)$, and so the DFE, operating at smoothing lag $n = 0$, is optimum here. For this channel model, no smoothing at all is better than any smoothing.

In figure 4.15, however, we observe that at high SNR some smoothing is better than no smoothing at all. The best result comes from matching the smoothing lag n to the peak delay $m = 3$, since from (4.5) the largest channel coefficient is $h_1^{(3)}(3)$. At high SNR the BER values of the linear, nonminimum–phase channel in figure 4.15 are orders of magnitude higher than those of the linear, minimum–phase channel in figure 4.14. Nevertheless, on the difficult linear, nonminimum–phase channel we see that at lags $n > 0$, the FLSDFE performs better than the DFE.

The cubic ($p = 3$) models in figures 4.16 and 4.17 have essentially the same pattern as their linear counterparts, with the exception that at high SNR the BER values in figure 4.16 are much larger than the comparable values in figure 4.14.

4.3.3 State transition probabilities

In figures 4.18–4.25 we present the average state transition probabilities α_k and β_k between the Ψ_k and Φ_k states in Choy and Beaulieu's single–errors state space model (section 3.4.6). These quantities were estimated via simulation from (3.270) and (3.271). Figures 4.18, 4.19, 4.22 and 4.23 give the data for the minimum phase channels; while figures 4.20, 4.21, 4.24 and 4.25 give the corresponding data for the nonminimum phase channels. Conversely, figures 4.18–4.21 give the data for the linear ($p = 0$) models; while figures 4.22–4.25 give the data for the cubic ($p = 3$) channels.

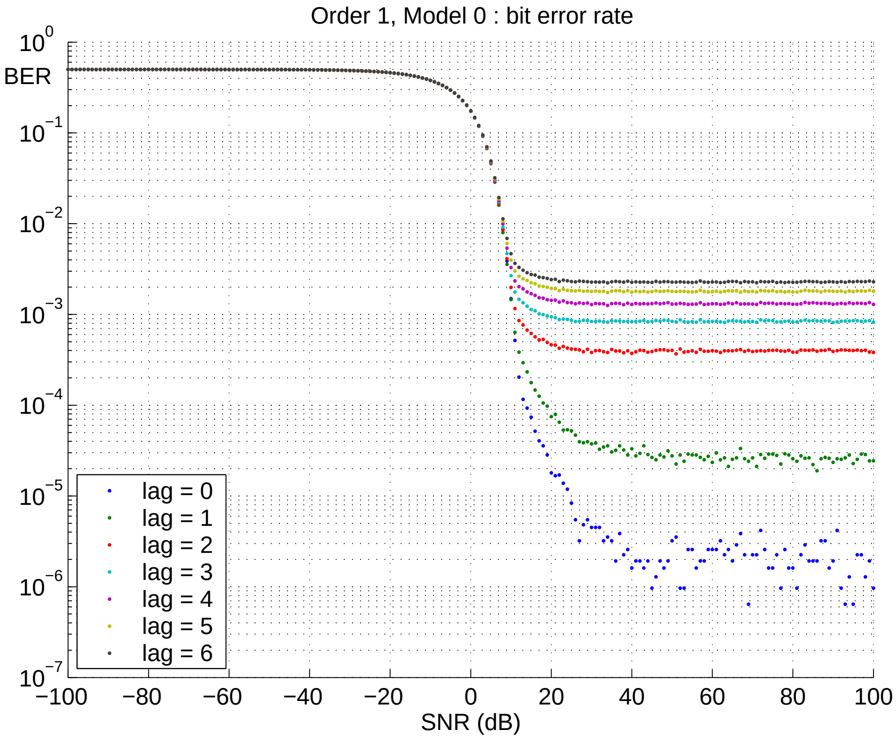

Figure 4.14. Bit error rate curves for $p = 1$ and $m = 0$.

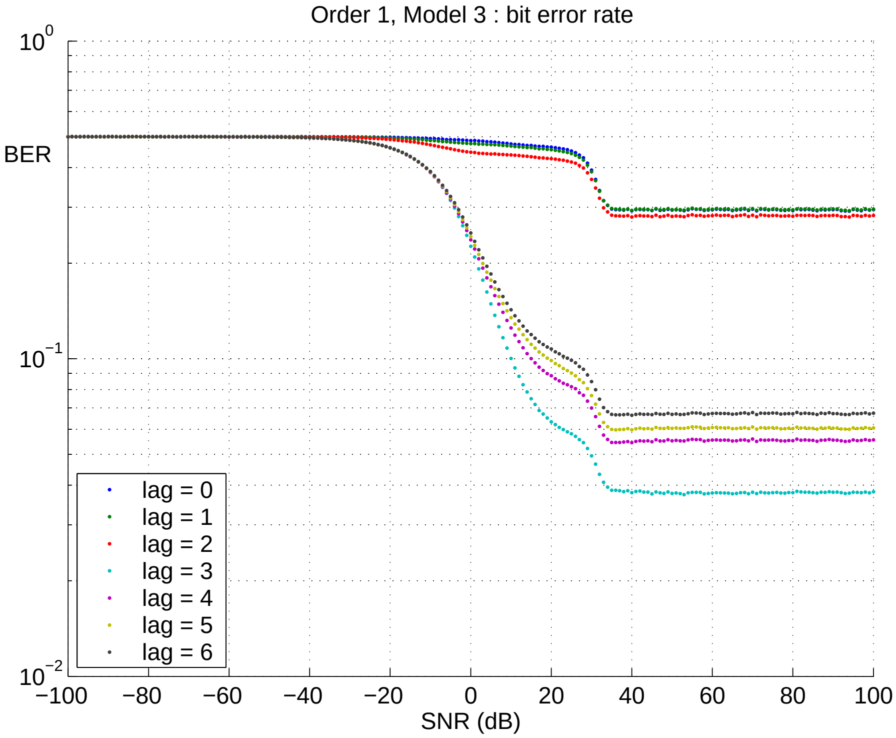

Figure 4.15. Bit error rate curves for $p = 1$ and $m = 3$.

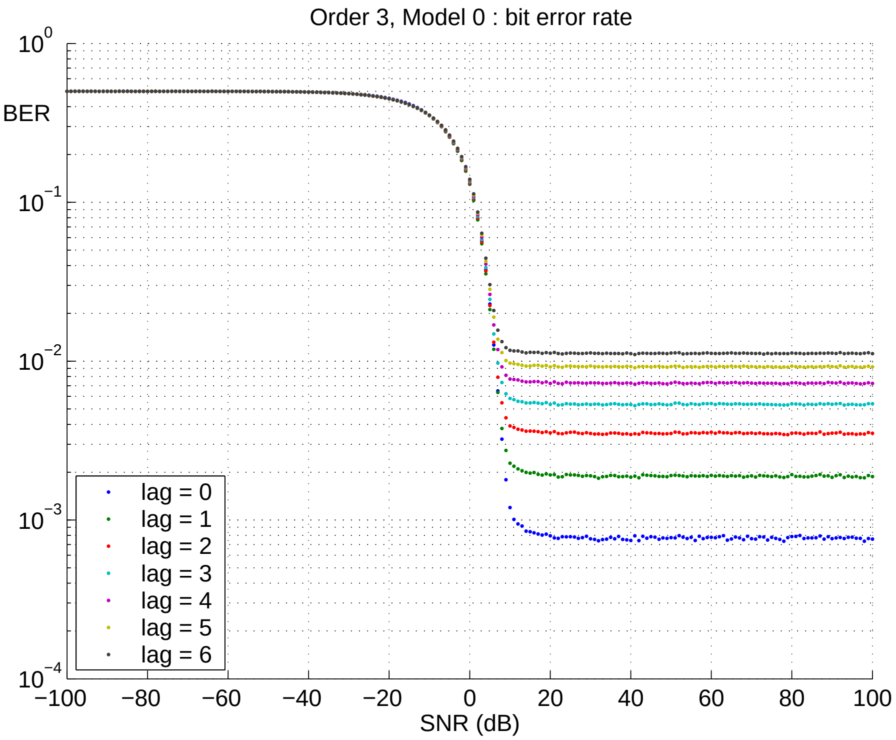

Figure 4.16. Bit error rate curves for $p = 3$ and $m = 0$.

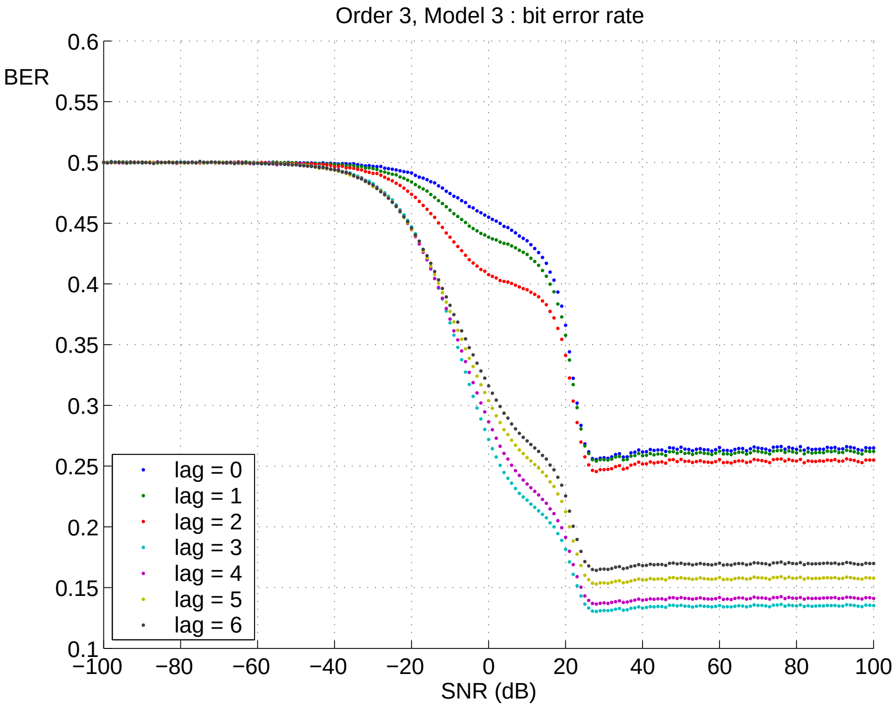

Figure 4.17. Bit error rate curves for $p = 3$ and $m = 3$.

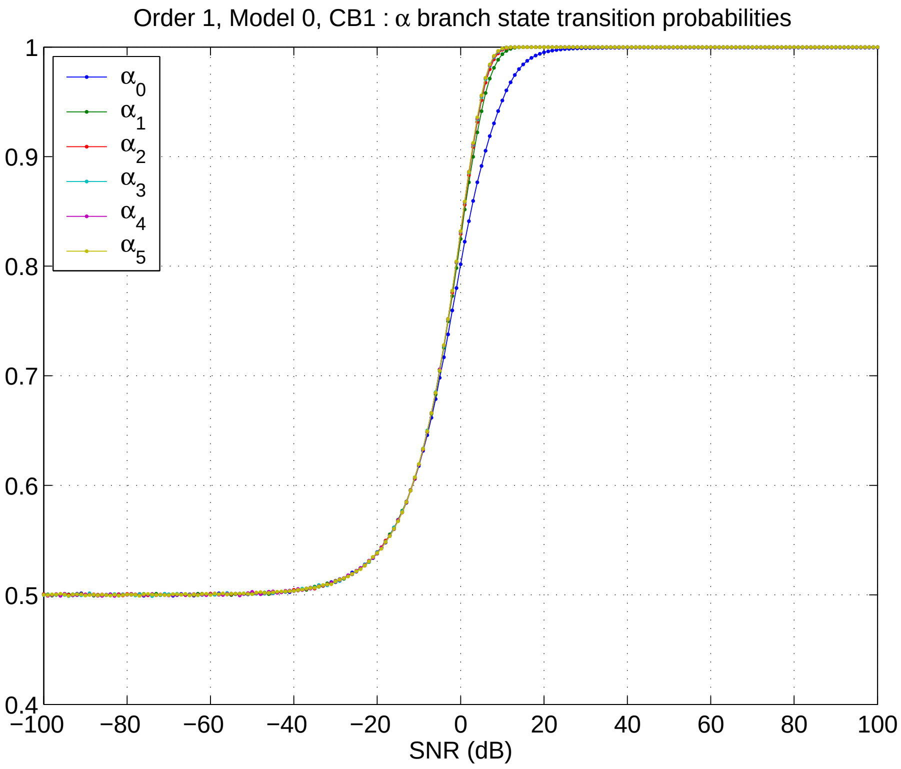

Figure 4.18. State transition probabilities α_k for $p = 1$ and $m = 0$.

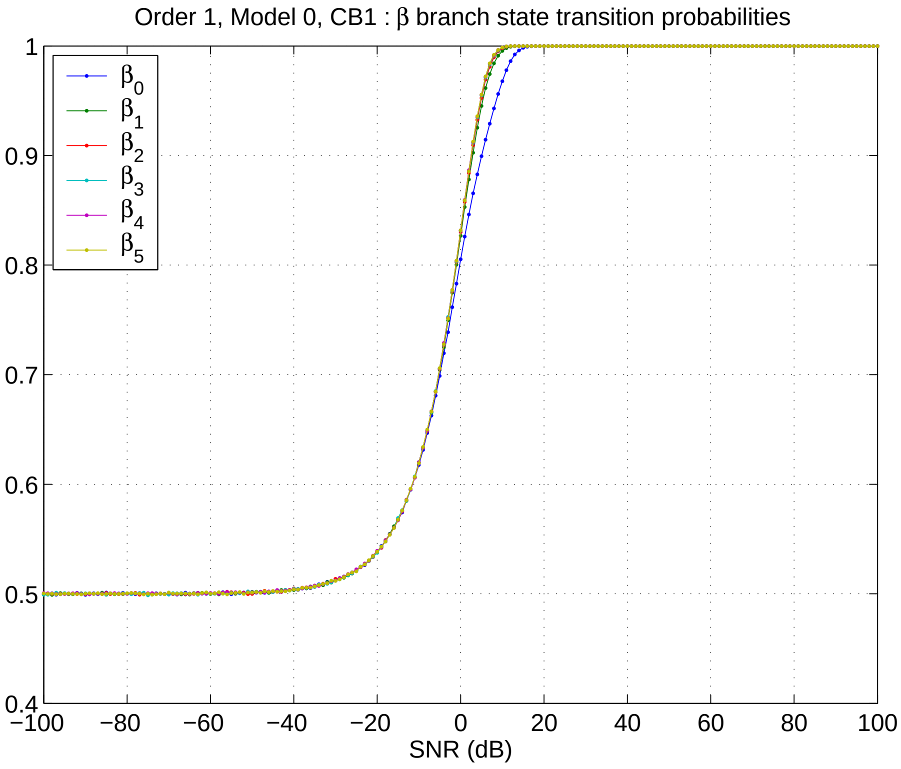

Figure 4.19. State transition probabilities β_k for $p = 1$ and $m = 0$.

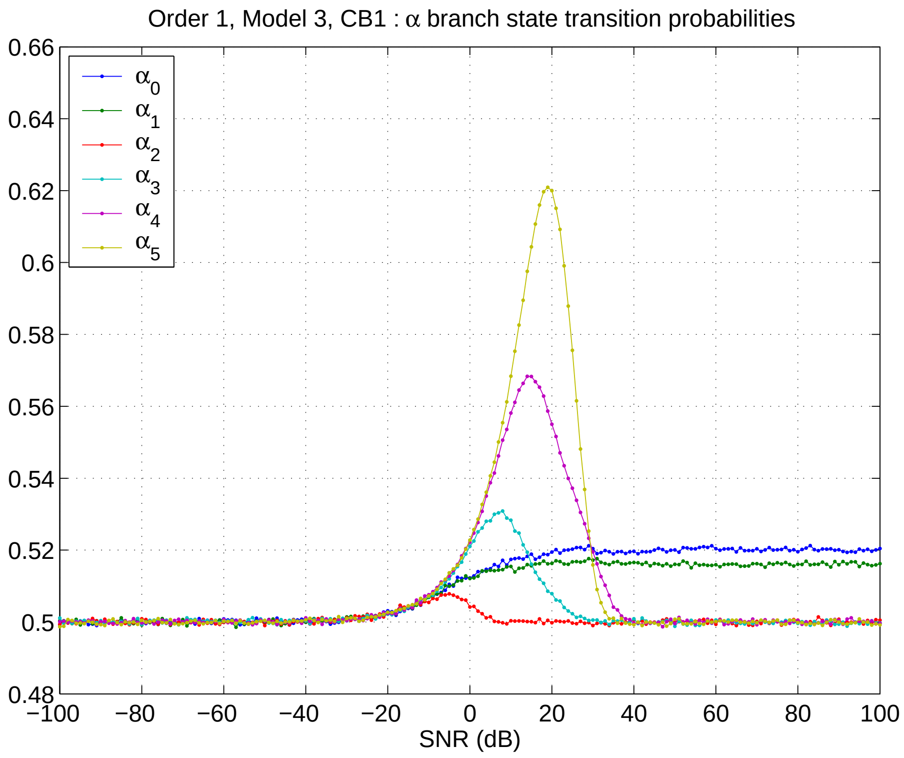

Figure 4.20. State transition probabilities α_k for $p = 1$ and $m = 3$.

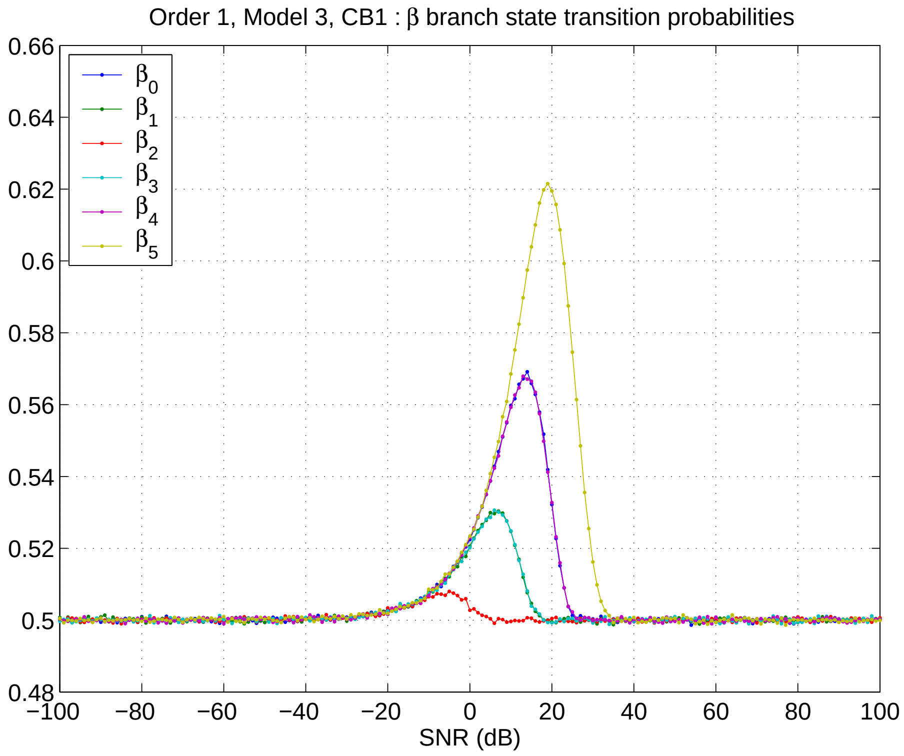

Figure 4.21. State transition probabilities β_k for $p = 1$ and $m = 3$.

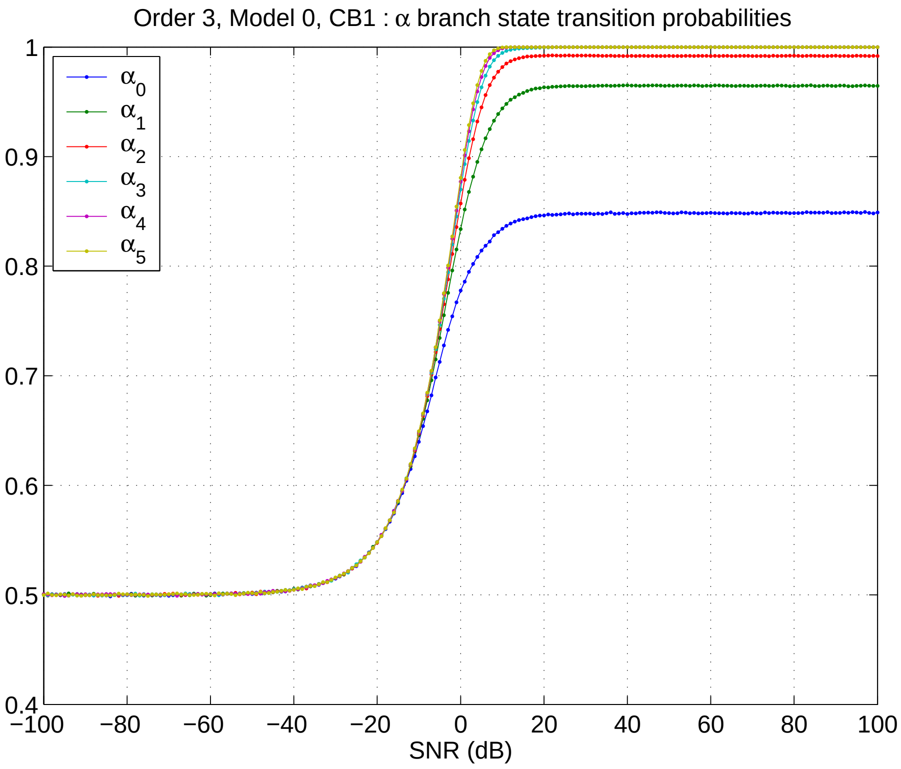

Figure 4.22. State transition probabilities α_k for $p = 3$ and $m = 0$.

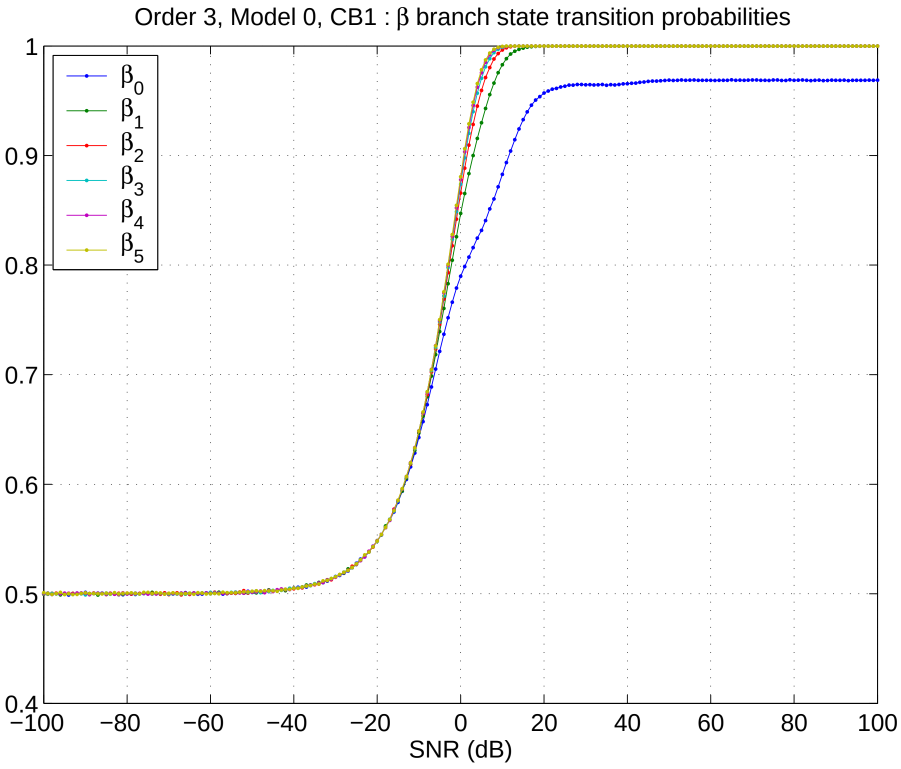

Figure 4.23. State transition probabilities β_k for $p = 3$ and $m = 0$.

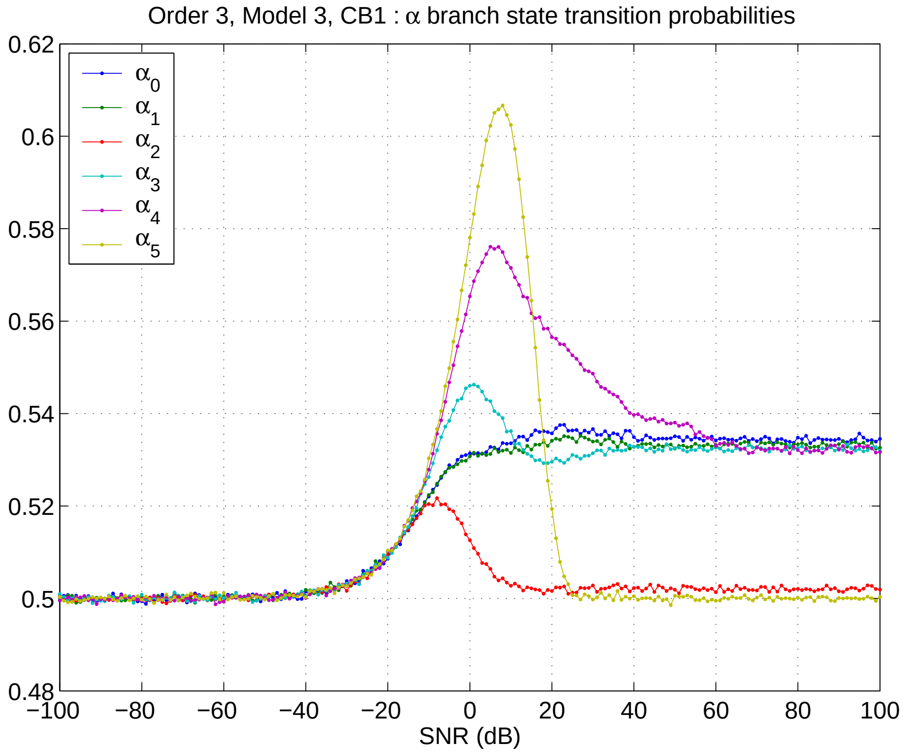

Figure 4.24. State transition probabilities α_k for $p = 3$ and $m = 3$.

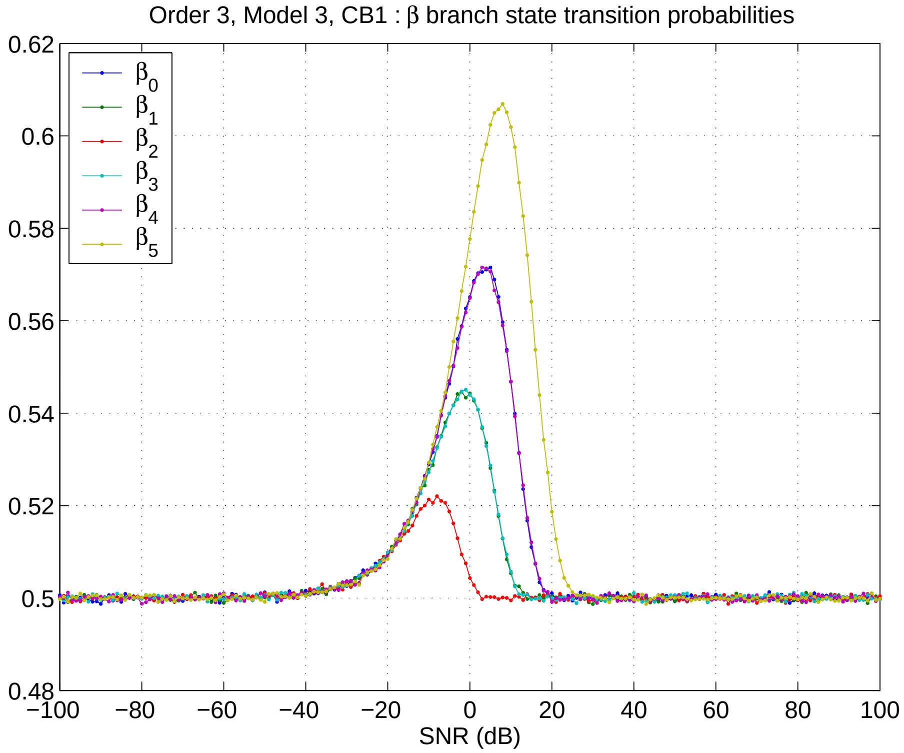

Figure 4.25. State transition probabilities β_k for $p = 3$ and $m = 3$.

As we *flow* along a branch, we progress towards the error–free state Γ. We conceptually think of flow along either the upper (β) branch or the lower (α) branch, although it is possible to cross from the upper to the lower branch. For convenience, in the following discussion we shall ignore the possibility of crossing between branches.

In figures 4.18, 4.19, 4.22 and 4.23, there is a steady rise in each transition probability, from a minimum of 1/2, at the noise–only limit (SNR $\to -\infty$), to a maximum value at the signal–only limit (SNR $\to \infty$). Thus, for the minimum phase channels, the upper (β) and lower (α) branches of the single–errors model state diagram 3.11 show a monotonic decrease in restriction of flow, with increasing SNR, towards the error–free state Γ.

For the cubic minimum phase channel ($p = 1$, $m = 3$), in figures 4.22 and 4.23, note that not all transition probabilities reach the value 1 at infinite SNR. Compare this with the corresponding figures 4.18 and 4.19 for the linear minimum phase channel ($p = 1$, $m = 0$), which show all transition probabilities reaching 1 at infinite SNR. For the cubic minimum phase channel there appear to be, in effect, *barriers* that constrict flow towards the error–free state along each branch. One such barrier is between states Φ_0 and Φ_1, for example. We see that the value of transition probability α_0 levels off at about 0.85 at high SNR. In order to flow along the α branch at high SNR, and eventually reach the error–free state Γ, this barrier must be overcome, with only an 85–percent chance of success, on average. Note, however, that the lower signal–only values of α_0, α_1 and β_0 for the cubic minimum phase channel do not significantly affect the signal–only recovery time. Figure 4.26 shows a value for R_0 that is indistinguishable from the theoretical value $R_0 = N = 6$ attained by the linear minimum phase channel.

Figures 4.20, 4.21, 4.24 and 4.25 show a very different behaviour for the nonminimum phase channels. For these channels, the transition probability curves show a kind of 'resonance' in the transition probability versus input SNR, at which the probability rises to a peak value, before dropping again. Of note is the fact that the peaks occur at different SNR values along a given branch, but they are correlated across branches. Thus, for example, the linear model has a strong resonance peak at about 19 dB for both α_5 and β_5, meaning that the transitions from Φ_5 to Γ, or Ψ_5 to Γ, are most likely at 19 dB. The previous peaks, for α_4 and β_4, however, occur at the lower SNR of 15 dB.

In contrast to the transition probability curves for the minimum phase channels, note that the asymptotic transition probabilities for the nonminimum phase channels, at infinite SNR, are close to 0.5 for the α branch, and they attain 0.5 for the β branch. That is, apart from the resonance region, most of the behaviour of the FLSDFE is noiselike. At high SNR, it appears that the decision–feedback part of the FLSDFE algorithm is not effective on the nonminimum phase channels in this subsection. Since FLSDFE smoothing outputs $\hat{x}_{t-n|t}$, $n \in \{1, \ldots, N\}$, are dependent on the FLSDFE filtering outputs, $\hat{x}_{t|t}$, the FLSDFE is not expected to perform as well as it did on the minimum phase channels at any smoothing lag. This is confirmed by figures 4.15 and 4.17, which show high bit–error rates for the nonminimum phase channels across a wide range of input SNR values and smoothing lags.

4.3.4 Error recovery time

The state–transition diagram for the single–distinct–errors model of Choy and Beaulieu was given earlier as figure 3.11. For $N = 6$, this model has the 13 mutually exclusive states shown in table 4.2 below.

Table 4.2. Choy and Beaulieu's single–distinct–errors model for a channel of memory $N = 6$ (linear or nonlinear), BPSK signalling, and filtering only ($n = 0$).

s_t	e_{t-1}	e_{t-2}	e_{t-3}	e_{t-4}	e_{t-5}	e_{t-6}	e_{t-7}	e_{t-8}	e_{t-9}	e_{t-10}	e_{t-11}	e_{t-12}
Ψ_0	E	O	O	O	O	O	X	X	X	X	X	X
Ψ_1	O	E	O	O	O	O	O	X	X	X	X	X
Ψ_2	O	O	E	O	O	O	O	O	X	X	X	X
Ψ_3	O	O	O	E	O	O	O	O	O	X	X	X
Ψ_4	O	O	O	O	E	O	O	O	O	O	X	X
Ψ_5	O	O	O	O	O	E	O	O	O	O	O	X
Φ_0	E	X	X	X	X	X	X	X	X	X	X	X
Φ_1	O	E	X	X	X	X	X	X	X	X	X	X
Φ_2	O	O	E	X	X	X	X	X	X	X	X	X
Φ_3	O	O	O	E	X	X	X	X	X	X	X	X

Continued on next page

Table 4.2 – concluded from previous page

s_t	e_{t-1}	e_{t-2}	e_{t-3}	e_{t-4}	e_{t-5}	e_{t-6}	e_{t-7}	e_{t-8}	e_{t-9}	e_{t-10}	e_{t-11}	e_{t-12}
Φ_4	O	O	O	O	E	X	X	X	X	X	X	X
Φ_5	O	O	O	O	O	E	X	X	X	X	X	X
Γ	O	O	O	O	O	O	X	X	X	X	X	X

In this table, s_t refers to a particular error state, chosen from the set of states

$$\mathbf{S} = \{\Psi_0, \ldots, \Psi_{N-1}, \Phi_0, \ldots, \Phi_{N-1}, \Gamma\}; \tag{4.10}$$

e_{t-k} is an error symbol at time $t-k$; E indicates a filtering error ($\hat{x}_{t-k|t-k} \neq x_{t-k}$); O means no error ($\hat{x}_{t-k|t-k} = x_{t-k}$); and X means we do not care whether or not there was a filtering error. E is colour–coded yellow; an error–free run of length $N-1$ is colour–coded blue; and the complement of this, a run of length $N-1$ with at least one error, is colour–coded grey. Errors e_{t-k} are all X for $k \geq 2N$.

As discussed in section 3.4.6 and as defined by Choy and Beaulieu [24], the recovery time R_0 is the mean number of steps taken to first reach Γ, commencing in Ψ_0. Figure 4.26 below gives the error recovery time, as computed from the formula [24, eq. 21]

$$R_0 = \frac{1}{\alpha_5}\left[1 + \sum_{i=0}^{4}\prod_{j=0}^{i}\beta_j + \left(1 - \prod_{j=0}^{4}\beta_j\right)\left(1 + \sum_{i=0}^{4}\prod_{j=i}^{4}\frac{1}{\alpha_j}\right)\right]. \tag{4.11}$$

α_j and β_j are the steady–state state transition probabilities shown in figure 3.11.

Γ is the error–free state, with N correct decisions. Error recovery begins in Φ_0, with the corruption of an error–free channel by a recent filtering error. Progress towards Γ involves transition over the states $\Phi_0 \to \Phi_1 \to \cdots \to \Phi_5 \to \Gamma$, which is the upper ($\beta$) branch of figure 3.11; or the states $\Psi_0 \to \Psi_1 \to \cdots \to \Psi_5 \to \Gamma$, which is the lower ($\alpha$) branch of figure 3.11. Further filtering errors along the way, however, can force a jump back to either Φ_0 or Ψ_0, the start of the β and α branches, respectively, thereby delaying error recovery.

Note from figure 4.26 that we reproduce the noise–only limit (SNR $\to -\infty$) for all models, as per Choy and Beaulieu's analytical result $2(2^N - 1) = 126$. As for the signal–only limit (SNR $\to \infty$), however, we observe Choy and Beaulieu's theoretical result of $N = 6$ just for the minimum phase channels.

The nonminimum phase channels have a long mean recovery time at infinite SNR, and an optimum SNR at which recovery time is least. At either side of this optimum,

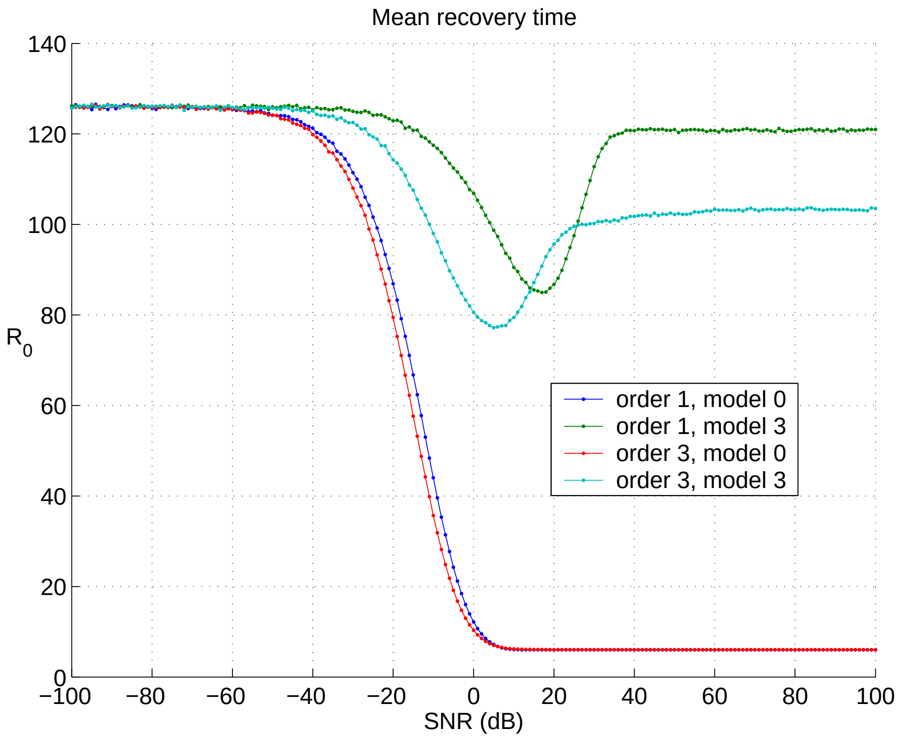

Figure 4.26. Mean time to recover from an error, R_0, from Choy and Beaulieu's single–distinct–errors state space model.

however, the FLSDFE is prone to longer error bursts. There is a tantalising connection with the notion of stochastic resonance, in which the addition of noise may *lower* the recovery time, by driving the SNR downwards towards the optimum value.

Choy and Beaulieu considered filtering only, but their models are applicable to an analysis of the FLSDFE in Volterra channels. We note from chapter 2 that the output $\hat{x}_{t-n|t}$ of the FLSDFE, at smoothing lags $n > 0$, is functionally dependent on the filtered values $\hat{x}_{t|t}$. A proper study of FLSDFE smoothing errors $x_{t-n} - \hat{x}_{t-n|t}$ must therefore begin with a study of DFE filtering errors $x_t - \hat{x}_{t|t}$ [61].

4.3.5 Error bursts

Further elucidation of the different performance of the FLSDFE algorithm on the linear and cubic, minimum– and nonminimum–phase channels is obtained from a study of the phenomenon of error bursts. Via simulation we produced figures 4.27–4.30 below, which show the mean and standard deviations of the lengths of error bursts. An error burst of length m is such that $\hat{x}_{t|t} = x_t$, $\hat{x}_{t-k|t-k} \neq x_{t-k}$ and $\hat{x}_{t-m-1|t-m-1} = x_{t-m-1}$, for $k \in \{1, \ldots, m\}$.

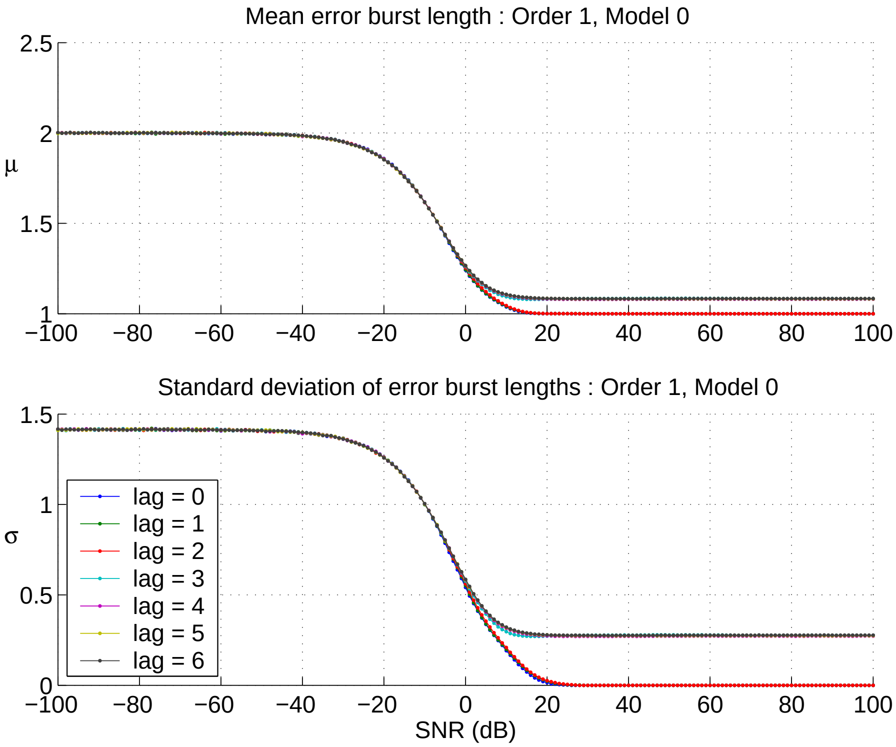

Figure 4.27. Error burst statistics for $p = 1$ and $m = 0$.

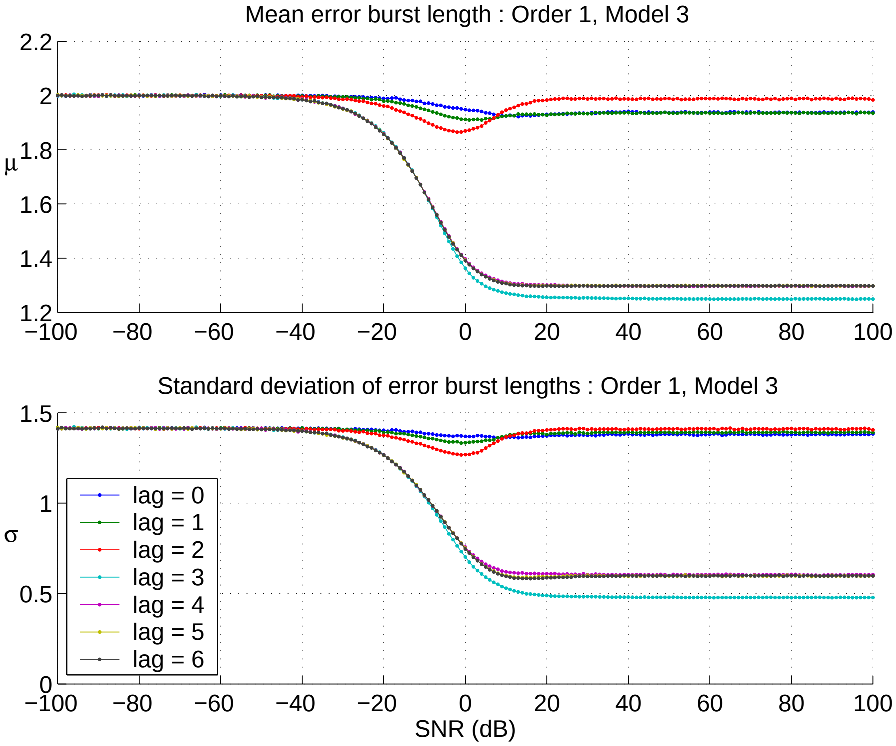

Figure 4.28. Error burst statistics for $p = 1$ and $m = 3$.

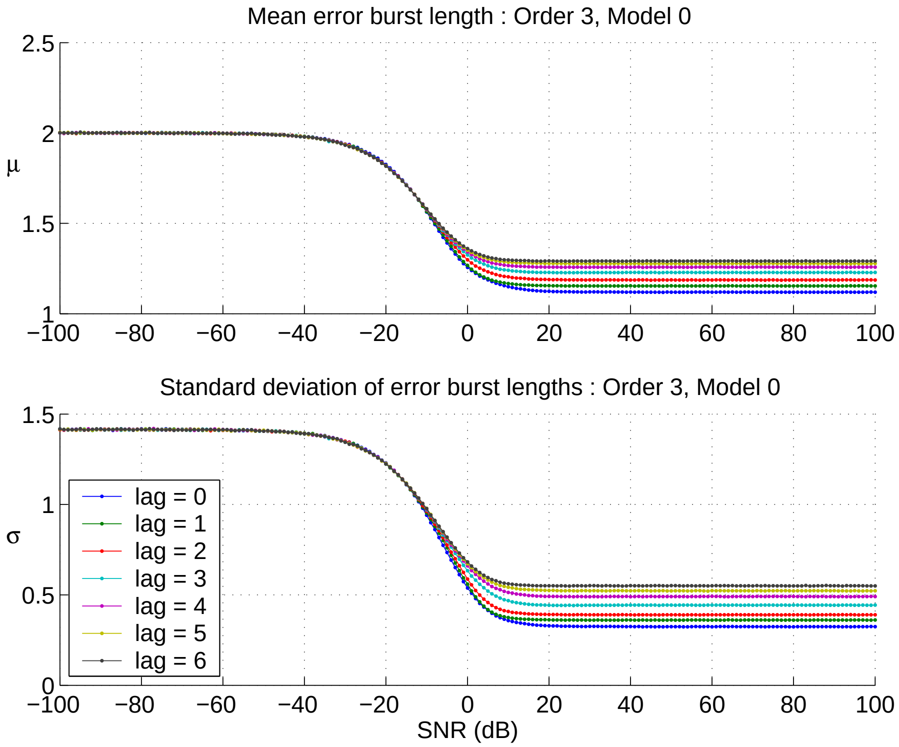

Figure 4.29. Error burst statistics for $p = 3$ and $m = 0$.

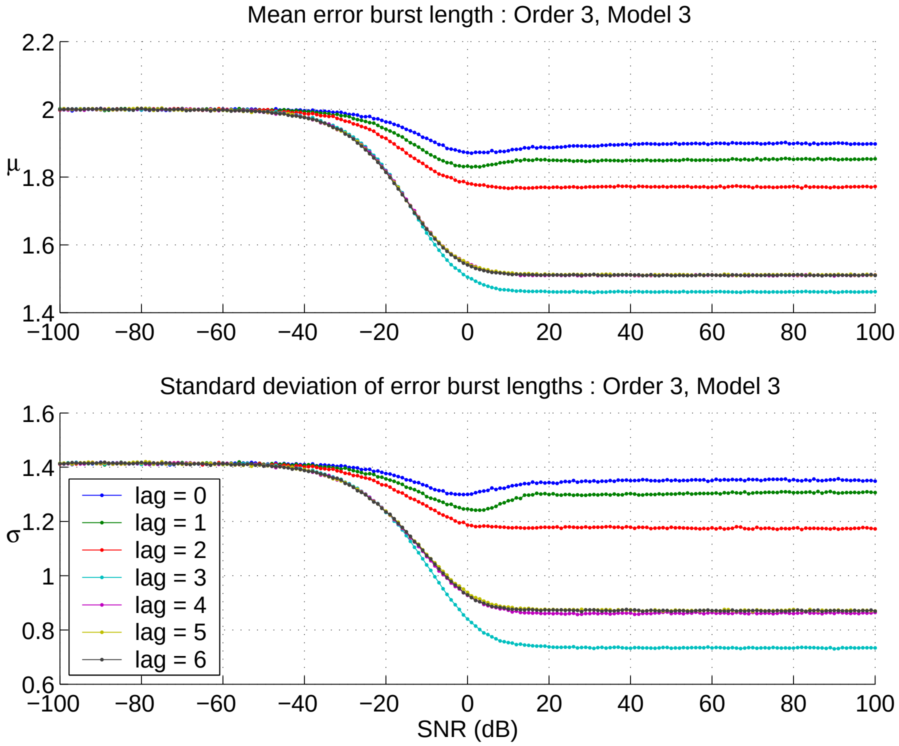

Figure 4.30. Error burst statistics for $p = 3$ and $m = 3$.

In the noise–only limit, the mean error burst length across all four models, and at all smoothing lags, is two samples. The corresponding standard deviation is $\sqrt{2}$, or about 1.4. This is shown as follows. The symbol values x_{t-n} and FLSDFE outputs $\hat{x}_{t-n|t}$ are, in this case, uncorrelated and independent Bernoulli random variables, with probability $\frac{1}{2}$ of assuming either of their values from alphabet $\mathcal{A}_2 = \{-1, 1\}$. We thus have the expected value of the error burst length L given by

$$\mathbb{E}(L) = \sum_{l=1}^{\infty} l \left(\frac{1}{2}\right)^l = 2, \tag{4.12}$$

and the standard deviation given by

$$\sqrt{\mathbb{E}((L - \mathbb{E}(L))^2)} = \sum_{l=1}^{\infty} (l-2)^2 \left(\frac{1}{2}\right)^l = \sqrt{2}. \tag{4.13}$$

At the signal–only limit for the linear, minimum phase channel ($p = 1$, $m = 0$ in figure 4.27), we find that the mean error burst length is about one sample at smoothing lags 0, 1 and 2, but jumps to about 1.1 samples at smoothing lags 3, 4 and 5. There is a corresponding jump in the standard deviation, being close to zero at lags 0, 1 and 2. For this channel, there seem to be two 'modes' at the two sets of lags, with slightly worse equalisation performance at the higher lags.

The signal–only limit for the cubic, minimum phase channel ($p = 3$, $m = 0$ in figure 4.29) shows a more dispersed range of mean error burst lengths, from a minimum of about 1.15, at lag 0, to a maximum of about 1.35, at lag 6. There is a gradual increase of mean burst length with increasing lags, reflecting the fact that we are moving away from the optimum smoothing lag at $n = m = 0$. Note that the standard deviation mirrors this pattern as well.

For the linear, nonminimum phase channel ($p = 1$, $m = 3$ in figure 4.28) we see a more complicated pattern of error burst statistics. At the signal–only limit, there is an increase of mean burst length as the lag increases to 2, followed by a sudden drop at the optimum lag $n = m = 3$. At still higher smoothing lags, the mean burst length increases slightly, though still significantly below that of smaller, sub–optimal lags $n < m$. At intermediate SNR values, we note a dip in mean burst length at lags 0, 1 and 2 and a reversal of the high–SNR pattern. Thus, at 0 dB, for instance, we see that the mean run length decreases from lags 0 to 2. The standard deviations mirror the means.

For the cubic, nonminimum phase channel ($p = 3$, $m = 3$ in figure 4.30), we note that the high–SNR means and standard deviations decrease monotonically towards the optimum smoothing lag $n = m = 3$, then (slightly) increase past that. There is a slight dip at intermediate SNR values. This appears to be related to the resonances observed in section 4.3.3.

4.4 Conclusion

The behaviour of the FLSDFE algorithm on nonminimum phase, nonlinear channels is complicated. There is potential benefit in applying fixed–lag smoothing to a variety of channels, and smoothing seems to improve the results. One caveat, however, is that it is important to choose an optimum smoothing lag, one that keeps error bursts to a minimum and for which the bit error rate is as low as can be expected. Some of the channels treated in this chapter are difficult to equalise by decision–feedback equalisation, even with the application of fixed–lag smoothing, but the FLSDFE definitely shows promise. There may be other equalisation techniques that are more suited to channels such as the cubic, nonminimum phase channel discussed above, but the FLSDFE algorithm is a simple extension to the DFE that, when used wisely, leads to improved equalisation over that of the DFE alone. The resonance phenomena observed in section 4.3.3 are new to the DFE literature.

Chapter 5

Conclusions and Future Work

This thesis covered some of the theory and application of the FLSDFE algorithm, first introduced by Perreau *et al.* [52]. The FLSDFE incorporates a fixed–lag smoother into a decision–feedback equaliser. According to Anderson [4], '... at high signal to noise ratios, smoothing gives greater improvement over filtering than at low signal to noise ratios.' The rationale behind the FLSDFE is thus that by delaying symbol estimation until more channel measurements are available, the FLSDFE may produce less symbol errors than the DFE alone would. In this thesis, we investigate the relative performance of fixed–lag smoothing in some detail, with mixed results.

In chapter 2 we provided an alternative derivation of the FLSDFE algorithm, showing the specific forms for BPSK, MPSK and MQAM signalling on SISO channels with input–output relationships in the form of a Volterra series. In particular, the combination of the FLSDFE algorithm and the use of Volterra channel models is new. This work extended the linear FIR channel models of Perreau *et al.*, and provides a guide to the application of the FLSDFE on other nonlinear channels. In addition, the probability distribution functions of the FLSDFE estimator $\hat{X}_{t-n|t}$ were discussed in sections 2.3.3 and 2.4.3, and the explicit form of the distribution was given for the simpler BPSK case. The distribution of $\hat{X}_{t-n|t}$ has not appeared in the FLSDFE literature before.

Chapter 3 introduced state space models that proved useful in modelling the transient and limiting temporal dynamics of error propagation. We discussed atomic state space models in sections 3.1–3.3, with different models required for the filtering–only and smoothing–only cases. These models were based on a similar model of Kennedy and Anderson [39], with the difference being that the models in this thesis incorporate all data up to the current time t, rather than the previous time $t-1$. The smoothing–only model in section 3.3 is new.

For the filtering–only model in section 3.2 we provided a novel formulation for the state transition probability matrix, illustrated for the case of BPSK signalling. This matrix is hierarchically structured and sparse. No general results were found for the stationary distribution vector for either the filtering–only or the smoothing–only atomic state space models. Within the context of the error recovery time of Choy and Beaulieu [24], some novel connections with Fibonacci and related series were made for the filtering–only atomic state space model. Some conjectures were raised concerning these observations. In addition, connections were drawn between the filtering–only model and the theory of integer partitions.

In section 3.4 we discussed aggregated state space models, where the atomic states were formed into mutually disjoint subsets. We illustrated the existence of a unique optimum aggregation for the case of BPSK signalling on a channel of memory $N = 1$, with filtering only, in section 3.4.4. This aggregation was found by brute force, using set partitions and restricted growth strings. The method used appears to be a novel approach to the problem of finding an aggregation of an atomic state space that satisfies the finite–state Markov process (FSMP) property. The unique optimal aggregation exactly models the transient dynamics of the underlying atomic state space model, but does so with fewer aggregated states. It is not known in general whether unique optimal aggregations exist for the filtering–only and smoothing–only atomic state space models. The brute force method used in the example of this thesis is too prohibitive for general use, related to the exponential growth of the size of Bell numbers (which give the number of set partitions of a given set of atomic states).

A suboptimal aggregation was illustrated in section 3.4.5, again for the simple case of BPSK signalling, the filtering–only atomic state space model, and a channel memory of $N = 1$. This aggregation also satisfied the FSMP property, and thus could exactly model the transient dynamic behaviour of the atomic state space model, but it did so with six states instead of 16 (the optimum used four states). Moreover, this aggregation was related to the optimum in that four of the states were formed by splitting two of the optimal states. In addition, the suboptimum aggregation had the neat property that each of the states was a subset of one of the states of a model of Choy and Beaulieu [24], introduced in their work on DFE error recovery time. Thus we demonstrated a bridge between the transient models of this thesis and the asymptotic steady–state models of Choy and Beaulieu.

While Choy and Beaulieu's aggregated states did not satisfy the FSMP property, the suboptimal aggregation in this thesis did. Although we only demonstrated the suboptimal model for the $N = 1$ case, it would be interesting to see whether suboptimal models can be found for more general filtering–only and smoothing–only atomic state space models that are closely related to the models of Choy and Beaulieu, and yet satisfy the FSMP property. This is a subject for future work.

In chapter 4 we applied the FLSDFE algorithm to some channels and showed that the FLSDFE had both benefits and disadvantages. In particular, we saw that the behaviour of the FLSDFE was closely coupled to that of the ordinary DFE, so that some channels were still difficult to equalise. Nonetheless, for those difficult (nonlinear, nonminimum phase) channels, we observed that the FLSDFE provided a performance advantage over the DFE alone. That is, the application of fixed–lag smoothing to the outputs of the DFE was of benefit on those channels. Finally, we observed that it was possible to glean some insight into the complex transient behaviour of the FLSDFE algorithm by using the single–errors state space model of Choy and Beaulieu. In particular, we observed 'resonance' behaviour and barriers to error recovery. These phenomena result from the fact that the FLSDFE is a nonlinear process with DFE feedback.

The behaviour of the FLSDFE algorithm on general Volterra channels is that of a complex nonlinear stochastic dynamical process. This complexity makes it difficult to analyze, in general, although we can make headway in specific cases. What is lacking, however, is an over–arching and powerful theory to describe the dynamics of the FLSDFE algorithm on arbitrary channels. This is an area of future research that may prove useful. Of note is the observation that during the analysis of the performance of the FLSDFE algorithm in section 2.3, for example, we saw the presence of generalized feedback errors such as

$$x_{t-k-k_1} x_{t-k-k_2} x^*_{t-k-k_3} - \hat{x}_{t-k-k_1} \hat{x}_{t-k-k_2} \hat{x}^*_{t-k-k_3}, \tag{5.1}$$

involving three separate times. Characterisation of feedback errors such as these would imply the use of a three–dimensional lattice. Most published efforts to model and deal with the phenomenon of error–feedback in decision–feedback equalisation (with and without fixed–lag smoothing) have been concerned merely with linear (first–order) forms such as

$$x_{t-k-k_1} - \hat{x}_{t-k-k_1}. \tag{5.2}$$

Insufficient effort seems to have been placed on the more difficult and higher–dimensional problem of dealing with generalized decision–feedback errors, which add complexity to the stochastic dynamics of decision–feedback equalisation algorithms.

The FLSDFE algorithm discussed in this thesis did not feed back smoothed outputs $\hat{x}_{t-n|t}$ for lags $n > 0$. It is unknown whether feeding these quantities back into the FLSDFE might improve the performance. The results, in any case, might vary from channel to channel. Nevertheless, it would be an interesting area of future research to quantify the performance of these variant FLSDFE algorithms.

In tables 5.1–5.12 on the following pages is an illustration of the operation of a modified form of the FLSDFE algorithm presented in this thesis. The description of the proposed algorithm is contained in the table captions. The SISO channel has memory $N = 2$ and the smoothing lag n takes on each of the values in the set $\{0, \ldots, N\}$. The input–output relationship is unspecified, although we assume an additive noise form,

$$Y_t = F_t(X_t, X_{t-1}, X_{t-2}) + V_t, \tag{5.3}$$

so that the estimator $\hat{X}_{t-n|t}$ is obtained from the least–squares criterion used to develop the specific MPSK, MQAM and BPSK estimators shown in chapter 2.

Table 5.1. The first step in a modified FLSDFE algorithm, wherein smoothing outputs $\hat{x}_{t-n|t}$ are fed back. No estimates of x_{t-5} and x_{t-6} are available, so set $\hat{x}_{t-5|t-5}$ and $\hat{x}_{t-6|t-6}$ to zero. Symbol x_{t-4} is estimated with $\hat{x}_{t-4|t-4}$, which is a function of y_{t-4}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-4}$	0	0

Table 5.2. When y_{t-3} becomes available, estimate x_{t-3} with $\hat{x}_{t-3|t-3}$, which is a function of $\hat{x}_{t-4|t-4}$ and y_{t-3}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-4}$	0	0
y_{t-3}	$\hat{x}_{t-3\|t-3}$	$\hat{x}_{t-4\|t-4}$	0

Table 5.3. Revise estimate of x_{t-4}, using FLSDFE with smoothing lag $n = 1$. $\hat{x}_{t-4|t-3}$ replaces $\hat{x}_{t-4|t-4}$, and is a function of $\hat{x}_{t-3|t-3}$, y_{t-3} and y_{t-4}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-3}$	0	0
y_{t-3}	$\hat{x}_{t-3\|t-3}$	$\hat{x}_{t-4\|t-3}$	0

Table 5.4. When y_{t-2} becomes available, estimate x_{t-2} with $\hat{x}_{t-2|t-2}$, which is a function of $\hat{x}_{t-3|t-3}$, $\hat{x}_{t-4|t-3}$ and y_{t-2}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-3}$	0	0
y_{t-3}	$\hat{x}_{t-3\|t-3}$	$\hat{x}_{t-4\|t-3}$	0
y_{t-2}	$\hat{x}_{t-2\|t-2}$	$\hat{x}_{t-3\|t-3}$	$\hat{x}_{t-4\|t-3}$

Table 5.5. Revise estimate of x_{t-3}, using FLSDFE with smoothing lag $n = 1$. $\hat{x}_{t-3|t-2}$ replaces $\hat{x}_{t-3|t-3}$, and is a function of $\hat{x}_{t-2|t-2}$, $\hat{x}_{t-4|t-3}$, y_{t-2} and y_{t-3}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-3}$	0	0
y_{t-3}	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-3}$	0
y_{t-2}	$\hat{x}_{t-2\|t-2}$	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-3}$

Table 5.6. Revise estimate of x_{t-4}, using FLSDFE with smoothing lag $n = 2$. $\hat{x}_{t-4|t-2}$ replaces $\hat{x}_{t-4|t-3}$, and is a function of $\hat{x}_{t-2|t-2}$, $\hat{x}_{t-3|t-2}$, y_{t-2}, y_{t-3} and y_{t-4}.

output	symbol estimates		
y_{t-4}	$\hat{x}_{t-4\|t-2}$	0	0
y_{t-3}	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$	0
y_{t-2}	$\hat{x}_{t-2\|t-2}$	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$

Table 5.7. When y_{t-1} becomes available, estimate x_{t-1} with $\hat{x}_{t-1|t-1}$, which is a function of $\hat{x}_{t-2|t-2}$, $\hat{x}_{t-3|t-2}$ and y_{t-1}.

output	symbol estimates		
y_{t-3}	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$	0
y_{t-2}	$\hat{x}_{t-2\|t-2}$	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t-1}$	$\hat{x}_{t-2\|t-2}$	$\hat{x}_{t-3\|t-2}$

Table 5.8. Revise estimate of x_{t-2}, using FLSDFE with smoothing lag $n = 1$. $\hat{x}_{t-2|t-1}$ replaces $\hat{x}_{t-2|t-2}$, and is a function of $\hat{x}_{t-1|t-1}$, $\hat{x}_{t-3|t-2}$, $\hat{x}_{t-4|t-2}$, y_{t-1} and y_{t-2}.

output	symbol estimates		
y_{t-3}	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$	0
y_{t-2}	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-2}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t-1}$	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-2}$

Table 5.9. Revise estimate of x_{t-3}, using FLSDFE with smoothing lag $n = 2$. $\hat{x}_{t-3|t-1}$ replaces $\hat{x}_{t-3|t-2}$, and is a function of $\hat{x}_{t-1|t-1}$, $\hat{x}_{t-2|t-1}$, $\hat{x}_{t-4|t-2}$, y_{t-1}, y_{t-2} and y_{t-3}.

output	symbol estimates		
y_{t-3}	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$	0
y_{t-2}	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t-1}$	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$

Table 5.10. When y_t becomes available, estimate x_t with $\hat{x}_{t|t}$, which is a function of $\hat{x}_{t-1|t-1}$, $\hat{x}_{t-2|t-1}$ and y_t.

output	symbol estimates		
y_{t-2}	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t-1}$	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}_{t\|t}$	$\hat{x}_{t-1\|t-1}$	$\hat{x}_{t-2\|t-1}$

Table 5.11. Revise estimate of x_{t-1}, using FLSDFE with smoothing lag $n = 1$. $\hat{x}_{t-1|t}$ replaces $\hat{x}_{t-1|t-1}$, and is a function of $\hat{x}_{t|t}$, $\hat{x}_{t-2|t-1}$, $\hat{x}_{t-3|t-1}$, y_t and y_{t-1}.

output	symbol estimates		
y_{t-2}	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t-1}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}_{t\|t}$	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t-1}$

Table 5.12. Revise estimate of x_{t-2}, using FLSDFE with smoothing lag $n = 2$. $\hat{x}_{t-2|t}$ replaces $\hat{x}_{t-2|t-1}$, and is a function of $\hat{x}_{t|t}$, $\hat{x}_{t-1|t}$, $\hat{x}_{t-3|t-1}$, $\hat{x}_{t-4|t-2}$, y_t, y_{t-1} and y_{t-2}.

output	symbol estimates		
y_{t-2}	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}_{t\|t}$	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t}$

The steps illustrated in tables 5.10–5.12 are then repeated for each subsequent time step t.

Note that the variant FLSDFE algorithm illustrated in tables 5.1–5.12 above is one possibility, but there are others. For instance, a different variant may be obtained by iterating through the steps in tables 5.10–5.12. The first two reestimation passes are shown in tables 5.13–5.18.

Table 5.13. Revise estimate of x_t using updated estimates $\hat{x}_{t-1|t}$ and $\hat{x}_{t-2|t}$ of x_{t-1} and x_{t-2}, respectively. $\hat{x}^{(1)}_{t|t}$ is a function of $\hat{x}_{t-1|t}$, $\hat{x}_{t-2|t}$ and y_t.

output	symbol estimates		
y_{t-2}	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(1)}_{t\|t}$	$\hat{x}_{t-1\|t}$	$\hat{x}_{t-2\|t}$

Table 5.14. Revise estimate of x_{t-1} using updated estimates $\hat{x}^{(1)}_{t|t}$ and $\hat{x}_{t-2|t}$ of x_t and x_{t-2}, respectively. $\hat{x}^{(1)}_{t-1|t}$ is a function of $\hat{x}^{(1)}_{t|t}$, $\hat{x}_{t-2|t}$, $\hat{x}_{t-3|t-1}$, y_t and y_{t-1}.

output	symbol estimates		
y_{t-2}	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(1)}_{t\|t}$	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}_{t-2\|t}$

Table 5.15. Revise estimate of x_{t-2} using updated estimates $\hat{x}^{(1)}_{t|t}$ and $\hat{x}^{(1)}_{t-1|t}$ of x_t and x_{t-1}, respectively. $\hat{x}^{(1)}_{t-2|t}$ is a function of $\hat{x}^{(1)}_{t|t}$, $\hat{x}^{(1)}_{t-1|t}$, $\hat{x}_{t-3|t-1}$, $\hat{x}_{t-4|t-2}$, y_t, y_{t-1} and y_{t-2}.

output	symbol estimates		
y_{t-2}	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(1)}_{t\|t}$	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$

Table 5.16. Revise estimate of x_t using updated estimates $\hat{x}^{(1)}_{t-1|t}$ and $\hat{x}^{(1)}_{t-2|t}$ of x_{t-1} and x_{t-2}, respectively. $\hat{x}^{(2)}_{t|t}$ is a function of $\hat{x}^{(1)}_{t-1|t}$, $\hat{x}^{(1)}_{t-2|t}$ and y_t.

output	symbol estimates		
y_{t-2}	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(2)}_{t\|t}$	$\hat{x}^{(1)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$

Table 5.17. Revise estimate of x_{t-1} using updated estimates $\hat{x}^{(2)}_{t|t}$ and $\hat{x}^{(1)}_{t-2|t}$ of x_t and x_{t-2}, respectively. $\hat{x}^{(2)}_{t-1|t}$ is a function of $\hat{x}^{(2)}_{t|t}$, $\hat{x}^{(1)}_{t-2|t}$, $\hat{x}_{t-3|t-1}$, y_t and y_{t-1}.

output	symbol estimates		
y_{t-2}	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}^{(2)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(2)}_{t\|t}$	$\hat{x}^{(2)}_{t-1\|t}$	$\hat{x}^{(1)}_{t-2\|t}$

Table 5.18. Revise estimate of x_{t-2} using updated estimates $\hat{x}^{(2)}_{t|t}$ and $\hat{x}^{(2)}_{t-1|t}$ of x_t and x_{t-1}, respectively. $\hat{x}^{(2)}_{t-2|t}$ is a function of $\hat{x}^{(2)}_{t|t}$, $\hat{x}^{(2)}_{t-1|t}$, $\hat{x}_{t-3|t-1}$, $\hat{x}_{t-4|t-2}$, y_t, y_{t-1} and y_{t-2}.

output	symbol estimates		
y_{t-2}	$\hat{x}^{(2)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$	$\hat{x}_{t-4\|t-2}$
y_{t-1}	$\hat{x}^{(2)}_{t-1\|t}$	$\hat{x}^{(2)}_{t-2\|t}$	$\hat{x}_{t-3\|t-1}$
y_t	$\hat{x}^{(2)}_{t\|t}$	$\hat{x}^{(2)}_{t-1\|t}$	$\hat{x}^{(2)}_{t-2\|t}$

This iteration could be continued indefinitely, raising questions about the convergence of the sequence of estimates $\{\hat{x}^{(k)}_{t|t}, \hat{x}^{(k)}_{t-1|t}, \hat{x}^{(k)}_{t-2|t}\}$ as $k \to \infty$. For some channel models, the sequence may converge after a small number of iterations; for other channel models, the sequence may perhaps exhibit chaotic behaviour. It is unknown, in general, how the FLSDFE algorithm will behave if smoothed outputs are fed back.

Note that by allowing smoothed FLSDFE outputs to be fed back, error recovery may potentially be accelerated. Consider table 5.10, which contains estimates $\hat{x}_{t|t}$, $\hat{x}_{t-1|t-1}$, $\hat{x}_{t-2|t-1}$, $\hat{x}_{t-3|t-1}$ and $\hat{x}_{t-4|t-2}$ of symbols $x_t, \ldots, x_{t-4}$, respectively. Suppose that there is no current filtering error, so that $\hat{x}_{t|t} = x_t$; but that there were previous smoothing errors in that $\hat{x}_{t-1|t-1} \neq x_{t-1}$ and $\hat{x}_{t-2|t-1} \neq x_{t-2}$. (For $N = 2$ we don't care about whether $\hat{x}_{t-3|t-1}$ and $\hat{x}_{t-4|t-2}$ are accurate estimates of x_{t-3} and x_{t-4}.) We could describe this error state as the shorthand *OEE*, going backwards in time from left to right, where O means no error and E denotes an error.

Suppose that at the next step, illustrated in table 5.11, the new estimate $\hat{x}_{t-1|t}$ matches the symbol x_{t-1}, thus correcting for the error made with the previous flawed estimate $\hat{x}_{t-1|t-1}$ of table 5.10. Then we would be in the error state *OOE*, having advanced a stage in error recovery while still remaining at the current time t.

Finally, suppose that at the subsequent step, illustrated in table 5.12, the new estimate $\hat{x}_{t-2|t}$ now matches x_{t-2}, thereby correcting for the previous flawed estimate $\hat{x}_{t-2|t-1}$. Then we would be in the error state *OOO*, which is the error–free state with a channel memory $N = 2$. We would therefore have jumped from an error–just–occurred state *OEE* to the error–free state in a single time step with this variant FLSDFE, which allows smoothing errors to be fed back.

As a final note, the FLSDFE may potentially be used with a wider range of channel models than those discussed so far, such as the rational Volterra model (2.60).