

# Cross-Layer Multimedia Transmission Over Cognitive Radio Networks

*by*  
*Saqib Ali*

A thesis submitted to the Faculty of Graduate  
Studies and Research in partial fulfillment to  
the requirements for the degree of

Masters of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering  
Department of Systems and Computer Engineering  
Carleton University  
Ottawa, Ontario, Canada

December 2007

© 2007 Saqib Ali



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*

*ISBN: 978-0-494-36819-0*

*Our file* *Notre référence*

*ISBN: 978-0-494-36819-0*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# Abstract

In cognitive radio (CR) networks, the perceived reduction of application layer quality of service (QoS), such as multimedia distortion, by secondary users may impede the success of CR technologies. Most previous work in CR networks ignores application layer QoS. In this thesis we take an integrated cross-layer design approach to jointly optimize multimedia intra-refresh rate, an application layer parameter, together with access strategy, and spectrum sensing for multimedia transmission in a CR system with time varying wireless channels. Primary network usage and channel gain are modeled as a finite state Markov process. With sensor sensing and channel state information (CSI) errors, the system state cannot be directly observed. We formulate the whole system as a partially observable Markov decision process (POMDP). A low complexity dynamic programming framework is presented to obtain the optimal policy. Numerical results show the effectiveness of the proposed scheme.

# Acknowledgments

I would like to thank the invaluable supervision and support of my supervisor Professor Fei Richard Yu during the development of this work.

To my family, their constant support and love through the course of my studies took me where I am. This would not have been possible without the constant encouragement from Seema, Shahina, Karim, and Rahim. A special thanks to my parents, who lit the candle and shielded it from the wind.

My wholehearted thanks go to to my friends for their support during the good and bad times.

Specially, to my beloved Andaleep whose care and patience carried me through this endeavor.

The undersigned recommends to  
The Faculty of Graduate Studies and Research  
acceptance of the thesis  
**Cross-Layer Multimedia Transmission  
Over Cognitive Radio Networks**  
submitted by  
Saqib Ali, B.Sc.  
in partial fulfillment  
of the requirements for the degree of  
Masters of Applied Science in Electrical Engineering

Fei Richard Yu

---

Thesis Supervisor

---

Chairperson, Department of Systems and Computer Engineering

Carleton University

December 5, 2007

---

# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Abbreviations</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview and Motivation . . . . .	1
1.2 Thesis Contributions . . . . .	3
1.3 Thesis Structure . . . . .	4
<b>2 Research Background</b>	<b>5</b>
2.1 Cognitive Radio . . . . .	5
2.1.1 Dynamic Spectrum Access . . . . .	6
2.2 Rate-Distortion (R-D) Model for Multimedia Applications . . . . .	16
2.2.1 Overview of the Multimedia Communication System . . . . .	17
2.2.2 Adaptive Mode Selection . . . . .	24
2.3 Cross-layer Wireless Multimedia Transmission . . . . .	28
2.4 Related Work . . . . .	30
2.4.1 Some work in Dynamic Spectrum Access . . . . .	30
2.4.2 Some work in cross-layer multimedia wireless transmission . . . . .	31
2.5 Summary . . . . .	33

---

<b>3</b>	<b>MDP and POMDP</b>	<b>34</b>
3.1	Markov Decision Process . . . . .	34
3.1.1	MDP Model . . . . .	34
3.1.2	Policies . . . . .	36
3.1.3	Horizon . . . . .	36
3.1.4	Policies in MDPs . . . . .	37
3.2	Partially Observable Markov Decision Process . . . . .	38
3.2.1	POMDP Model . . . . .	38
3.2.2	Information State . . . . .	39
3.2.3	Policies for POMDPs . . . . .	40
3.3	Algorithms Overview . . . . .	43
3.4	Summary . . . . .	45
<b>4</b>	<b>Proposed Multimedia Transmission over Cognitive Radio Networks</b>	<b>46</b>
4.1	Overview . . . . .	46
4.2	System Model . . . . .	46
4.3	Rate-Distortion (R-D) Model for Multimedia Applications . . . . .	49
4.4	Solving the Multimedia Transmission over Cognitive Radio Networks Problem . . . . .	51
4.4.1	State Space, Transition Probabilities and Observation Space . . . . .	52
4.4.2	Information State . . . . .	54
4.4.3	Action Space . . . . .	54
4.4.4	Cost and Policy . . . . .	56
4.4.5	Objective and Constraint . . . . .	57
4.4.6	Value Function . . . . .	57
4.4.7	Intra Refreshing Strategy . . . . .	58

---

4.5	Summary . . . . .	58
<b>5</b>	<b>Simulation Results</b>	<b>60</b>
5.1	Selection of Rate-Distortion Parameters . . . . .	60
5.2	One Channel Scenario . . . . .	61
5.2.1	Performance Improvement . . . . .	61
5.2.2	Effects of the Parameters in the State Transition Matrix . . . . .	62
5.2.3	Effects of the Parameters in the Observation Matrix . . . . .	63
5.3	Two Channel Scenario . . . . .	65
5.3.1	The Affect of Transition Matrix on Channel Selection Policy . . . . .	68
5.3.2	The Affect of Observation Matrix on Channel Selection Policy . . . . .	71
5.4	Summary . . . . .	73
<b>6</b>	<b>Conclusions and Future Work</b>	<b>76</b>
6.1	Conclusions . . . . .	76
6.2	Future Research . . . . .	78
	<b>References</b>	<b>79</b>
	<b>Appendix A: MATLAB Programs</b>	<b>91</b>
A.1	Single Channel Scenario . . . . .	91
A.1.1	Transition and Observation Matrix Generation . . . . .	91
A.1.2	State Vector Generation . . . . .	94
A.2	Two Channel Scenario . . . . .	97
A.2.1	Transition Matrix Generation . . . . .	97
A.2.2	POMDP Matrix Generation . . . . .	101
A.2.3	State Vector Generation . . . . .	105
A.2.4	System Simulation . . . . .	106

---

A.2.5	Get Acknowledgement . . . . .	107
A.2.6	Update Information State . . . . .	108
<b>Appendix B: POMDP Files</b>		<b>111</b>
B.1	Example POMDP File for Two Channel Scenario . . . . .	111

---

## List of Figures

2.1	Spectrum Opportunity Detection. . . . .	11
2.2	Receiver Operating Curve [1]. . . . .	13
2.3	Video transmission scheme. . . . .	17
2.4	DCT-based Video Encoder Block Diagram. . . . .	19
2.5	Spatial and temporal error propagation. . . . .	20
3.1	Markov Decision Process. . . . .	35
3.2	POMDP Dynamic Model. . . . .	39
3.3	Value function drawn over information space. . . . .	41
3.4	PWLC Value Function. . . . .	42
4.1	Finite State Markov Channel with primary network usage. . . . .	47
4.2	The slot structure. . . . .	48
4.3	The block diagram of multimedia transmission over cognitive radio networks. . . . .	48
4.4	Sequence of operations in a slot. . . . .	52
4.5	Information state in POMDP. . . . .	55
5.1	Average distortion vs. the number of states in different schemes. . . . .	62

---

5.2	Average distortion vs. state transition matrix parameters (Case I: The probability of staying in the same state). . . . .	64
5.3	Average distortion vs. state transition matrix parameters (Case II: The probability of transitioning to the busy state). . . . .	65
5.4	Average distortion vs. observation matrix parameters (Case I: Receiver channel estimation standard deviation, $\sigma$ ). . . . .	66
5.5	Average distortion vs. observation matrix parameters (Case II: Sensor operating point, $\epsilon$ ). . . . .	67
5.6	Two channel spectrum occupancy and time varying channel gain sample scenario. . . . .	67
5.7	Two Channel Scenario: Spectrum Utilization vs. channel 1 state transition matrix parameters (Case I: The probability of staying in the busy state). . . . .	68
5.8	Two Channel Scenario: Spectrum Utilization vs. channel 1 state transition matrix parameters (Case II: The probability of transitioning to the busy state). . . . .	69
5.9	Two Channel Scenario: Average distortion vs. channel 1 state transition matrix parameters (Case I: The probability of staying in the busy state). . . . .	70
5.10	Two Channel Scenario: Average distortion vs. channel 1 state transition matrix parameters (Case II: The probability of transitioning to the busy state). . . . .	71
5.11	Two Channel Scenario: Spectrum Utilization vs. channel 1 observation matrix parameters (Case I: Sensor operating point, $\epsilon$ ). . . . .	72

---

5.12 Two Channel Scenario: Spectrum Utilization vs. channel 1 observation matrix parameters (Case II: Receiver channel estimation standard deviation, $\sigma$ ). . . . .	73
5.13 Two Channel Scenario: Average distortion vs. channel 1 observation matrix parameters (Case I: Sensor operating point, $\epsilon$ ) . . . . .	74
5.14 Two Channel Scenario: Average distortion vs. channel 1 observation matrix parameters (Case II: Receiver channel estimation standard deviation, $\sigma$ ). . . . .	75

## List of Abbreviations

BSC	Binary Symmetric Channel
CLD	Cross-Layer Design
CR	Cognitive Radio
CSI	Channel State Information
CTS	Clear-To-Send
DARPA	Defense Advanced Research Project Agency
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
DSA	Dynamic Spectrum Access
FCC	Federal Communications Commission
FEC	Forward Error Correction
HVS	Human Visual System
ISM	Industrial, Scientific, and Medical Radio Band
MAC	Medium Access Control
MB	Macroblock
MDP	Markov Decision Process
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnection

---

PCM	Pulse Code Modulation
POCS	Projection Onto Convex Sets
POMDP	Partially Observable Markov Decision Process
PSNR	Peak Signal-to-Reconstructed Image
PWLC	Piecewise Linear and Convex
QoS	Quality of Service
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RS	Reed-Soloman
RTS	Request-To-Send
SDR	Software Defined Radio
SU	Spectrum Utilization
VLC	Variable Length Coding
WiFi	IEEE 802.11 wireless local area network

# Chapter 1

## Introduction

### 1.1 Overview and Motivation

Recent widespread acceptance of wireless applications has triggered a huge demand for radio spectrum. For many years radio spectrum has been assigned to licensed (primary) users. Most of the time, some frequency bands in the radio spectrum remain largely unoccupied by primary users. Spectrum usage measurements by the Federal Communications Commission (FCC) Spectrum Policy Task Force [2] shows that at any given time and location, most of the spectrum is actually idle. That is, the spectrum shortage results from the spectrum management policy not the actual physical scarcity of usable spectrum. Cognitive radio (CR), which has been introduced in [3], is considered as an enabling technology that allows unlicensed (secondary) users to operate in licensed spectrum bands. This can help overcome the lack of available spectrum in wireless communications. A CR device is capable of sensing its surrounding environment and adapting its internal states by making corresponding changes in certain operating parameters in real-time for reliable and spectrally efficient communication [4]. The FCC in the United States and the NeXt Generation program

of the Defense Advanced Research Project Agency (DARPA) aim to redistribute allocated spectrum dynamically [5, 6].

Although much work has been done in CR networks, most previous work considers maximizing the throughput of secondary users as one of the most important design criteria. As a consequence, other QoS measures for secondary users, such as distortion for multimedia applications, are mostly ignored in the literature. However, recent work in cross-layer design shows that maximizing throughput does not necessarily benefit QoS at application layer for some multimedia applications, such as video [7, 8]. From a user's point of view, QoS at application layer is more important than at other layers. Moreover, CR-based services for secondary users would have a strictly lower QoS than radio services that enjoy guaranteed spectrum access [9]. Therefore, if the application layer QoS is not carefully considered in CR networks, the perceived reduction in QoS associated with CR may impede the success of CR technologies.

Multimedia applications such as video telephony, conferencing, and video surveillance are being targeted for wireless networks, including CR networks. Lossy video compression standards, such as MPEG4 and H.264, exploit the spatial and temporal redundancy in video streams to reduce the required bandwidth to transmit video. Compressed video comprises of intra- and inter-coded frames. The *intra-refresh rate* is an important application layer parameter to improve error resilience [10]. Adaptively adjusting the intra-refresh rate for online video encoding applications, can combat the time varying wireless channels available to secondary users in CR networks.

Cross-layer wireless multimedia transmission, where parameter optimization is considered jointly across Open Systems Interconnection (OSI) layers, has been well studied in the literature. Recent work shows promising improvement to video QoS by considering resource management, adaptation, and protection strategies available at the physical, medium access control (MAC), and network/transport layers in conjunc-

tion with multimedia compression and streaming algorithms. Although cross-layer design techniques for wireless multimedia transmission exist in the literature, little work investigates channel adaptive multimedia transmission over a cognitive radio network.

Therefore the motivation of this thesis is to investigate cross-layer multimedia transmission over CR networks.

## 1.2 Thesis Contributions

*We propose a cross-layer scheme to maximize application layer QoS for multimedia transmission over cognitive radio networks.* Based on the sensed channel condition, secondary users can adapt the intra-refresh rate at application layer, in addition to parameters at other layers. Some distinct features of the proposed scheme are as follows.

- For secondary users in CR networks, channel selection for spectrum sensing, access decision, and intra refreshing rate are determined concurrently to maximize the QoS at application layer (i.e., minimize distortion for video applications).
- Unlike previous work [11–13], that use channel availability to make the secondary user channel access decision, in this thesis, channel state information (CSI) (channel gain) is also used by secondary users to help make the optimal decision to maximize the application layer QoS.
- Primary network usage and channel gain are modeled as a finite state Markov process. With channel sensing and CSI errors, the state cannot be directly observed. We formulate the whole system as a partially observable Markov decision process (POMDP) [14]. The optimal policy can be obtained by solving the POMDP with dynamic programming-based hidden Markov model algorithms.

- Using numerical examples, we show that application layer parameters have significant impact on the QoS perceived by secondary users in CR networks. We also show that application layer QoS can be improved significantly if the intra refreshing rate is adapted together with parameters at low layers, such as spectrum sensing. This study reveals a number of interesting observations and provides insights into the design and optimization of CR networks from a cross-layer perspective.

A paper based on part of this work has been submitted to the IEEE International Conference on Communications (ICC):

- S. Ali and F. R. Yu, “*Cross-Layer Multimedia Transmission Over Cognitive Radio Networks*”, submitted to ICC’08, Beijing, China.

Another paper based on part of this work has been submitted to IEEE Transactions on Wireless Communications:

- S. Ali and F. R. Yu, “*Cross-Layer Multimedia Transmission Over Cognitive Radio Networks*”, submitted to IEEE Trans. on Wireless Commun.

### 1.3 Thesis Structure

The structure of this thesis is as follows. Chapter 2 presents the background for this thesis, and summarizes the state-of-the-art in cognitive radio, rate-distortion modeling, and cross-layer wireless transmission. Chapter 3 describes the POMDP problem and solving algorithms. The policy and value function of POMDP are also described in this chapter. Our proposed cross-layer multimedia transmission scheme for cognitive radio networks is illustrated in Chapter 4. Chapter 5 shows the numerical examples of our proposed scheme. Finally, Chapter 6 summarizes this thesis’s contribution and provides further work directions.

## Chapter 2

# Research Background

### 2.1 Cognitive Radio

Cognitive Radio (CR), first introduced by Mitola in [3], is defined as an intelligent wireless communication system with environment awareness that is able to learn and adapt to statistical variations in the input stimuli. The main goals for CR is to provide time and location agnostic reliable communication and to provide an efficient utilization of the radio spectrum. As a result, CR has been presented as a solution to address the shortage of available radio spectrum. Adding spectrum intelligence into a wireless network is a challenging task and researchers have presented various approaches in the body of literature. The purpose of this section is to give an overview of CR concepts, applications, and state-of-the-art.

A more rigorous definition of CR is provided by Haykin who defines CR as [4], “*cognitive radio is an intelligent wireless communication system that is aware of its surrounding environment (i.e. outside world), and uses the methodology of understanding-by-building to learn from the environment and adapt its internal states to statistical variations in the incoming RF stimuli by making corresponding changes*

*in certain operating parameters (e.g. transmit power, carrier frequency, and modulation strategy) in real-time, with two primary objectives in mind:*

- *highly reliable communications whenever and wherever needed;*
- *efficient utilization of the radio spectrum.”*

Parameter adaptation can be based on several factors such as radio spectrum availability, channel bandwidth and gain, primary user behavior, and QoS requirements. To build highly reliable communication, a secondary user must limit interference with primary users.

Software defined radio (SDR) is an enabling technology allowing a wireless node to tune to any frequency band and receive any modulation across a large frequency spectrum [15]. CR builds on SDR by adding a cognitive function, where a device is capable of thinking, reasoning, or remembering, to drive the radio controller. Although CR can refer to every possible parameter observable to a wireless node, we will only consider the radio frequency spectrum.

### 2.1.1 Dynamic Spectrum Access

One important application of CR is dynamic spectrum access (DSA) and an excellent literature survey can be found in [1]. The authors develop a taxonomy for the different DSA strategies available in the literature and is used throughout this section. DSA strategies can be broadly categorized under three models:

- Dynamic Exclusive Use Model
- Open Sharing Model
- Hierarchical Access Model

The dynamic exclusive use model is a variation of the existing spectrum regulation policy where spectrum is licensed for exclusive use. The difference is in the flexibility in how the spectrum is licensed to users. Spectrum property rights [16] allows spectrum owners to sell or trade spectrum and to freely choose technology. Dynamic spectrum allocation exploits the spatial and temporal traffic statistics to provision exclusive spectrum use [17]. Although similar to the current model, dynamic spectrum allocation occurs at a much faster time scale than the current static spectrum allocation strategy.

While dynamic exclusive use model provides exclusive spectrum allocation, the open sharing model allows open sharing among peer users to manage a spectral region. One example of open sharing is WiFi where devices share the unlicensed industrial, scientific, and medical (ISM) radio band. Two types of spectrum sharing strategies have been investigated (1) centralized [18,19] and (2) distributed [20-22].

With the hierarchical access model, licensed spectrum is opened to secondary users under the condition that secondary users do not interfere with primary users beyond a certain probability of collision or interference temperature. Spectrum underlay and overlay are two strategies allowing the coexistence of primary and secondary users. Underlay refers to the approach where the transmission power of secondary users is limited to be less than the noise floor of primary users. Ultra-wideband is one example of an underlay strategy. Overlay does not limit the transmit power of secondary users but imposes restrictions on when and where secondary users can transmit. Secondary users can exploit spatial and temporal spectrum white space in a nonintrusive manner. The overlay method is the DSA strategy envisioned by Mitola [3] and is the strategy used in our proposed scheme. The hierarchical access model is likely the most compatible with the current spectrum management policies and provides better spectrum efficiency in the licensed bands.

Our proposed scheme falls in the spectrum overlay hierarchical access model classification. Thus the remainder of this section will focus on concepts and review the state-of-the-art for this class of DSA.

### Spectrum Sensing and Opportunity Identification

Spectrum sensing and opportunity identification are essential for spectrum overlay systems. Given a system with a set of channels, where a channel can refer to a frequency band, spreading codes in a CDMA network, or a set of tones in an orthogonal frequency division multiplexing (OFDM) system, a secondary CR device will need to sense the channel and identify transmission opportunities. With spectrum sensing error, spatial spectral differences at the transmitter and receiver, and limited interference tolerance with primary users, spectrum sensing and opportunity identification are challenging tasks and have received much attention in the body of literature.

A channel opportunity occurs when two secondary users can communicate successfully while limiting interference with primary users below an interference threshold defined by the regulatory policy. For a pair of secondary devices, the spectral environment at the transmitter may be different than at the receiver. Thus a spectrum opportunity depends not only on the transmitter location but on the receiver location as well [23]. Note that spectrum opportunity is determined by the activities of primary users rather than secondary users [23]. In a system with multiple secondary users, collisions among secondary users do not disqualify spectrum opportunities.

The level of interference with primary users is defined using two parameters: (1) interference power level,  $\eta$  and (2) maximum probability  $\zeta$  that the interference level at an active primary receiver may exceed  $\eta$  [1]. The first parameter,  $\eta$ , is essentially the noise floor for primary users where only interference power above  $\eta$  results in a collision. The second parameter,  $\zeta$ , is the maximum allowable collision probability.

Practical spectrum sensors will have sensing error thus collisions with primary users are inevitable. The work in [23] discusses a cautionary aspect regarding the definition of collision probability. The probability of collision could be defined as:

- joint probability that both primary and secondary users access the channel
- conditional probability that a secondary user transmits given that the channel is occupied by primary users

The drawback with the joint probability is that it includes the traffic statistic of primary users into the interference constraint. That is, primary users traffic statistics influence the level of interference protection.

In addition to whether a joint or conditional probability is used in the probability of collision definition, how the constraint is imposed may also influence the level of interference protection. For example, the following methods could be used to impose the probability of collision constraint [23]:

- the constraint is imposed in each slot over each channel
- the constraint is averaged over a set of channels and a long period of time

Applying the constraint in each slot over each channel offers a specific level of protection to primary users no matter when and over which channel they transmit. On the other hand, using an average constraint offers unpredictable protection when primary users have bursty arrivals of short messages and varies for different primary users depending on their traffic statistics.

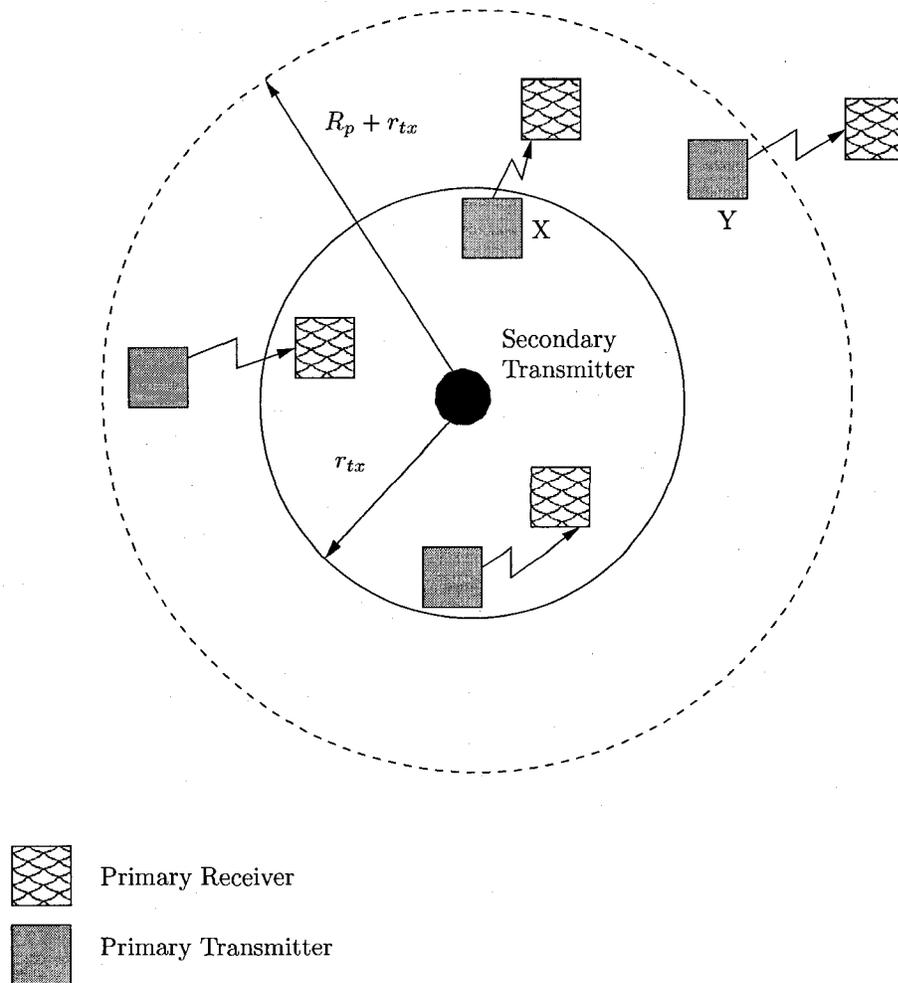
Given the benefits presented here, our proposed scheme uses a probability of collision constraint based on the conditional probability that a secondary user transmits given that the channel is occupied by primary users and impose this limit in each slot over each channel.

Due to spatial spectrum differences, opportunity detection must be performed jointly at the transmitter and receiver [1]. One approach presented in [11], a secondary transmitter first detects the receiving activities of primary users in its surroundings. If the channel is available, that is there are no primary receivers nearby, the transmitter-receiver pair exchange a request-to-send (RTS) and clear-to-send (CTS) handshake analogous to WiFi communication. As described in [11], this method facilitates opportunity detection, addresses contention among secondary users, and mitigates the hidden/exposed terminal problem.

Without cooperation from primary users, detecting primary receivers is a difficult task. Secondary transmitters can detect primary receivers by exploiting the local oscillator leakage power emitted by the RF front end of primary receivers [24]. This approach is limited by short detection range and long detection time [1]. An alternative approach is to detect primary transmitters as opposed to primary receivers. A secondary transmitter can detect an available channel if there are no primary transmitters within a distance of  $R_p + r_{tx}$  where  $R_p$  denotes the transmission range of primary users and  $r_{tx}$  denotes the transmission range of the secondary transmitter [1]. A detection range of  $R_p + r_{tx}$  is a conservative measure and can be adjusted to tradeoff secondary user throughput for interference with primary users. Fig. 2.1 shows the approach where the problem of detecting primary receivers is transformed to that of detecting primary transmitters. As can be seen, the secondary transmitter may overlook opportunities when primary users X and Y are active despite the primary receivers being outside the secondary users transmission range.

Three classical techniques can be used to detect primary transmitters [25]:

- matched filter
- energy detector (radiometer)



**Fig. 2.1** Spectrum Opportunity Detection.

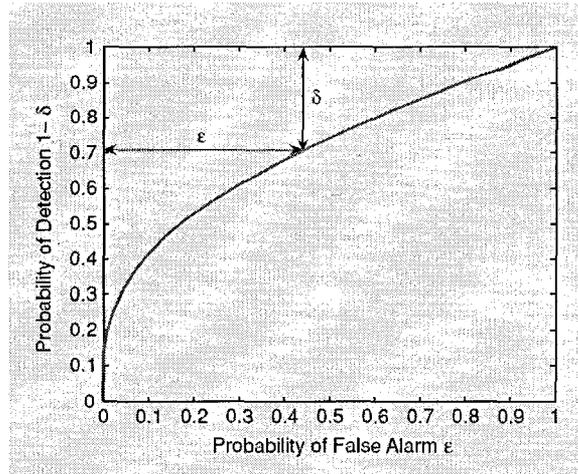
- cyclostationary feature detector.

A matched filter performs coherent detection but requires synchronization and a priori knowledge of primary users' signaling. A noncoherent energy detector requires basic information of primary users' signal characteristics but suffers from long detection time. A cyclostationary feature detector can improve energy detector performance by exploiting inherent periodicity in the primary users' signal [26]. The limitations of using these techniques to detect primary transmitters is discussed in [27].

Cooperative spectrum sensing, where secondary users exchange sensing information over a control channel, has been presented in [12,13,25,28,29]. These cooperative sensing methods suffer from an overhead associated with sensing information exchange, the need for a control channel, and slow response to spectrum changes [1].

A spectrum opportunity detector will perform a binary hypotheses test, where the hypothesis  $\mathcal{H}_0$  indicates the absence of primary users (an opportunity), and the hypothesis  $\mathcal{H}_1$  indicates the presence of a primary user. A Neyman-Pearson or Bayesian criterion could be used however it is shown in [30] that an energy detector, which is used in our proposed scheme, is optimal under the Neyman-Pearson criterion. Practical sensors will have sensing error. A false alarm occurs when the detector hypothesis is  $\mathcal{H}_1$  but in actuality there are no primary users. A miss detection occurs when the detector hypothesis is  $\mathcal{H}_0$  when in actuality a primary user signal is present. Let  $\epsilon$  and  $\delta$  represent the probability of false alarm and miss detection respectively. The receiver operating characteristic (ROC) curve is used to represent the performance of the detector as shown in Fig. 2.2 and displays the tradeoff between the probability of false alarm,  $\epsilon$ , and the probability of miss detection,  $\delta$ . Decreasing  $\epsilon$  results in an increase in  $\delta$ . Selecting the appropriate sensor operating point should limit interference with primary users without being overly conservative to secondary users [31].

While practical sensors are bound to sensing error, due to hardware limitations practical CRs are limited in the number of channels that can be simultaneously sensed. The authors in [11,32] develop a sensing strategy that achieves the optimal tradeoff between gaining immediate access in the current time slot and gaining system state information for future use. They formulate the optimal sensing strategy in the framework of a partially observable Markov decision process (POMDP) and show that the throughput of secondary users is significantly improved by employing this sensing strategy.



**Fig. 2.2** Receiver Operating Curve [1].

The efficacy of the spectrum sensing strategy requires accurate statistical models of spectrum occupancy. Spectrum monitoring testbeds and cognitive radio prototypes are being developed by academia and industry [1].

### Opportunity Exploitation

Once spectrum opportunities have been identified, a secondary user must determine how to exploit them. Thus a secondary user requires an access strategy. The access strategy is complicated in the presence of sensing errors and must consider the operating characteristics of the spectrum sensor [1]. For example, with a high miss detection probability the access strategy should be conservative to avoid collisions with primary users. If the sensor operating point has a high probability of false alarms then a more aggressive access strategy should be employed to improve spectrum opportunity utilization. The authors in [31] develop an optimal access and sensing strategy using a POMDP framework. The authors find that the optimal joint design of spectrum detector, sensing, and access strategies requires that the detector be designed under the Neyman-Pearson criterion and operate at the transition point  $\delta^* = \zeta$  where  $\delta^*$

denotes the optimal operating point. The optimal access strategy is to trust the detector, access if the channel is sensed to be available and do not access if the channel is sensed to be busy. By employing this scheme we have an optimal sensor operating point and access strategy that respects the spectrum policy probability of collision constraint.

Opportunities may not present themselves over contiguous frequency bands. Modulation using OFDM is useful to exploit these spectral holes [33–35]. The subcarrier structure of OFDM can be reconfigured for efficient usage of spectral holes. However the subcarrier spacing and symbol interval need to match with the spectral and temporal duration of spectrum opportunities [34]. Signal truncation in the time domain results in cross-channel spectrum leakage in the frequency domain. The spectral leakage coupled with nonlinearity of the transmitter’s power amplifier needs to be controlled to ensure nonintrusive communication with primary users. The impact of nulled subcarriers on the peak-to-average-power ratio of the transmitted OFDM signal also requires careful study.

Spectrum opportunities are not limited to the spatial domain. Approaches such as [11,36,37] exploit spectrum opportunities during the idle periods remaining between the primary users’ packet transmissions. The amount of white space is a function of traffic characteristics and typical wireless applications are shown to have sufficient temporal opportunities [36,38].

### **Cooperative Spectrum Sharing**

Various cooperative spectrum sharing schemes among secondary users have been investigated. One application of cognitive radio is to exploit locally unused TV broadcast frequency bands. With this application, mainly spatial spectrum opportunity sharing among secondary users have been investigated [37,39,40]. Since spectrum

opportunity is relatively static at a particular location, real-time opportunity identification is not critical.

One approach for spatial opportunity allocation is graph coloring [39, 40]. Secondary users form the vertices in a graph. Edges between these vertices indicate two interfering users. The graph coloring problem is to color the vertices giving a different color to vertices that are connected along an edge. The goal is to obtain a color assignment that maximizes a given utility function. Suboptimal centralized and distributed approaches are presented in [39, 40].

Game theory is another approach to spectrum sharing. An excellent overview of game theoretical dynamic spectrum sharing from an analysis of network users' behaviors, efficient dynamic distributed design, and optimality analysis is presented in [41]. Since CR devices make intelligent decisions on spectrum usage, game theory provides a useful analysis tool to study the intelligent behaviors and interactions of network users (cooperative, selfish, or malicious). Game theory provides a model of strategic interactions among agents by using formalized incentive structures and derives equilibrium criteria for optimality studies. In the context of dynamic spectrum sharing, game theory models sharing among network users (primary and secondary users) as games and facilitates network users' behaviors and actions to be analyzed in a formalized game structure.

### **Frequency Hopping Synchronization**

A secondary transmitter and receiver must tune to the same channel for successful communication. Maintaining synchronized frequency hopping in non-centralized opportunistic spectrum access is an interesting problem. The problem is further complicated by spectrum sensing errors and different spectral environments at the transmitter and receiver.

The authors in [31] use the information state in a POMDP framework to maintain frequency synchronization. Given that spectrum observations could be different at the transmitter and the receiver, the information state is updated by only using common information available to both the transmitter and receiver.

A dynamic frequency hopping scheme is presented in [42] for IEEE 802.22 wireless regional area networks, which is an emerging standard based on CR technologies. In this work the authors assume spectrum sensing and data transmission occur simultaneously and propose coordinated communication over dynamic frequency hopping communities.

## 2.2 Rate-Distortion (R-D) Model for Multimedia

### Applications

Multimedia applications such as streaming video, multi-party video sessions, interactive entertainment, multimedia instant messaging, and push-to-talk communication are being targeted for mobile and wireless applications. Online encoding applications such as video telephony, conferencing, and video surveillance can adaptively change encoding parameters to adjust to varying bandwidth and error conditions of the underlying wireless network. To facilitate the study of this technique we require an accurate model to measure the performance of the adaptive transmission scheme. Although the term multimedia refers to media types such as video and audio, the focus of this thesis is on video transmission, thus multimedia and video are used interchangeably throughout this thesis. The purpose of this section is to present fundamental concepts on video compression and rate-distortion modeling relevant to the understanding of the proposed scheme.

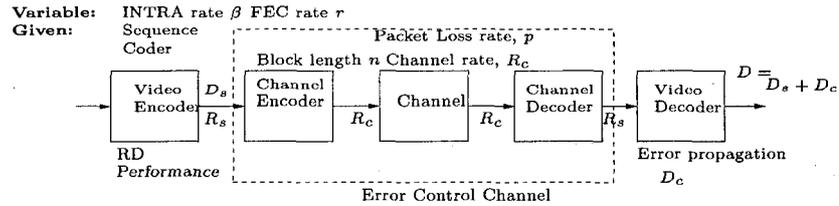


Fig. 2.3 Video transmission scheme.

### 2.2.1 Overview of the Multimedia Communication System

A typical video transmission scheme is shown in Fig. 2.3 which comprises of three components: (i) video encoder (ii) error control channel (iii) video decoder. Video is compressed at the video encoder to meet a target encoding bit rate,  $R_s$ . The tradeoff between encoding rate and distortion is modeled at the video encoder. Forward error correction is used to reduce the packet loss rate,  $p$ , over a lossy wireless channel with limited bit rate,  $R_c$ , at the error control channel. Error detection and concealment strategies are employed at the video decoder to mitigate error propagation due to channel errors. The variable intra refresh rate,  $\beta$  and FEC rate,  $r$  can be adjusted for error resilience and protection respectively. Note that the model presented in Fig. 2.3 does not provide a comprehensive view of all parameters that can be optimized for video transmission however it successfully captures the parameters relevant to our work. The total distortion  $D$  is the sum of source distortion  $D_s$  and channel distortion  $D_c$ .

Wireless channels have limited bandwidth and are error-prone. Highly efficient coding algorithms such as H.264 and MPEG4 can compress video to reduce the required bandwidth for the video stream. Rate control is used in video coding to control the video encoder output bit rate based on various conditions to improve video quality [43, 44]. For example, the main tasks of MPEG4 object-based video coding are (1) to determine how many bits are assigned to each video object in the scene and

(2) to adjust the quantization parameter to accurately achieve the target coding bit rate [45]. In previous work, rate and distortion are treated as two separate parameters based on the quantization parameter, percentage of zeros among DCT coefficients, or intra-refresh rate [10, 45, 46].

## Video Encoder

The video encoder uses lossy video compression techniques to compress the video to meet a target bit rate. Motion-compensated DCT-based video coding forms the basis of the recent and highly efficient video coding standards, H.264 and MPEG-4 .

Frames to be compressed are divided into macroblocks (MB). A MB comprises of a block of pixels (luma and chroma samples). The video is then compressed into I-, P-, and B-frames. I-frames, or intra-coded frames, compress raw frames by removing the spacial redundancy indistinguishable by the eye. P- and B-frames, also known as inter-frames, significantly enhance compression by removing the temporal redundancy in the video stream.

The standard video coding process consists of motion compensation, discrete cosine transform (DCT), quantization, and coding as shown in Fig. 2.4. Motion estimation and compensation is used to remove the temporal redundancies between frames. In a video sequence, the image in successive frames may be very similar. Motion compensation uses prediction, where a prediction error is coded together with MB motion vectors, to transmit the residual image, significantly improving the compression of the video sequence. Prediction frames can be forward-predicted, P-frames, or bidirectionally predicted, B-frames, and rely on reference frames to reconstruct the image. In contrast, I-frames do not rely on reference frames however require more bits to code.

DCT is a technique used in MPEG-4 and H.264 for data compression. Analogous

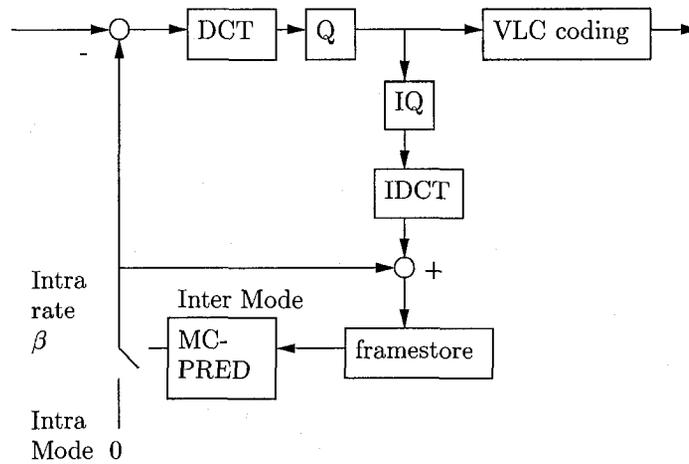


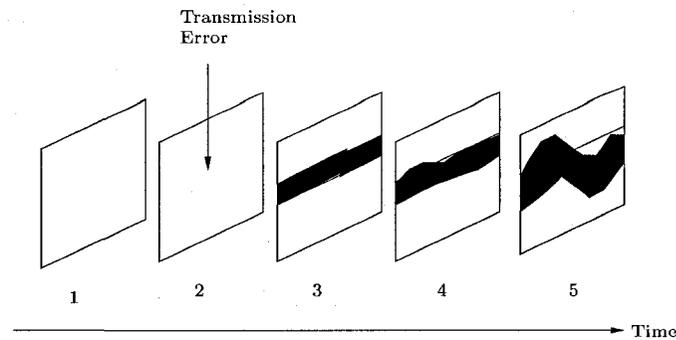
Fig. 2.4 DCT-based Video Encoder Block Diagram.

to a Fourier Transform, DCT converts pixels into a set of frequencies which are then quantized. The quantization stepsize,  $q$ , for the DCT-coefficients can be adjusted to achieve a target bit rate [47]. This forms a fundamental tradeoff between rate and distortion. In [43, 44] the rate  $R$  and distortion  $D$  are treated as functions of the quantization parameter  $q$ . Simple and more accurate R-D analysis is to treat  $R$  and  $D$  as functions of  $\rho$  where  $\rho$  denotes the percentage of zeros among the quantized DCT coefficients [45].

Variable length coding (VLC) is a reversible procedure for entropy coding that assigns shorter bit strings to symbols expected to be more frequent and longer bit strings to symbols expected to be less frequent [47].

### Error Resilience and Protection

Wireless channels have limited bandwidth and are error prone. The video encoder can compress video to meet the available bit rate of the channel, however this compressed video is vulnerable to channel errors [46, 48]. Error control and intra-update of MBs are two effective approaches for error resilience and protection [46, 48–50]. Forward error



**Fig. 2.5** Spatial and temporal error propagation.

correction (FEC) [10, 46], packet retransmission [51], or a hybrid of the two [52–54] are common error control techniques that add controlled redundancy information to mitigate channel errors. Intra-update of MBs, also known as intra refreshing, is another way to mitigate error propagation. Intra frames do not depend on other frames that could have been corrupted by channel errors. On the other hand, inter-coded MBs rely on information in other frames. Thus even if an inter-coded MB is received and decoded successfully, error from previous frames may still propagate to the current frame along the motion-compensation path [48, 49, 55]. An example of spatial and temporal domain error propagation is shown in Fig 2.5 showing how a single bit error can propagate during the decompression process.

Increased error resilience comes at the cost of higher bandwidth consumption. The increased vulnerability of compressed video [46, 48] makes error resilience at the video encoder essential. Intra update, also called intra-refreshing, of MBs is one approach for video error resilience and protection [10, 46, 48–50]. An intra coded MB does not need information from previous frames which may have already been corrupted by channel errors. This makes intra coded MBs an effective way to mitigate error propagation.

Channel coding using FEC schemes such as Reed-Soloman (RS) codes [56] is a

commonly used strategy to mitigate channel errors. Assigning more bits to channel coding can correct more bit errors. However since the channel has a limited bit rate, increasing channel coding will result in less available bits for the source encoder resulting in an increased source coding distortion [10]. Thus there is a tradeoff between the bit allocation for source coding and channel coding and several joint source-channel coding strategies have been presented in the literature [10, 46].

The pattern of channel errors can also impact the total distortion. The work in [46, 57] analyzes the affect burst length has on video distortion and shows that larger burst lengths increases distortion compared to random bit errors. Interleaving is a common technique used to spread bursts of bit errors. Some R-D models such as [46] take into account the burst length and show that interleaving can be used to reduce the effective burst length resulting in reduced distortion. Other work such as [10] assumes sufficient interleaving is employed such that the assumption of a binary symmetric channel (BSC) holds.

### Video Decoder

The video decoder will attempt to reconstruct the original video sequence. Post processing error concealment refers to operations at the decoder to recover the damaged areas based on characteristics of image and video signals. An excellent overview of error control and concealment for video communications is available in [58] and is used in the preparation of this section.

Random errors caused by the wireless channel can cause bit inversion, bit insertion, and bit deletion. With variable length coding (VLC), random bit errors can desynchronize the coded information resulting in the loss of the remaining bits until the next synchronization word appears.

Error detection can be done using sequence numbers at the packet level, using FEC

channel coding, or using the video decoder [58]. Video decoder error detection can use differences of pixel values between two neighboring lines in pulse code modulation (PCM) and differential (D)PCM coding [59,60], examining the difference between the boundary pixels in a block and its four neighbor blocks [61], invalid VLC code words, or syntax errors in the embedded bit stream [58].

As discussed before there are typically two coding modes at the MB level, intra-coding and inter-coding. In intra-mode, each block is transformed using block DCT and the DCT coefficients are quantized and coded whereas in inter-mode a motion vector and the DCT coefficients of the prediction error block are quantized and coded. If coded MBs are packetized in an interleaved manner, then damaged MBs due to channel errors are usually surrounded by correctly received MBs [58]. Postprocessing error concealment strategies at the video decoder make use of the correlation between a damaged MB and its adjacent MBs in the same frame and/or the previous frame. One simple method is to exploit the temporal correlation in video signals by replacing a damaged MB with the spatially corresponding MB in the previous frame [58]. Due to its simplicity, our proposed scheme assumes this error concealment strategy. Improved performance can be achieved by replacing damaged MBs with the block specified by the motion vector provided that motion information is available [62]. Other error concealment schemes include

- Maximally smooth recovery - missing DCT coefficients are estimated by minimizing a measure of spatial and temporal variation between adjacent pixels between this block and the spatially and temporally neighboring blocks [58,63–65].
- Projection onto Convex Sets (POCS), where a recovered block is required to have a limited bandwidth either isotropically or along a particular direction [66]
- Interpolation in the spatial and frequency domain where lost coefficients in a

damaged block are interpolated with neighboring blocks [67–69].

### Quality Assessment Metric

Distortion is a metric commonly used to measure QoS in multimedia applications. Assigning a numerical value to the quality of an image or video facilitates easy comparison of different schemes. Care should be taken when using the distortion as different methods in computing distortion appear in the literature and the peak signal-to-reconstructed image ratio (PSNR) does not equate with human subjective perception.

Mean-square-error (MSE) and peak signal-to-reconstructed image (PSNR) are the most commonly used quality metrics to evaluate video system performance. They can be computed mathematically based on the amount of errors in the processed video and video coding algorithms. Although mathematically easy to compute, these methods weigh all errors equally and do not necessarily correlate with human perceived degradations [70, 71]. Objective quality assessment metrics that include the human visual system (HVS) have been presented in [70, 72–75]. Due to the simplicity in computation, this thesis uses MSE/PSNR to measure video quality.

The MSE is given by

$$\text{MSE} = \frac{\sum [F(i, j) - \hat{F}(i, j)]^2}{N^2}, \quad (2.1)$$

where  $F(i, j)$  denotes the  $N \times N$  pixels of the original image and  $\hat{F}(i, j)$  denotes the  $N \times N$  pixels of the reconstructed image of  $F(i, j)$  after decoding, channel errors, and encoding. Some authors use the formula

$$\text{MSE} = \frac{\sum [F(i, j) - \hat{F}(i, j)]^2}{N}. \quad (2.2)$$

The root mean square error (RMSE) is the square root of the MSE,  $\text{RMSE} = \sqrt{\text{MSE}}$ . The PSNR in decibels (dB) is computed using

$$\text{PSNR} = 20 \log_{10} \left( \frac{255}{\text{RMSE}} \right). \quad (2.3)$$

Other definitions of PSNR use

$$\text{PSNR} = 20 \log_{10} \left( \frac{255^2}{\text{MSE}} \right). \quad (2.4)$$

MSE/PSNR are used to make relative comparisons to study different video compression and transmission schemes which can be obtained with consistent use of MSE/PSNR distortion definitions.

### 2.2.2 Adaptive Mode Selection

Adaptively changing the intra refresh rate of MBs is an effective way to improve error resilience and protection. The authors in [10] present a R-D model for DCT-based video coding incorporating the MB intra refreshing rate. This model facilitates distortion estimation for a particular intra refresh rate, bit rate, and channel packet loss ratio.

Although inter coding of MBs has better R-D performance, channel error propagation has a severe impact on video quality. The MB coding mode needs to tradeoff R-D

performance for error resilience. Let  $\beta$  denote the intra refreshing rate, the percentage of MBs coded with intra mode.  $\beta$  can be then selected to maximize the overall picture quality at the receiver end for a given transmission channel condition.

Let  $R_s$  and  $R_c$  denote the source and channel coding bit rates respectively. Let  $D$  denote the MSE overall picture distortion at the receiver end. There are two major types of distortion in an end-to-end wireless system: (1) quantization distortion introduced in source coding,  $D_s$  and (2) distortion caused by channel errors,  $D_c$ .  $D_s$  and  $D_c$  denote the “source distortion” and “channel distortion” respectively. Source distortion is the MSE between the original frame and the reconstructed frame at the encoder. Channel distortion is the MSE between the decoded video frame at the receiver and the reconstructed frame at the encoder. It is shown in [10, 46] that the source and channel distortion are additive to give the total end-to-end distortion,  $D = D_s + D_c$ .

Determining a source distortion model with adaptive intra-refreshing is a difficult task. R-D models have been presented in the literature based on the quantization parameter,  $q$ , or the percentage of zeros among quantized DCT coefficients,  $\rho$ . However, incorporating the intra refresh rate will cause changes to the input data to the quantizer and the entropy encoder.

Determining a channel distortion model must consider the complex error propagation in the video decoding loopback caused by inter coding. For real-time adaptive video transmission, a low complexity channel distortion estimate is required.

Increasing  $\beta$ , by definition, leads to more intra coded MBs resulting in a larger average coding bit rate. Thus for a given coding bit rate,  $R_s$ , the source distortion  $D_s$  also increases with  $\beta$ . Using extensive simulation, the authors in [10] find that the source distortion follows the quadratic approximation

$$D_s(R_s, \beta) = D_s(R_s, 0) + \beta(1 - \lambda + \lambda\beta)[D_s(R_s, 1) - D_s(R_s, 0)], \quad (2.5)$$

where  $\lambda$  is a constant depending on specific characteristics of the video sequence. To estimate  $D_s(R_s, \beta)$ , we need to estimate the R-D functions of the current frame at the all intra- and inter-modes. Determining  $D_s(R_s, 1)$  and  $D_s(R_s, 0)$  can be done using the R-D model without adaptive mode selection as presented in [10].

At the receiver end, when a corrupted codeword in the bit stream cannot be decoded, the entropy decoder will jump to the next packet starting with a resynchronization mark skipping all intermediate bits. An accurate channel distortion model must consider the complex error propagation in video decoding, varying characteristics of the input video data, specific channel conditions, complex data representation and coding scheme employed by the video encoder, sophisticated error resilience and concealment methods, the operation mechanism of the video decoder, and the random nature of the bit errors.

A simple yet effective error concealment scheme is assumed where in the event of an error the decoder copies the MB at the same location from the previous decoded frame. Let  $p$  denote the packet-loss ratio. Assuming each packet contains the same number of MBs (or pixels), then the loss ratio of a pixel is also  $p$  [48]. Let  $F(n, i)$  denote the original value of pixel  $i$  in the  $n$ th video frame, and  $\hat{F}(n, i)$  be the corresponding reconstruction value in the feedback loop at the encoder. Let  $\tilde{F}(n, i)$  denote the reconstruction value at the receiver end. Due to the randomness of bit errors,  $\tilde{F}(n, i)$  is actually a random variable. The expected picture distortion at the receiver end is given by

$$D(n) = E\{[F(n, i) - \tilde{F}(n, i)]^2\} \quad (2.6)$$

where  $E\{x(n, i)\}$  denotes the average (over all pixels) expected value of the random variable  $x(n, i)$ . The source and channel distortion is given by

$$D_s(n) = E\{[F(n, i) - \hat{F}(n, i)]^2\}, \quad (2.7)$$

$$D_c(n) = E\{[\hat{F}(n, i) - \tilde{F}(n, i)]^2\} \quad (2.8)$$

respectively. If  $D_s(n)$  and  $D_c(n)$  are uncorrelated then

$$D(n) = D_s(n) + D_c(n). \quad (2.9)$$

Using a statistical analysis of channel distortion, the channel distortion model is given by

$$D_c(n) = ((1 - \beta)(1 - p)b + p)D_c(n - 1) + paF_d(n, n - 1) \quad (2.10)$$

where  $p$  is the packet loss rate,  $b$  is a constant describing motion randomness of the Video Scene,  $a$  is the energy loss ratio of the encoder filter,  $n$  is the  $n$ th video frame, and  $F_d$  is the MSE between original frames  $n$  and  $n - 1$ . Note that this model assumes a random BSC or that bursty errors are converted into random errors with pre-interleaving [76].

The average channel distortion over several frames is given by

$$\overline{D}_c = \left( \frac{a}{1 - b + b\beta} \right) \left( \frac{p}{1 - p} \right) E[F_d(n, n - 1)] \quad (2.11)$$

where  $E[F_d(n, n - 1)]$  is the average value of the frame difference  $F_d(n, n - 1)$  over  $T$  slots. Note that  $\overline{D}_c$  is asymptotically related to the packet loss.

$\beta$  can be selected to minimize the overall distortion  $D(n) = D_s(n) + D_c(n)$ . Once the optimal intra refreshing rate  $\beta$  is determined,  $L = \beta M$  MBs are selected to be intra coded, where  $M$  is the total number of MBs. MBs are selected randomly since at the decoder the previous frame is a random variable due to the randomness of channel errors. All R-D computations and optimizations should be handled in a statistical sense. That is, the developed R-D model for adaptive intra refresh is based on the behavior of the whole frame/video as opposed to specific characteristics of each individual MB.

### 2.3 Cross-layer Wireless Multimedia Transmission

Cross-layer wireless multimedia transmission, where parameter optimization is considered jointly across OSI layers, shows promising performance gain for wireless multimedia transmission. The purpose of this section is to provide background information on the cross-layer design approach.

The authors in [7, 8] formalize the cross-layer problem, discuss its challenges, and present several possible solutions. Cross-layer design (CLD) is a concept that optimally allocates resources and adapts to dynamically changing environments by optimizing communication across traditional layer boundaries [8]. Traditional network design uses a stack of layers and designs them in isolation, such as the OSI stack.

This layered approach has been widely used and has contributed to the proliferation of the Internet [77]. However, this approach is no longer adequate to meet the challenges of next generation mobile systems [8]. Particularly wireless multimedia communications due to the time varying transmission characteristics of the wireless channel and the dynamic QoS requirements of the application layer. In the layered approach, resource management, adaptation, and protection strategies available at the lower layers of the stack such as the physical (PHY), medium access control (MAC) , and network/transport layers, are optimized without explicitly considering the specific characteristics of the multimedia application layer. Similarly, multimedia compression and streaming algorithms do not consider the mechanisms provided by the lower layers for error protection, scheduling, resource management, etc. However a network that adapts to physical layer dynamics to drive application layer QoS is able to maintain optimal allocation of resources.

CLD techniques can be classified based on the order in which cross-layer optimization is performed [7].

- Top-down approach. The higher-layer protocols optimize their parameters and the strategies at the next lower layer. For example, the application layer could optimize MAC parameters and strategies, and the MAC could select the optimum PHY layer modulation scheme.
- Bottom-Up approach. The lower layers try to insulate the higher layers from losses and bandwidth variations. This solution is not optimal for multimedia transmission due to delay and throughput reduction.
- Application-centric approach. The application layer optimizes the lower layer parameters one at a time in a bottom-up or top-down manner. The drawback of this approach is that the application layer operates at slower timescales and

coarser data granularity than the lower layers and is unable to instantaneously adapt the lower layer parameters for optimal performance.

- **MAC-centric approach.** The application layer passes traffic information and requirements to the MAC. In turn, the MAC decides which application layer packets/flows should be transmitted and at what QoS level. The MAC also decides the PHY layer parameters based on channel information. In this approach, the MAC layer is unable to adapt source channel coding trade-offs given the time-varying channel conditions and multimedia requirements.
- **Integrated approach.** Strategies are determined jointly. However, exhaustively trying all possible strategies and their parameters in order to choose the optimal composite strategy is complex. One possible solution to solve this complex cross-layer optimization problem is to use learning techniques.

## 2.4 Related Work

### 2.4.1 Some work in Dynamic Spectrum Access

The work in [31] is closely related to our proposed scheme as it considers spectrum sensor design, access strategy, and channel sensing strategy jointly for opportunistic spectrum access. The authors explicitly consider spectrum sensing errors and propose a solution to the frequency hopping synchronization problem. The joint design is formulated under the framework of a POMDP. One of the key contributions in this paper is that they develop a separation principle which significantly simplifies computing the solution for the joint design. They also show that channel correlation can be exploited at the physical layer to significantly improve throughput. The solution to the POMDP has exponential complexity with respect to the number of channels. When

the number of channels is large, searching for the optimal policy becomes impractical. The authors in [11] extend this work by obtaining a sufficient statistic whose dimension grows linearly instead of exponentially with the number of channels. Based on this reduced-dimension sufficient statistic, a suboptimal greedy approach is proposed to maximize per slot throughput. These approaches aim to maximize throughput and do not consider application layer QoS.

Other work in DSA has been presented as well. The benefits of cooperation in CR are illustrated in [12] and [13] for two user networks and multiple user networks, respectively. A dynamic frequency hopping scheme is presented in [42] for IEEE 802.22 wireless regional area networks, which is an emerging standard based on CR technologies. In [78], the authors present a game theoretical dynamic spectrum sharing framework for analysis of network users' behaviors, efficient dynamic distributed design, and optimality analysis. Other game theoretic DSA methods are presented in [21, 79]. The authors in [36] exploit channel availability in the time domain and demonstrate the throughput performance for a Bluetooth/WLAN system. Spectrum opportunity is also exploited in the time domain in [37] where the authors present an ad hoc secondary MAC protocol to facilitate DSA.

#### 2.4.2 Some work in cross-layer multimedia wireless transmission

Various channel adaptive distortion driven cross-layer transmission strategies have been explored [80–83]. Within a rate-distortion framework source coding, retransmission, and adaptive modulation parameters are jointly considered for video summary in [80]. The authors in [81] investigate a classification based system where the optimal cross-layer strategy for various video and channel conditions are computed offline thereby reducing the transmission-time complexity of the compression and transmission strategy. The authors in [82] take a cross-layer approach to allocate power level,

---

source coding rate, and channel coding rate delivering basic and enhanced QoS levels for distant and near receivers in a CDMA network respectively. In [83] multimedia adaptation is implemented at two different time scales, a macro time scale at the application layer and a micro time scale at the MAC layer. Although cross-layer design techniques for wireless multimedia transmission exist in the literature, little work investigates channel adaptive multimedia transmission over a cognitive radio network.

## 2.5 Summary

In this chapter we introduced cognitive radio, dynamic spectrum access, video transmission, and cross-layer design. Cognitive radio is a fairly new concept and is used to address the drawbacks in the current licensed spectrum management policy. Spectrum underlay and overlay dynamic spectrum access allows secondary users to utilize spectrum opportunities while limiting interference with primary users. The improved spectral efficiency makes spectrum overlay DSA an exciting area for research. The basics of DCT-based video coding, transmission, and decoding is presented. Intra refreshing can be used to improve error resilience by reducing error propagation along the motion compensation path. The existing layered architecture is inadequate to meet the time varying channel condition and application requirements in multimedia wireless transmission. Cross-layer QoS driven channel adaptive schemes have proven to significantly improve application layer QoS. Little work has been done to investigate cross-layer multimedia transmission over cognitive networks.

## Chapter 3

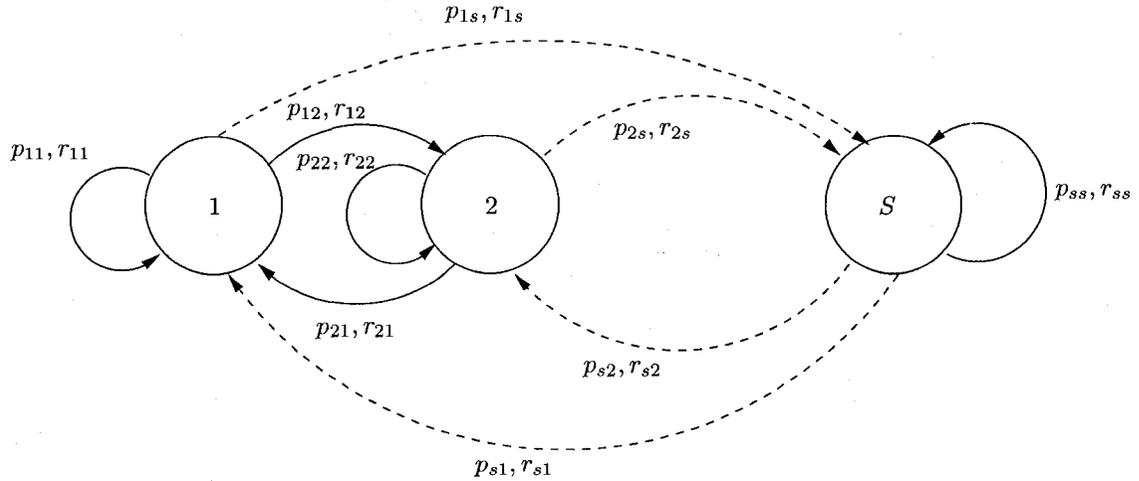
# MDP and POMDP

Decision making is the cognitive process leading to the selection of a course of action among variations. People make decisions everyday however it is often difficult to automate the decision making process. One way to automate the decision making process is to provide a model of the dynamics of the domain in which the machine will be making decisions. A reward structure can be used to motivate immediate decision which will maximize future reward. Markov decision process (MDP) and partially observable Markov decision process (POMDP) are tools that aid in this type of automated decision making. An excellent overview of MDP and POMDP is presented in [14, 84] and is used in the generation of this section.

### 3.1 Markov Decision Process

#### 3.1.1 MDP Model

The MDP model comprises of a finite set of states, a finite set of actions, and a reward structure defined for each state-action pair. Fig. 3.1 shows an example of a Markov decision process. It is assumed that the agent, the automated process that has to



**Fig. 3.1** Markov Decision Process.

make decisions, can be in one state at any given time. Actions are the things the agent can do. Associated with each action is an immediate reward. The effectiveness of an action depends on the state the agent is in thus an immediate reward is assigned for each combination of states and actions.

An MDP is a quadruple,  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$  where  $\mathcal{S}$  denotes a finite set of states with state  $i$  denoted by  $s_i$ ,  $\mathcal{A}$  denotes a finite set of actions with action  $i$  denoted by  $a_i$ ,  $\mathcal{P}$  denotes the transition probabilities for each action in each state, and  $\mathcal{R}$  denotes the reward structure.  $p_{ij}^a$  represents the probability of transitioning from state  $i$  to state  $j$  if the agent performs action  $a$ .  $r_{ij}^a$  represents the immediate reward the agent will acquire if it performs action  $a$  while in state  $s_i$  and moves to state  $s_j$ . The expected rewards, the reward expected to be received on average in the long term, is represented by

$$q_i^a = \sum_j p_{ij}^a r_{ij}^a. \quad (3.1)$$

That is, the expected reward is the weighted sum of rewards over all possible next states.

### 3.1.2 Policies

A policy tells the agent what action to execute. It can be a function or a mapping and typically depends upon the state the agent is in. Policies can be either deterministic or stochastic. Deterministic policies specify a single action to take in each state whereas stochastic policies specify a number of possible actions to execute in each state. Policies can be further categorized as stationary or non-stationary. A stationary policy only depends on the state of the agent and/or environment and is independent of time. In contrast, a non-stationary policy is dependent on time, where the action taken depends on when the policy will be used. A memoryless policy is one that uses no history when choosing an action. A  $k$ -memory policy chooses actions based on the last  $k$  pieces of the agent's history. The policy,  $\mu_i$ , is a complete mapping from all states to actions,  $\mu_i : \mathcal{S} \rightarrow \mathcal{A}$ . The action specified by the policy for state  $s_i$  is  $\mu_i(s_i)$ .

### 3.1.3 Horizon

Horizon refers to the number of decisions the agent needs to make and can be either finite or infinite. A finite  $k$ -horizon problem is a problem where the agent will make decisions for  $k$  time steps. The lifetime or size of the horizon is the period of  $k$  time steps. Finite horizon problems can have deterministic non-stationary policies, where a different policy exists for each time step.

An infinite horizon problem is a problem where the lifetime of the agent is unknown. Solving the optimal policy for the infinite horizon class of problems requires the addition of a discount factor such that distant future rewards are discounted more than rewards closer to the present time. The solution to infinite horizon problems

result in stationary policies.

### 3.1.4 Policies in MDPs

The value function is a metric used to measure the usefulness of a policy. Let the value function,  $V_t^{\mu}(s_i)$  be the expected reward that the agent can accumulate in its lifetime if it executes policy  $\mu = \{\mu_0, \mu_1, \dots, \mu_t\}$  for  $t$  steps. If the agent is in state  $s_i$  then there are  $t$  steps where the agent needs to make decisions given that it executes actions according to the policy  $\mu_m$  at time  $m$ . The value function is determined by

$$V_t^{\mu}(s_i) = \sum_j p_{ij}^{\mu(s_i)} [r_{ij}^{\mu(s_i)} + \beta V_{t-1}^{\mu}(s_j)]. \quad (3.2)$$

The reward for moving to a state is the immediate reward received for the current action, current state, and next state,  $r_{ij}^{\mu(s_i)}$ , plus the value of state  $s_j$  with one less step remaining,  $V_{t-1}^{\mu}(s_j)$ . The value function of the next step is discounted by  $\beta$  which is needed for infinite horizon problems. In the case of undiscounted finite horizon problems  $\beta = 1$ . To solve the infinite horizon case, the value function must converge thus requiring  $0 \leq \beta \leq 1$ .

The value function is used to determine the usefulness of a policy and is ultimately used to solve for the optimal policy. Optimality is considered with respect to the value of the state-action pairs which are derived from the reward structure. Let  $\mu^*$  denote the optimal policy and  $V_t^*$  denote the associated value function. An optimal policy is one where, for all state,  $s_i$ , and all other policies,  $\mu$ ,  $V^{\mu^*}(s_i) \geq V^{\mu}(s_i)$ . Finding  $\mu^*$  is a non-trivial task.

## 3.2 Partially Observable Markov Decision Process

Partially Observable Markov Decision Processes (POMDP) are similar to MDP except the underlying state of the system cannot be directly observed. However, with knowledge of the system dynamics the current system state can be inferred based on actions, observations, and rewards.

### 3.2.1 POMDP Model

A POMDP is defined by a hextuple,  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \Theta, \mathcal{B}, \mathcal{R} \rangle$ . Similar to the MDP model,  $\mathcal{S}$  denotes a finite set of states with state  $i$  denoted by  $s_i$ ,  $\mathcal{A}$  denotes a finite set of actions with action  $i$  denoted by  $a_i$ ,  $\mathcal{P}$  denotes the transition probabilities for each action in each state, and  $\mathcal{R}$  denotes the reward structure. However the main difference between the POMDP and MDP model is the addition of the finite set of observations,  $\Theta$ , and the observation model,  $\mathcal{B}$ . Let observation  $i$  be denoted by  $\theta_i$ . The agent cannot directly observe the underlying Markov process. Let  $b_{j\theta}^a$  denote the probability that the agent observes  $\theta$  when it is in state  $s_j$  at time  $t$  and when the last action (at  $t - 1$ ) was  $a$ . The immediate rewards,  $r_{ij\theta}^a$ , denotes the immediate reward received at time  $t+1$  for performing action  $a$  in state  $s_i$  at time  $t$ , moving to state  $s_j$  at time  $t + 1$ , and making observation  $\theta$ . Fig. 3.2 shows the Markovian dynamics where observations are made after an action is taken. Equivalently, observations could have been taken before actions.

Let  $q_i^a$  denote the immediate reward of performing action  $a$  in state  $i$

$$q_i^a = \sum_{j,\theta} p_{ij}^a b_{j\theta}^a r_{ij\theta}^a, \quad (3.3)$$

which is the immediate rewards,  $r_{ij\theta}^a$ , weighted by the transition and observation

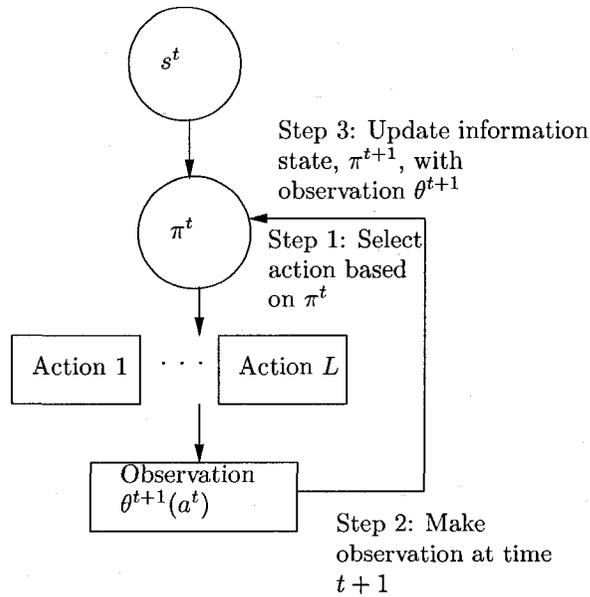


Fig. 3.2 POMDP Dynamic Model.

probabilities for the different actions and observations.

### 3.2.2 Information State

In the MDP model, the optimal policy was found by mapping states to actions. However in the POMDP case, the system state is not directly known but the agent can use observations to learn the most likely state it is in. Depending on the observation model, directly using the observations can result in a poor estimate of the current system state. The information state, also known as the belief vector, aids in determining the most likely state the system is in by storing all previous actions and observations in a summary statistic.

The information state is a probability distribution over the state space. Let  $\pi^t = \{\pi_0^t, \pi_1^t, \dots, \pi_{|S|}^t\}$  denote the information space where  $|S|$  is the number of states in the model and  $\pi_i^t$  represents the probability that we are currently in state  $s_i$  at time  $t$ . As will be shown later, knowledge of the system dynamics, the transition probabilities,

must be known in order to maintain an information state.

The information state is updated after each action and observation with the application of Bayes' rule:

$$\pi_j^{t+1} = \frac{\sum_i \pi_i^t p_{ij}^a b_{j0}^a}{\sum_{i,j} \pi_i^t p_{ij}^a b_{j0}^a}. \quad (3.4)$$

The resulting information state is a vector of probabilities computed using the above formula. The information transformation function is given by

$$\pi^{t+1} = T(\pi^t | a, \theta). \quad (3.5)$$

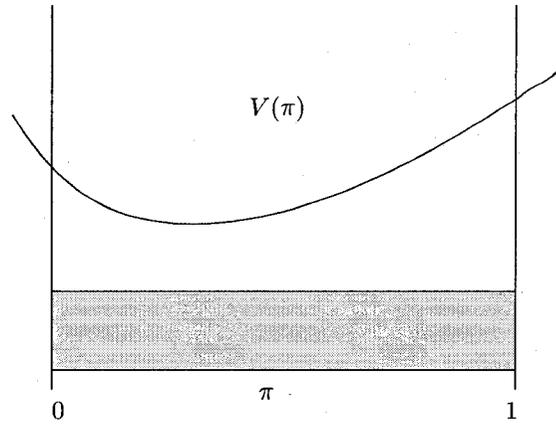
It is shown in [85] that the information state is a sufficient statistic for the past history of observations. That is, at any given time the information state represents all previous actions and observations.

### 3.2.3 Policies for POMDPs

In the MDP model a policy is a mapping from states to actions. However since the state is not known in the POMDP model, the policy maps the information states into actions. There is an infinite number of information states, since it is a probability distribution over all states, and storing the policy or value function in the form of tables, as can be done with the MDP model, is not feasible.

To solve for the policy the value function must be known. The basic form of the value function is similar to the MDP case except

- we must take the weighted average of all the states based on the information



**Fig. 3.3** Value function drawn over information space.

state

- we need to consider the weighted average value over all observations
- we need to consider the value for the next step based on the current information state, action, and observation.

The maximum value function for all actions is given by

$$V_t^*(\pi) = \max_a \left[ \sum_i \pi_i \sum_j p_{ij}^a \sum_{\theta} b_{j\theta}^a (r_{ij\theta}^a + V_{t-1}^*[T(\pi|a, \theta)]) \right]. \quad (3.6)$$

Simplified using the substitution for  $q_i^a$  the value function becomes

$$V_t^*(\pi) = \max_a \sum_i \pi_i q_i^a + \sum_{i,j,\theta} \pi_i p_{ij}^a b_{j\theta}^a V_{t-1}^*[T(\pi|a, \theta)]. \quad (3.7)$$

The value function drawn over the information state is shown in Fig. 3.3.

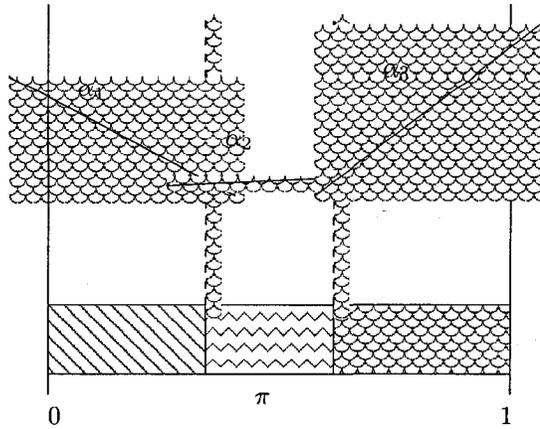


Fig. 3.4 PWLC Value Function.

An important property of the value function is that  $V_t^*(\pi)$  is piecewise linear and convex (PWLC) [85] and can be written simply as

$$V_t^*(\pi) = \max_k \sum_i \pi_i \alpha_i^k(t), \quad (3.8)$$

for some set of vectors  $\alpha_i^k(t) = \{\alpha_i^0(t), \alpha_i^1(t), \dots\}$ . The set of  $\alpha$ -vectors represents the coefficients of one of the linear pieces of a piecewise linear function. These piecewise linear functions can represent the value functions for each step in the finite horizon POMDP problem.

In Fig. 3.4, the value function is represented with the upper surface of 3 vectors  $\alpha_1, \alpha_2, \alpha_3$ . For 3 or more states, the value function can be represented with a hyper-plane through the information space and each hyper-plane can be represented with a vector of coefficients.

The value function is the dot product between the information state and the  $\alpha$ -vectors

$$\sum_i \alpha_i \pi_i = \alpha_0 \pi_0 + \alpha_1 \pi_1 + \dots + \alpha_{|S|} \pi_{|S|}. \quad (3.9)$$

At each time instant there is a set of  $\alpha$ -vectors and each vector has a corresponding action. We only need to find the vector that has the highest dot product with the information state to determine what action to take.

For infinite horizon problems the value function is represented by

$$V_t^*(\pi) = \max_a \sum_i \pi_i q_i^a + \beta \sum_{i,j,\theta} \pi_i p_{ij}^a b_{j\theta}^a V_{t-1}^*[T(\pi|a,\theta)], \quad 0 \leq \beta \leq 1. \quad (3.10)$$

Unlike finite horizon POMDP problems, the value function for infinite horizon POMDP problems are not necessarily piecewise linear [86]. In practice, non-linear value functions can be approximated with a piecewise linear function as closely as possible. Despite the uncertainty with a non-linear value function, it is shown in [86] that the value function is always convex.

### 3.3 Algorithms Overview

Various linear programming based algorithms have been presented to solve finite horizon POMDPs. These include

- Monahan's algorithm [87]: This intuitive and simple algorithm can only be used to solve problems with a small number of states.
- Sondik's algorithm [85, 86]: Sondik made a breakthrough in finding the optimal policy for general POMDP problems and all algorithms that followed are

derivations of Sondik's algorithm.

- Cheng's algorithm [88]: Cheng's Linear Support algorithm uses successive approximations of the value function to converge on the optimal value function by comparing the current approximation with the optimal value function at a set of points. This requires the use of an algorithm for finding all the corner points of the regions resulting in poor running times.
- Witness algorithm [14]: The witness algorithm is similar to Cheng's Linear Support algorithm except instead of having to discover all of the corner points of a region, a linear program finds points where the approximation differs from the optimal value function. This algorithm has better running times compared to Cheng's algorithm.
- Incremental pruning algorithm [89]: Incremental pruning combines elements of Monahan's enumeration and the witness algorithms. Similar to the witness algorithm, it considers constructing sets of vectors for each action individually and then focusing on each action one observation at a time.

### 3.4 Summary

In this chapter we presented the Markov decision process model which can be formulated as a quadruple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R} \rangle$ . Unlike MDP, in the partially observable Markov decision process model the underlying system state cannot be directly observed and the model can be formulated as a hextuple  $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \Theta, \mathcal{B}, \mathcal{R} \rangle$ . A discrete POMDP can be converted to a continuous MDP by using an information space. The value function for a POMDP is piecewise linear and convex (PWLC) and this property is exploited to solve POMDP problems. A brief overview of common algorithms to solve POMDP problems was also presented in this chapter.

## Chapter 4

# Proposed Multimedia Transmission over Cognitive Radio Networks

### 4.1 Overview

In this section we investigate a hierarchical access spectrum overlay DSA CR system. An integrated CLD approach is used to adaptively select the intra-refresh rate to minimize distortion with time varying wireless channels. We consider practical implementations by considering spectrum sensor error, CSI error, and hardware restrictions on the number of channels that can be sensed simultaneously. Both spectral and temporal opportunities will be exploited.

### 4.2 System Model

Consider a spectrum that consists of  $N$  channels, each with bandwidth  $W(n)$ ,  $1 \leq n \leq N$ . These  $N$  channels are licensed to primary users. Time is divided into slots of equal length  $T$ . Slot  $k$  refers to the discrete time period  $[kT, (k+1)T]$ .

When the slot is not in use by primary users, it will be comprised of AWGN noise

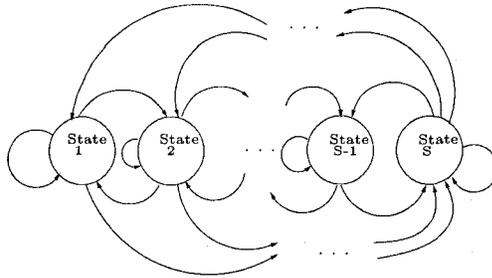


Fig. 4.1 Finite State Markov Channel with primary network usage.

and fading. The fading process and primary usage for a channel can be represented by a stationary and ergodic  $S$ -state Markov chain as shown in Fig. 4.1. Let  $i$  and  $\gamma$  denote the instantaneous channel state and fading gain, respectively. When the channel is in state  $i$ , the quantized fading gain is  $\gamma_i$ , where  $\gamma_i \leq \gamma \leq \gamma_{i+1}$ ,  $1 \leq i \leq S-1$ . When the channel is in state  $i = S$ , the channel is in use by the primary network. We assume that the phase of the channel attenuation can be perfectly estimated and removed at the receiver. The  $S$ -state Markov channel model is completely described by its stationary distribution of each channel state  $i$ , denoted by  $p(i)$ , and the probability of transitioning from state  $i$  into state  $j$  after each time slot, denoted by  $\{P_{i,j}\}$ ,  $1 \leq i, j \leq S$ .

In general, a finite state Markov channel (FSMC) model is constructed for a particular fading distribution by first partitioning the range of the fading gain into a finite number of sections. Then each section of the gain value corresponds to a state in the Markov chain. The application of FSMC to model Rayleigh channels has been well studied in [90,91]. Given knowledge of the fading process and primary network usage, the stationary distribution  $p(i)$  as well as channel state transition probabilities  $\{P_{i,j}\}$  can be derived. Once a channel gain has been determined for states  $1,2,\dots, S-1$ , the packet loss ratio is determined for each state based on the modulation and channel coding schemes. The intra refreshing rate that minimizes the total distortion

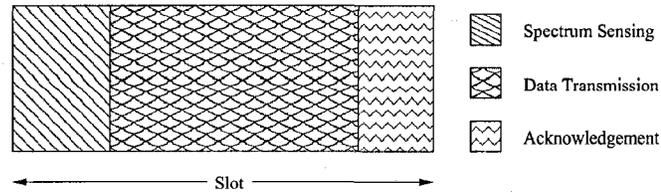


Fig. 4.2 The slot structure.

for each state can then be calculated using the Rate-Distortion model.

In the system, transmission time is divided into slots. The structure of each slot is shown in Fig. 4.2. At the beginning of a slot, the transmitter of secondary users will select a set of channels to sense. Based on the sensing outcome, the transmitter will decide whether or not to access a channel. If the transmitter decides to access a channel, some application layer parameters will be selected and the video content will be transmitted. At the end of the slot, the receiver will acknowledge the transfer by sending the perceived channel gain back to the transmitter. We will assume a system for real-time multimedia applications where packets are discarded if a primary user is using the slot or if the channel is not accessed. The system block diagram showing video transmission between two secondary users is shown in Fig. 4.3.

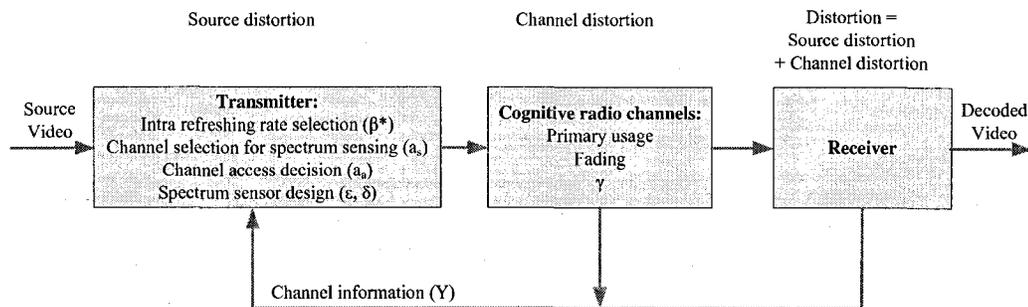


Fig. 4.3 The block diagram of multimedia transmission over cognitive radio networks.

### 4.3 Rate-Distortion (R-D) Model for Multimedia

#### Applications

We adaptively select a video encoding parameter, the intra-refresh rate, based on channel condition to minimize distortion. We will use the DCT-based rate-distortion model described in [10] in our study. The R-D model facilitates adaptive intra-mode selection and joint source-channel rate control.

The total end-to-end distortion comprises of  $D_s$ , the quantization distortion introduced by the lossy video encoder to meet a target bit rate, and  $D_c$ , the distortion resulting from channel errors. For DCT-based video coding, intra coding of a MB or a frame usually requires more bits than inter coding since inter coding removes the temporal redundancy between two neighboring frames. Let  $\beta$  be the intra refreshing rate, the percentage of MBs coded with intra mode. Inter coding of MBs has much better R-D performance than intra mode. Decreasing the intra refreshing rate decreases the source distortion for a target bit rate. However inter coding relies on information in previous frames. Packet losses due to channel errors result in error propagation along the motion-compensation path until the next intra coded MB is received. Increasing the intra refreshing rate decreases the channel distortion. Thus we have a tradeoff between source and channel distortion when selecting the intra refreshing rate. We aim to find the optimal  $\beta$  to minimize the total end-to-end distortion given the channel bandwidth and packet loss ratio.

The source distortion is given by

$$D_s(R_s, \beta) = D_s(R_s, 0) + \beta(1 - \eta + \eta\beta)[D_s(R_s, 1) - D_s(R_s, 0)], \quad (4.1)$$

where  $R_s$  denotes the source coding rate,  $\beta$  is the intra refreshing rate, and  $\eta$  is

a constant based on the video sequence.  $D_s(R_s, 0)$  and  $D_s(R_s, 1)$  denotes the time average all inter- and intra-mode selection for all frames over  $K$  time slots.

$$D_s(R_s, 0) = \frac{1}{K} \sum_{k=1}^K \frac{1}{M_k} \sum_{m=1}^{M_k} D_s(R_s, 0, m), \quad (4.2)$$

$$D_s(R_s, 1) = \frac{1}{K} \sum_{k=1}^K \frac{1}{M_k} \sum_{m=1}^{M_k} D_s(R_s, 1, m), \quad (4.3)$$

where  $M_k$  is the number of inter/intra frames in time slot  $k$ . The average channel distortion for each time slot is given by

$$D_c(p, \beta) = \left( \frac{a}{1 - b + b\beta} \right) \left( \frac{p}{1 - p} \right) E[F_d(m, m - 1)], \quad (4.4)$$

where  $p$  is the packet loss rate,  $b$  is a constant describing motion randomness of the video scene,  $a$  is the energy loss ratio of the encoder filter, and  $E[F_d(m, m - 1)]$  is the average value of the frame difference  $F_d(m, m - 1)$  over  $K$  slots. We will use the same error concealment strategy and packet loss ratio derivation as described in [10].

The total average distortion is given by

$$D(R_s, p, \beta) = D_s(R_s, \beta) + D_c(p, \beta). \quad (4.5)$$

The optimum  $\beta^*$  is then selected to minimize the total distortion.

$$\beta^* = \arg \min_{\beta} D(R_s, p, \beta). \quad (4.6)$$

## 4.4 Solving the Multimedia Transmission over Cognitive Radio Networks Problem

In CR networks with multimedia applications, we need to determine the optimal policy for channel sensing selection, sensor operating point, access decision, and intra refreshing rate to minimize application layer distortion subject to the system probability of collision. With channel sensing and CSI errors, the system state cannot be directly observed. We formulate the whole system as a partially observable Markov decision process (POMDP). Deriving a single POMDP formulation for all policies under the probability of collision constraint would result in a constrained POMDP. However, constrained POMDPs require randomized policies to achieve optimality, which is often intractable. Therefore, we use the separation principle in [31] for the sensor operating point and the access decision. The spectrum sensor operating point is set such that  $\delta = \zeta$ , where  $\delta$  is the probability of miss detection of the busy channel used by primary users and  $\zeta$  is the required probability of collision.

The sequence of operations in a slot is shown in Fig. 4.4. At the beginning of the slot, the system transitions to a new state. Using a POMDP derived policy, a channel is selected for spectrum sensing. An access decision is then made based on the sensing observation. Using the belief of the channel state, an intra refreshing rate is selected. The receiver acknowledges the transfer by sending the quantized perceived channel gain back to the secondary transmitter. The immediate cost for the time slot is derived based on the previous operations in the slot.

The system can be formulated as a POMDP with states, actions, transition probabilities, observations, and cost structures as follows.

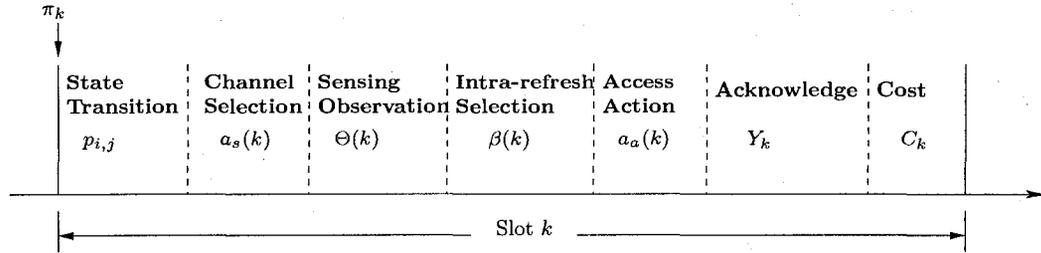


Fig. 4.4 Sequence of operations in a slot.

#### 4.4.1 State Space, Transition Probabilities and Observation Space

The system state is given by the network usage of primary users and channel state information. Let  $\{X(n)\}$  denote an  $S$ -state Markov chain for channel  $n$ ,  $X(n) \in \mathbb{X} = \{e_1, e_2, \dots, e_{S-1}, e_S\}$ , where  $e_i$  denotes the  $S$ -dimensional unit vector with 1 in the  $i$ th position and zeros elsewhere. The system with  $N$  channels is modeled as a discrete-time homogeneous Markov process with  $S^N$  states. The system state in time slot  $k$  is given by  $V_k = [X_k(1), \dots, X_k(N)]$ .

To simplify the presentation, we consider a system with a single channel in the formulation. It is straightforward to extend the formulation to include multiple channels which is considered in our simulations. For the system with a single channel,  $V_k = X_k$ . The transition probabilities of the system state are given by the  $S \times S$  matrix  $A$ . We assume the transition probabilities are known based on network usage and channel fading characteristics.

The observation available to the secondary transmitter and receiver is the sensed channel and channel gain acknowledgment,  $Y_k \in \mathbb{Y}$ , where

$\mathbb{Y} = \{\gamma_1, \dots, \gamma_{S-1}, \gamma_S$  (The channel is used by primary users)} and  $\gamma_i < \gamma_j, \forall i < j$ .

The spectrum sensor observation,  $\Theta(k) \in \{0(\text{busy}), 1(\text{idle})\}$ , may be different at the transmitter and receiver. If the transmitter and receiver use the same observations to derive the information state (described in the following Subsection), then the

information state can be used to maintain frequency hopping synchronization [31]. Thus the information state will be updated with  $Y_k$  and will not include the spectrum sensor observation.

Let  $B(y, x, a) = \Pr\{y|x, a\}$  denote the conditional probability of observing  $y$  given that the system state is in state  $x$  and composite action  $a$  was taken.

$$B(y, x, a) = \begin{cases} P_{ce}(x, v(y))(1 - \epsilon), & \text{if } y \neq \gamma_S, x \neq e_S, \\ \epsilon, & \text{if } y = \gamma_S, x \neq e_S, \\ 0, & \text{if } y \neq \gamma_S, x = e_S, \\ 1, & \text{if } y = \gamma_S, x = e_S \end{cases}, \quad (4.7)$$

where  $\epsilon$  is the probability of miss detection of the idle channel and  $v(y) = i, 1 < i < S$  given  $y = \gamma_i$ . When the channel is available and accessed, the probability of channel estimate by the receiver is given by  $P_{ce}(x, v(y))$ .

Using the work from Hoang and Motani [92], we assume the channel estimation error has a Gaussian distribution with zero mean and  $\sigma^2$  variance. At a particular time and channel, the estimated channel gain is

$$\hat{\gamma} = \gamma_i + w, \quad (4.8)$$

where  $\gamma_i$  is the actual channel gain and  $w$  is a Gaussian random variable with zero mean and  $\sigma^2$  variance. The receiver then quantizes the channel gain to the nearest

possible value. The probability that  $\hat{\gamma}$  is closest to  $\gamma_j$  is given by

$$P_{ce}(i, j) = \begin{cases} \frac{1}{2} \left[ \operatorname{erf} \left( \frac{\gamma_j + \gamma_{j+1} - 2\gamma_i}{2\sqrt{2}\sigma} \right) - \operatorname{erf} \left( \frac{\gamma_j + \gamma_{j-1} - 2\gamma_i}{2\sqrt{2}\sigma} \right) \right], & \text{if } j \neq e_1, e_{S-1}, e_S \\ \frac{1}{2} \left[ 1 + \operatorname{erf} \left( \frac{\gamma_1 + \gamma_2 - 2\gamma_i}{2\sqrt{2}\sigma} \right) \right], & \text{if } j = e_1 \\ \frac{1}{2} \left[ 1 - \operatorname{erf} \left( \frac{\gamma_{S-2} + \gamma_{S-1} - 2\gamma_i}{2\sqrt{2}\sigma} \right) \right], & \text{if } j = e_{S-1} \\ 0, & \text{if } j = e_S \end{cases}, \quad (4.9)$$

where  $\operatorname{erf}()$  denotes the error function.

#### 4.4.2 Information State

Information state is an important concept in POMDP. We will refer to a probability distribution over states as the information state and the entire probability space (the set of all possible probability distributions) as the information space. The information spaces for 2-state and 3-state systems are shown in Fig. 4.5. For a system with two states, its information space is a one-dimension line. The distance from the right end is the first component  $\pi(1)$  and the distance from the left end is the second component  $\pi(2)$ . For the system with 3 states, its information space is a two-dimension triangle. The value of a point in the information space can be obtained from the perpendicular distance from the sides of the triangle. An information state is a sufficient statistic for the decision and observation history.

#### 4.4.3 Action Space

Due to hardware limitations, we will assume that a secondary user is equipped with a single Neyman-Pearson energy detector and can only sense  $L = 1$  channel at each time instant. In each slot  $k$ , the secondary user needs to decide whether or not to sense, determine which sensor operating point on the Receiver Operating Curve (ROC) curve to use, whether to access the channel, and which quantized intra refreshing rate to

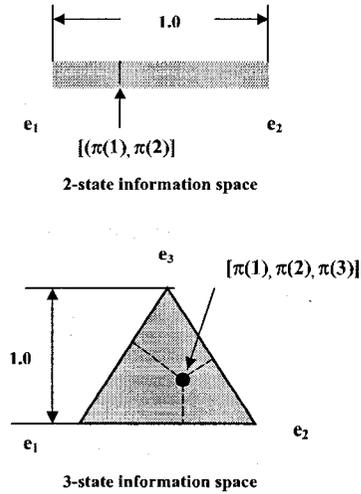


Fig. 4.5 Information state in POMDP.

use. Thus the action space consists of four parts: a channel selection decision  $a_s(k) \in \{0(\text{no sense}), 1(\text{sense})\}$ , a spectrum sensor design  $(\epsilon(k), \delta(k)) \in \mathbb{A}_{\epsilon\delta}$  where  $\mathbb{A}_{\epsilon\delta}$  are valid points on the ROC curve, an access decision  $a_a(k) \in \{0(\text{no access}), 1(\text{access})\}$ , and an intra refreshing rate  $\beta(k) \in \mathbb{A}_\beta$ . The composite action in slot  $k$  is denoted by

$$a_k = \{a_s(k), (\epsilon(k), \delta(k)), a_a(k), \beta(k)\} \in (\{0, 1\}, \mathbb{A}_{\epsilon\delta}, \{0, 1\}, \mathbb{A}_\beta). \quad (4.10)$$

Due to sensing and channel estimation errors, a secondary user cannot directly observe the true system state. It can infer the system state from its decision and observation history encapsulated by the information state. Information state  $\pi_k = \{\lambda_x(k)\}_{x \in \mathbb{X}} \in \Pi(\mathbb{X})$  where  $\lambda_x(k) \in [0, 1]$  denotes the conditional probability (given decision and observation history) that the system state is in  $x \in \mathbb{X}$  at the beginning of slot  $k$  prior to state transition.  $\Pi(\mathbb{X}) = \{\lambda_x(k) \in [0, 1], \sum_{x \in \mathbb{X}} \lambda_x = 1\}$  denotes the information space that includes all possible probability mass functions on the state

space  $\mathbb{X}$ .

At the end of the time slot, the transmitter receives observation  $Y_k$ . The information state is then updated using Bayes' rule *before* state transition

$$\lambda_x(k+1) = \frac{\sum_{x' \in \mathbb{X}} \lambda_{x'}(k) A_{x',x} B(y_k, x_k, a_k)}{\sum_{x \in \mathbb{X}} \sum_{x' \in \mathbb{X}} \lambda_{x'}(k) A_{x',x} B(y_k, x_k, a_k)}. \quad (4.11)$$

Given information vector  $\pi_k$  the distribution of the system state  $X_k$  in slot  $k$  *after* state transition is then given by

$$\Pr\{X_k = x\} = \sum_{x' \in \mathbb{X}} \lambda_{x'}(k) A_{x',x} \quad \forall x \in \mathbb{X}. \quad (4.12)$$

#### 4.4.4 Cost and Policy

From a user's point of view, QoS at application layer is more important than at other layers. Therefore, we model multimedia distortion as the immediate cost in our scheme. The immediate cost in time slot  $k$  is defined as

$$C_k = D(R, p(x_k, a_k), \beta(k)), \quad (4.13)$$

where  $R$  is the target bit rate and  $p(x_k, a_k)$  denotes the packet loss ratio when the system is in state  $x_k$  and composite action  $a_k$  is taken in time slot  $k$ . We assume  $a_a(k) = 0$  (no access) is the equivalent to 100% packet loss.

The expected total cost of the POMDP represents the overall distortion for a video sequence transmitted over  $K$  slots and can be expressed as

$$J_\mu = \mathbb{E}_{(\mu_s, \mu_{e\delta}, \mu_a, \mu_\beta)} \left[ \sum_{k=1}^K D(R, p(x_k, a_k), \beta(k)) \right], \quad (4.14)$$

where  $\mathbb{E}_{\{\mu_s, \mu_{\epsilon\delta}, \mu_a, \mu_\beta\}}$  indicates the expectation given that policies  $\mu_s, \mu_{\epsilon\delta}, \mu_a, \mu_\beta$  are employed.

A channel sensing policy  $\mu_s$  specifies a channel to sense,  $a_s$ . A sensor operating policy  $\mu_{\epsilon\delta}$  specifies a spectrum sensor design  $(\epsilon, \delta) \in \mathbb{A}_{\epsilon\delta}$  based on the system tolerable probability of collision,  $\zeta$ . An access policy  $\mu_a$  specifies the access decision  $a_a \in \{0, 1\}$ . An intra refreshing policy  $\mu_\beta$  specifies the intra refreshing decision  $\beta \in \mathbb{A}_\beta$  based on the current information state  $\pi_k$ .

#### 4.4.5 Objective and Constraint

We aim to develop the joint design of an optimal policy for multimedia transmission over CR networks,  $\{\mu_s^*, \mu_{\epsilon\delta}^*, \mu_a^*, \mu_\beta^*\}$ , that minimizes the expected total distortion in  $K$  slots under the collision constraint  $P_c$ .

$$\begin{aligned} \{\mu_s^*, \mu_{\epsilon\delta}^*, \mu_a^*, \mu_\beta^*\} = \\ \arg \min_{\mu_s, \mu_{\epsilon\delta}, \mu_a, \mu_\beta} \mathbb{E}_{\{\mu_s, \mu_{\epsilon\delta}, \mu_a, \mu_\beta\}} \left[ \sum_{k=1}^K D(R, p(x_k, a_k), \beta(k)) \right] \end{aligned} \quad (4.15)$$

subject to

$$P_c(k) = \Pr\{a_a(k) = 1 | X_k = e_S\} < \zeta, \forall k \in K. \quad (4.16)$$

#### 4.4.6 Value Function

Let  $J_k(\pi)$  be the value function that represents the minimum expected cost that can be obtained starting from slot  $k$  ( $1 \leq k \leq K$ ) given information state  $\pi_k$  at the beginning of slot  $k$ . Given that the secondary user takes action  $a_k$  and observes acknowledgment  $Y_k = y_k$ , the cost that can be accumulated starting from slot  $k$  consists of the immediate cost  $C_k = D(R, p(x_k, a_k), \beta(k))$  and the minimum expected future cost  $J_{k+1}(\pi + 1)$ .  $\pi_{k+1} = \{\lambda_x(k+1)\}_{x \in \mathbb{X}} = U(\pi_k | a_k, y_k)$ , which represents the updated knowledge of system state after incorporating the action  $a_k$  and the

acknowledgment  $y_k$  in slot  $k$ . The sensing policy is then given by

$$J_k(\pi_k) = \min_{a \in \mathbb{A}} \sum_{x \in \mathbb{X}} \sum_{x' \in \mathbb{X}} \lambda_{x'}(k) A_{x',x} \sum_{j=e_1}^{e_S} B(y_k, j, a_k) [D(R, p(x_k, a_k), \beta(k)) + J_{k+1}(U(\pi_k | a_k, y_k))], 1 \leq k \leq K - 1 \quad (4.17)$$

$$J_K(\pi_K) = \min_{a \in \mathbb{A}} \sum_{x \in \mathbb{X}} \sum_{x' \in \mathbb{X}} \lambda_{x'}(K) A_{x',x} \times \left[ \sum_{j=e_1}^{e_S} B(y_K, j, a_K) D(R, p(x_K, a_K), \beta(K)) \right]. \quad (4.18)$$

The value function of an unconstrained POMDP with finite action space is piecewise-linear convex and can be solved using linear programming techniques [85]. An excellent overview of computationally efficient algorithms are given in [14] and can be used to solve for the optimum sensing policy.

#### 4.4.7 Intra Refreshing Strategy

For a selected channel, the optimum  $\beta$  selected corresponds to the most likely available state based on  $\pi_k$ . Due to the asymptotic nature of the channel distortion, a busy or unaccessed channel has infinite distortion. In this case,  $\beta$  has no influence on the total distortion. If the most likely state based on  $\pi_k$  corresponds to a busy state then the optimum  $\beta$  is to select a  $\beta$  corresponding to the most likely available state. That way if the information state suggests the channel is busy but in reality it is available then a  $\beta$  has been selected that will minimize the effect of this error.

### 4.5 Summary

In this chapter we propose a wireless multimedia transmission scheme over CR. Intra-refresh rate, channel selection, channel access, and spectrum sensor strategies are determined concurrently under the framework of a POMDP. A low complexity scheme

has been proposed. The intra-refresh rate is selected based on the most likely state indicated by the information state. By employing the separation principle [31], myopic policies for the access and sensor operating point are derived based on the probability of collision constraint. Only the channel selection policy requires solving a POMDP.

# Chapter 5

## Simulation Results

In this section, we illustrate the performance of the proposed scheme by numerical examples. We consider the system performance in the following four cases: (1) using perfect knowledge of the system thus making optimal decisions, which is the best case possible, (2) making decisions based on the most likely state indicated by the information state, which is our proposed scheme, (3) making decisions solely based on the channel gain provided in the last acknowledgment, and (4) using a constant  $\beta$ , which represents existing schemes that do not consider application layer QoS. Our goal is to compare the distortion of different schemes as opposed to determining the absolute distortion. We use an average distortion metric that refers to the average distortion over the time slots when the channel is available and accessed. We first consider a system with one channel in Subsections 5.2. Then, we consider a system with two channels in Subsections 5.3.

### 5.1 Selection of Rate-Distortion Parameters

We use the rate-distortion model in equation 4.5 in our simulations. Building an adaptive intra-refresh encoder is a non-trivial task and is beyond the scope of this

work. Video rate-distortion parameters,  $D_s(R_s, 0)$ ,  $D_s(R_s, 1)$ ,  $\eta$ ,  $E[F_d(m, m - 1)]$ ,  $a$ , and  $b$ , are arbitrarily selected and remain constant for the duration of the simulation. Various rules were used in parameter generation to generate feasible results.  $D_s(R_s, 0) > 0$ ,  $D_s(R_s, 1) > 0$ ,  $D_c(p, \beta) > 0$ ,  $0 \leq \beta \leq 1$ ,  $0 \leq p \leq 1$ ,  $0 \leq a \leq 1$ ,  $b > 0$ ,  $E[F_d(m, m - 1)] > 0$ ,  $D_s(R_s, 1) > D_s(R_s, 0)$ . By using these rules we generated the following parameters that were used for all simulations.  $D_s(R_s, 0) = 74$ .  $D_s(R_s, 1) = 124$ .  $\eta = 1.4$ .  $a = 0.01$ .  $b = 1.0$ .  $E[F_d(m, m - 1)] = 100$ .

## 5.2 One Channel Scenario

### 5.2.1 Performance Improvement

Fig. 5.1 shows the distortion of different schemes. The number of states refers to  $S - 1$  quantized channel gains and one busy channel state. For simplicity we derive a transition matrix based on the probability that any available state stays in the same state,  $\Pr\{X_{k+1} = v | X_k = v\}$ , the probability of transitioning from an available state to a busy state,  $\Pr\{X_{k+1} = z | X_k = v\}$ , and the probability of a busy state staying busy,  $\Pr\{X_{k+1} = z | X_k = z\}$ , where  $v \in \{e_1, e_2, \dots, e_{S-1}\}$  and  $z = e_S$  indicate available and busy states, respectively. The following parameter values are used in this example.  $\Pr\{X_{k+1} = v | X_k = v\} = 0.85$ ,  $\Pr\{X_{k+1} = z | X_k = v\} = 0.05$ ,  $\Pr\{X_{k+1} = z | X_k = z\} = 0.1$ ,  $\epsilon = 0.6$ ,  $\sigma = 0.1$ . From Fig. 5.1, we can see that when perfect knowledge of the channel state is available, perfect decisions can be made for each time slot thus method (1) has the lowest average distortion. The more realistic cases occur in the presence of sensing and CSI errors. Our proposed method (i.e., method 2) uses the information state to select the most likely optimal decisions. This method tracks the ideal case fairly closely. Both method 3 and method 4 have worse performance compared to the proposed scheme. This illustrates the performance improvement of

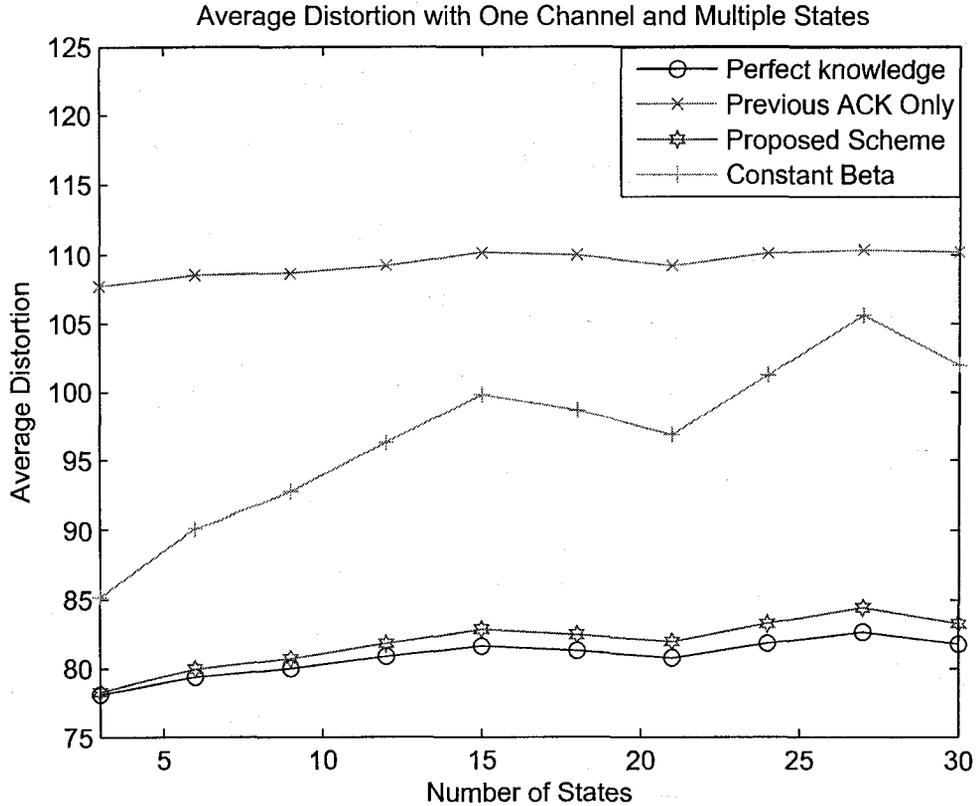


Fig. 5.1 Average distortion vs. the number of states in different schemes.

the proposed scheme over existing schemes. In addition, we also notice that using a constant  $\beta$  (i.e., method 4) can be worse than making decisions based solely on the previous acknowledgment (i.e., method 3), which shows the need to consider application layer parameters and application QoS.

### 5.2.2 Effects of the Parameters in the State Transition Matrix

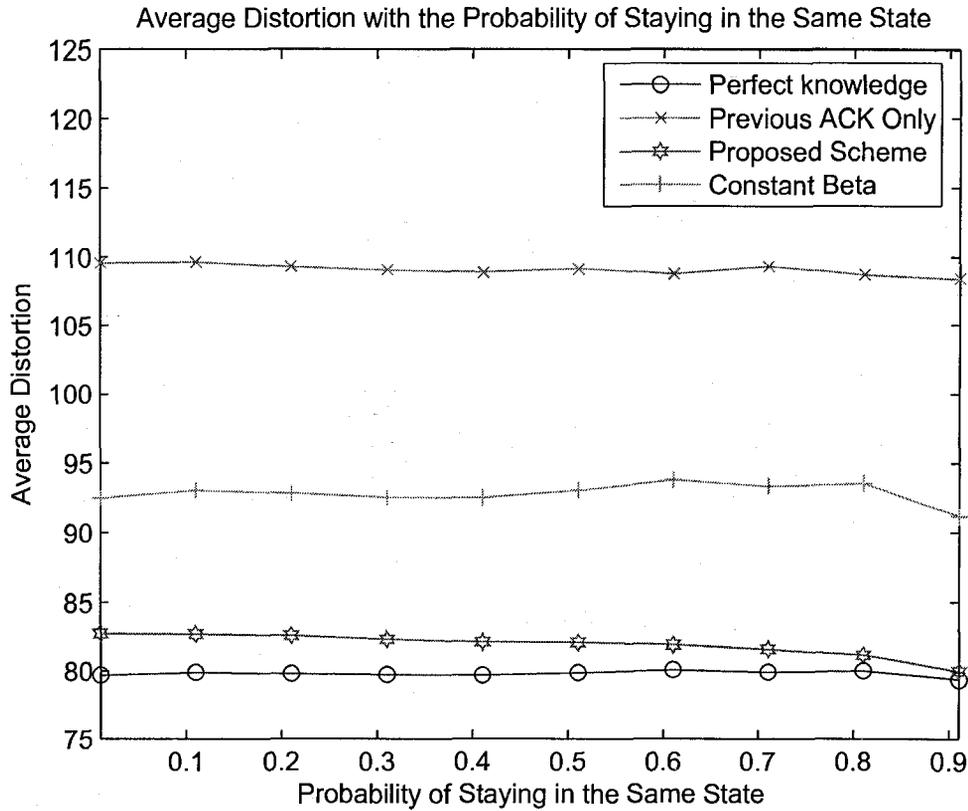
We evaluate how the parameters in the transition matrix affect the average distortion. The transition matrix can be selected based on channel fading and primary usage. We ignore quantization errors caused by the limited number of states and assume the actual channel gain matches the state channel gain. Fig. 5.2 and Fig. 5.3 show

the numerical results across  $\Pr\{X_{k+1} = v|X_k = v\}$  and  $\Pr\{X_{k+1} = z|X_k = v\}$ , respectively. In Fig. 5.2, there are 5 states.  $\epsilon = 0.6$ .  $\Pr\{X_{k+1} = z|X_k = v\} = 0.05$ . This example demonstrates the cognitive nature of the system. Our proposed method (i.e., method 2) approaches the method of using perfect knowledge of the channel state as  $\Pr\{X_{k+1} = v|X_k = v\}$  approaches 1. That is, the performance improves as the system dynamics slows down since it is easier to predict the actual system state. 5 states are used in Fig. 5.3.  $\epsilon = 0.6$ .  $\Pr\{X_{k+1} = v|X_k = v\} = 0.50$ . From this figure, we can see that  $\Pr\{X_{k+1} = z|X_k = v\}$  has little impact to the performance of the proposed method. The reason for this observation is that increasing  $\Pr\{X_{k+1} = z|X_k = v\}$  will increase the likelihood the system transitions to the busy state, which has little affect on the average distortion when the channel is available and accessed.

### 5.2.3 Effects of the Parameters in the Observation Matrix

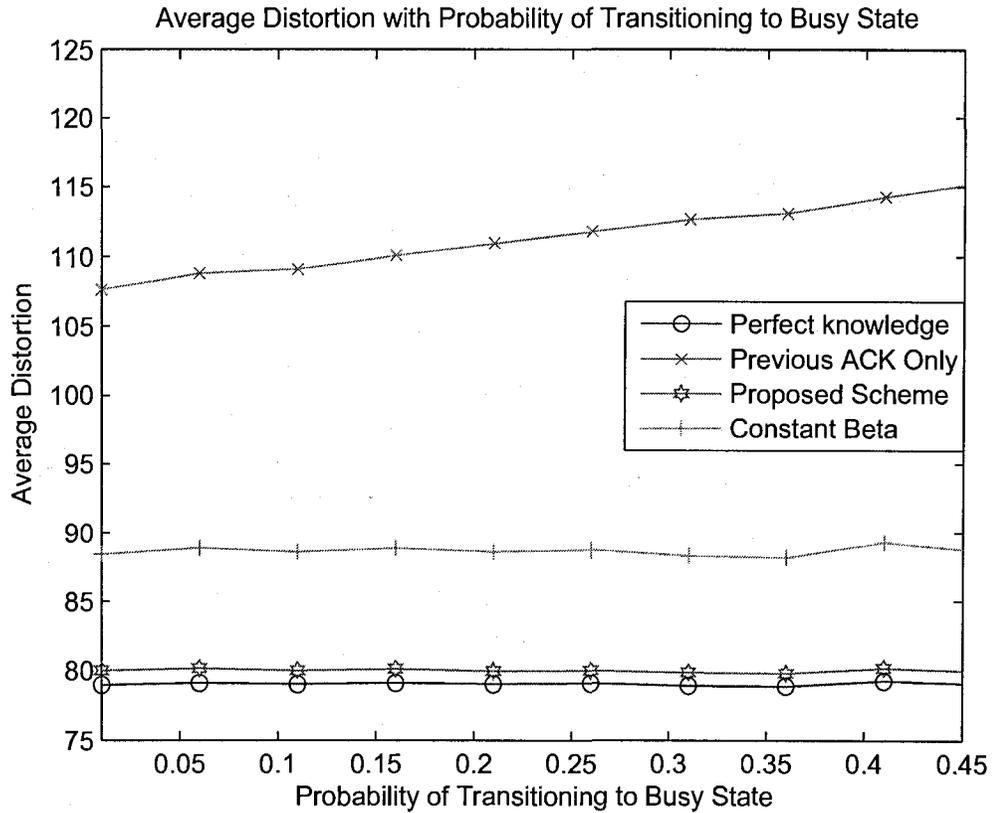
The observation matrix is derived from the sensor operating point,  $\epsilon$ , and the standard deviation of the receiver channel estimation error,  $\sigma$ . Fig. 5.4 and Fig. 5.5 show how  $\sigma$  and  $\epsilon$  affect the average distortion. The following parameters are used in Fig. 5.4. There are 5 states,  $\epsilon = 0.6$ ,  $\Pr\{X_{k+1} = v|X_k = v\} = 0.85$ , and  $\Pr\{X_{k+1} = z|X_k = v\} = 0.05$ . We can see from Fig. 5.4, as the receiver estimation degrades, the acknowledgment provides less information on the actual channel gain and the average distortion of our method increases.  $\epsilon$  and  $\delta$  are related based on the sensor ROC, and adjusting  $\epsilon$  implies a change to the system probability of collision requirement. In Fig. 5.5,  $\Pr\{X_{k+1} = v|X_k = v\} = 0.85$ ,  $\Pr\{X_{k+1} = z|X_k = v\} = 0.05$ , and  $\sigma = 0.1$ . This figure shows that the average distortion increases as the probability of false alarm increases.

These numerical results demonstrate some interesting trends in the design and optimization of CR networks from a cross-layer design perspective. Adaptively ad-



**Fig. 5.2** Average distortion vs. state transition matrix parameters (Case I: The probability of staying in the same state).

justing the intra-refresh rate to accommodate time varying wireless channels is an effective way to reduce distortion. By using all previous actions and observations we can build an information state that becomes more accurate over time. Performance of using the information state to select the intra-refresh rate improves as the system dynamics slows down. In a CR environment the MAC access strategy is derived from the accuracy of the spectrum sensor. The total distortion is limited to the availability of the channel. Distortion performance will degrade if primary usage increases or a very low system tolerable probability of collision is required.



**Fig. 5.3** Average distortion vs. state transition matrix parameters (Case II: The probability of transitioning to the busy state).

### 5.3 Two Channel Scenario

In this case we explore a  $N = 2$  channel with  $S = 3$  states per channel system to evaluate the performance of the channel selection policy. We will use a spectrum utilization (SU) metric to evaluate the sensor policy performance. SU represents the percentage of time slots where an available channel was selected for sensing. SU is an important parameter when evaluating video QoS. The channel distortion is infinite when a channel is busy or not accessed. Improving the SU will reduce the percentage of time slots where a busy channel was selected for sensing thus improving the application layer QoS. The application layer QoS is improved using a two step process. First we

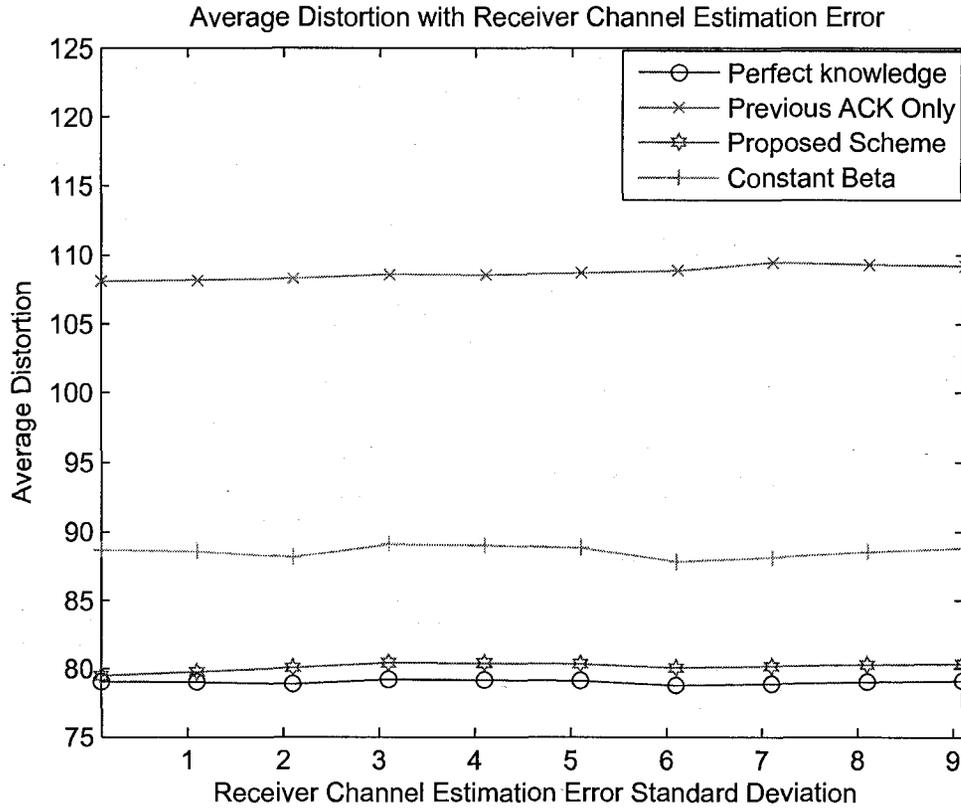


Fig. 5.4 Average distortion vs. observation matrix parameters (Case I: Receiver channel estimation standard deviation,  $\sigma$ ).

select a channel to maximize SU thus reducing the large distortion introduced when the channel is unavailable. Second for an available and accessed channel, we select the intra-refresh rate to minimize distortion for a particular channel gain.

The two channels, channel 1 and channel 2, are simulated having the same number of states (i.e. quantized channel gains) and observation probabilities but asymmetric transition probabilities. Channel 2 will have a higher primary usage than channel 1. Based on previous observations, actions, and the POMDP derived policy the secondary transmitter/receiver pair dynamically selects the channel that will most likely maximize application layer QoS.

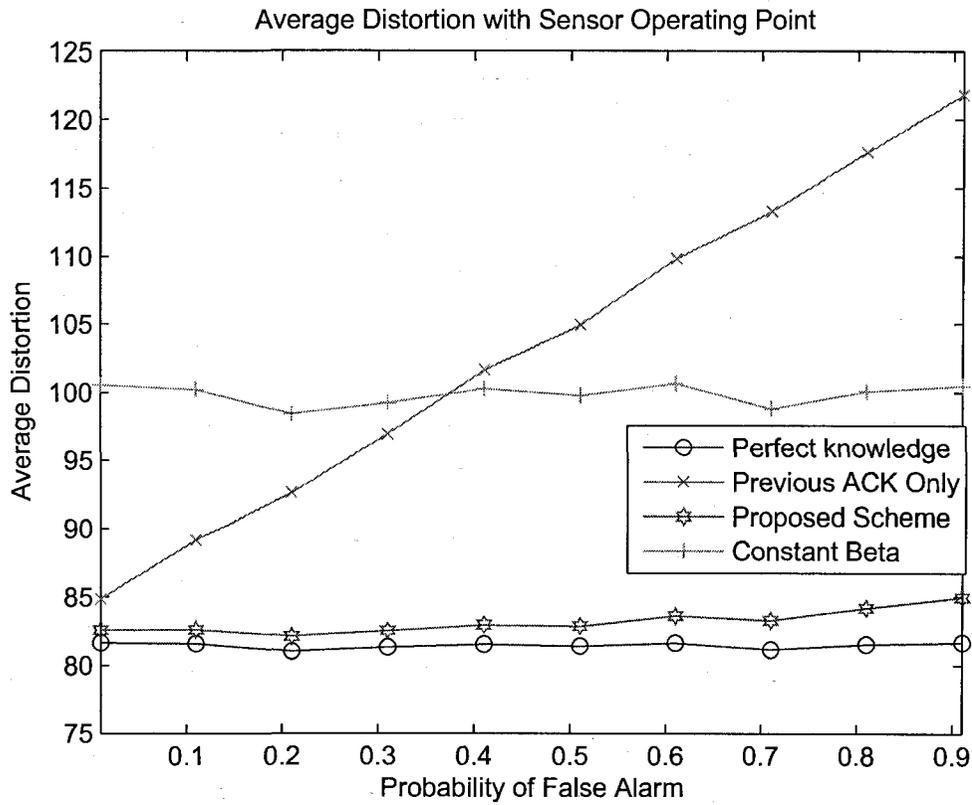


Fig. 5.5 Average distortion vs. observation matrix parameters (Case II: Sensor operating point,  $\epsilon$ ).

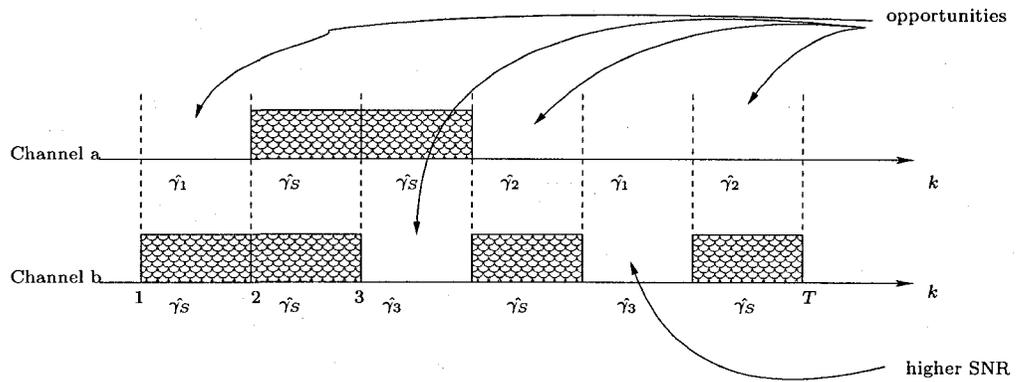
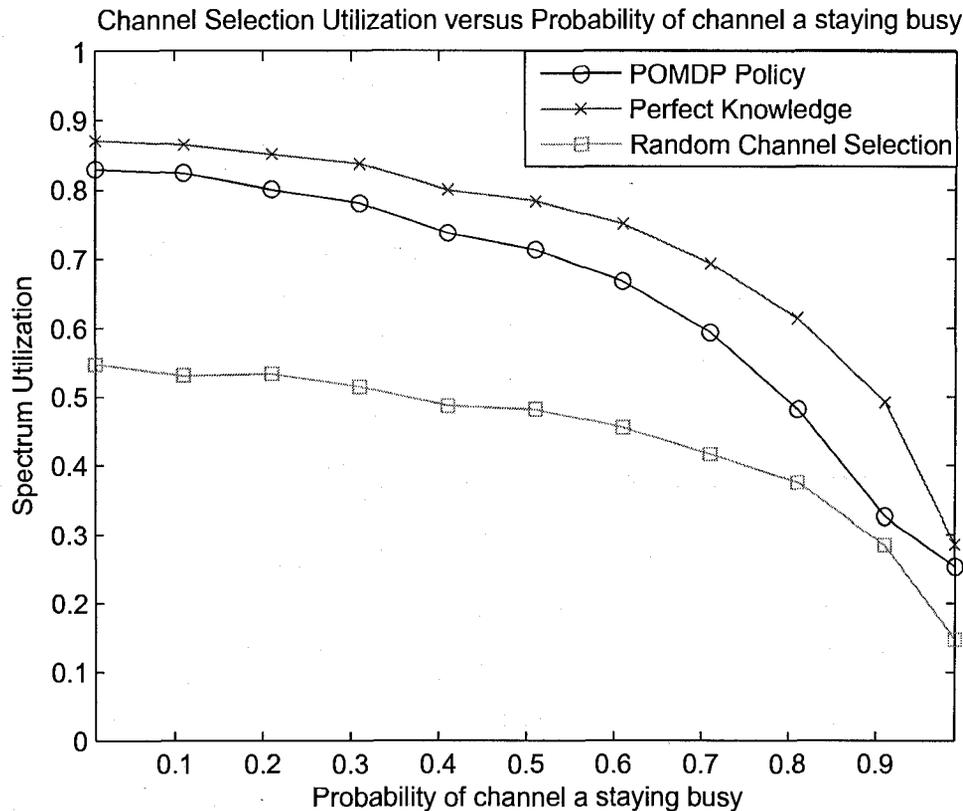


Fig. 5.6 Two channel spectrum occupancy and time varying channel gain sample scenario.



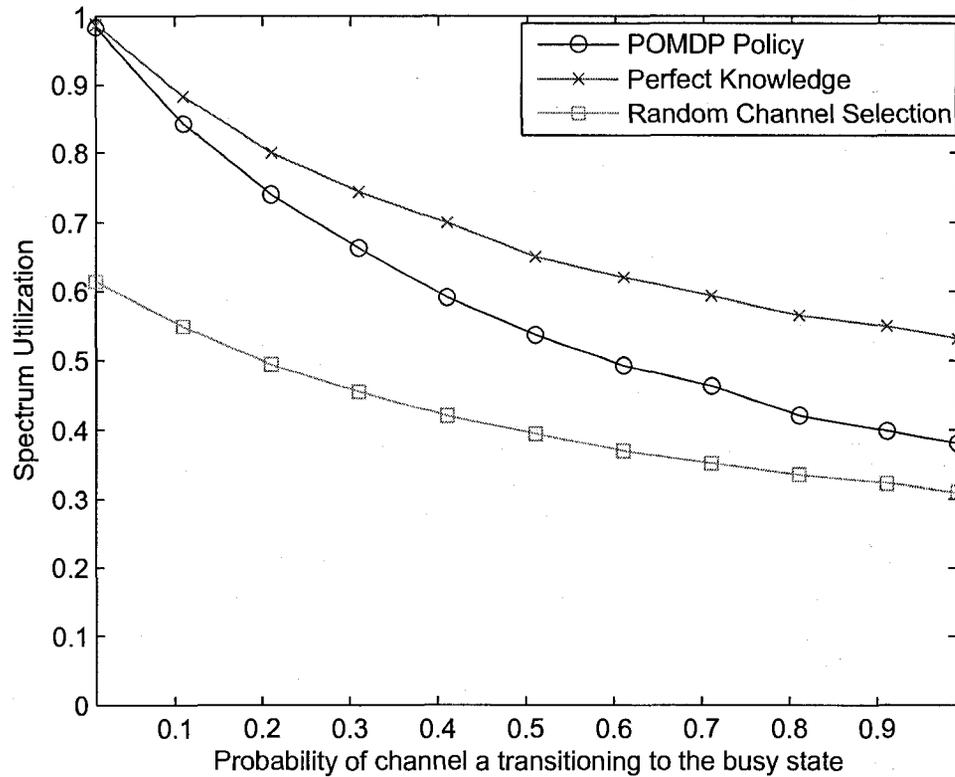
**Fig. 5.7** Two Channel Scenario: Spectrum Utilization vs. channel 1 state transition matrix parameters (Case I: The probability of staying in the busy state).

We evaluate SU and average distortion performance for three cases (1) POMDP channel selection, which is our proposed scheme, (2) randomly selecting channel 1 or 2 and using a constant  $\beta = 0.1$ , which represents a non-adaptive scheme, and (3) using perfect knowledge of the system state, which represents the ideal case.

### 5.3.1 The Affect of Transition Matrix on Channel Selection Policy

SU performance with varying transition matrix parameters is shown in Fig. 5.7 and Fig. 5.8. In both plots we only vary the transition matrix parameters of channel 1. Both channels have equal observation matrix parameters  $\epsilon^1 = \epsilon^2 = 0.62$  and

Channel Selection Utilization versus Probability of channel a transitioning to the busy state

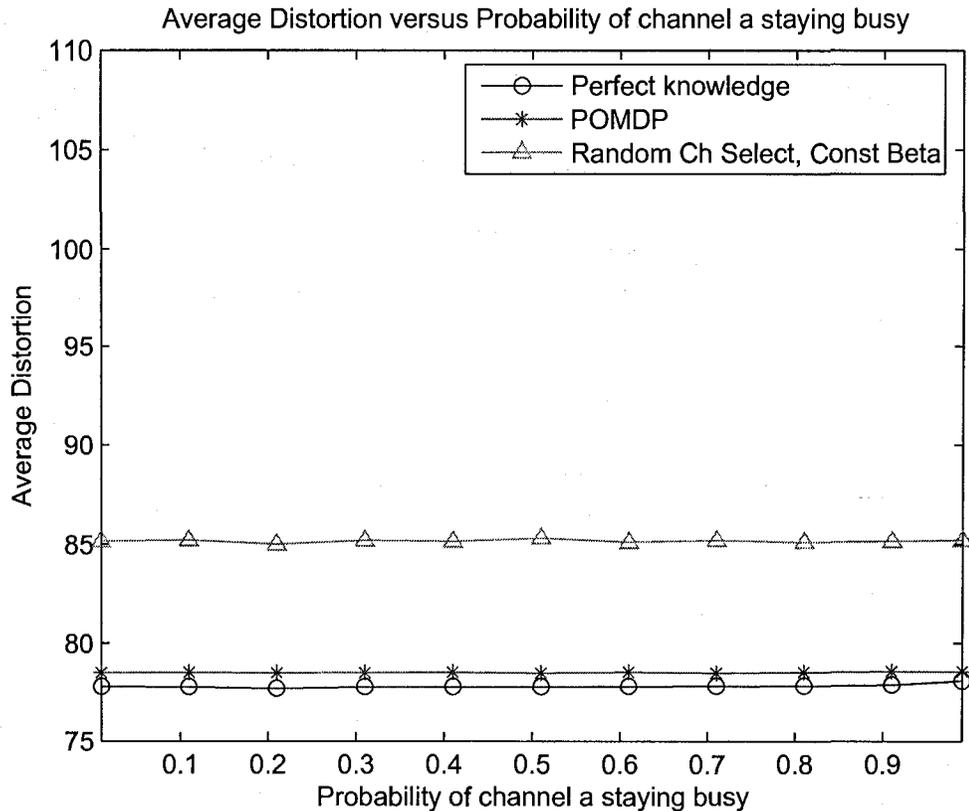


**Fig. 5.8** Two Channel Scenario: Spectrum Utilization vs. channel 1 state transition matrix parameters (Case II: The probability of transitioning to the busy state).

$$\sigma^1 = \sigma^2 = 0.1.$$

In Fig. 5.7 we vary the probability channel 1 stays busy,  $\Pr\{X_{k+1}^1 = z | X_k^1 = z\}$ .  $\Pr\{X_{k+1}^1 = z | X_k^1 = v\} = 0.2$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = z\} = 0.8$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = v\} = 0.6$ . In Fig. 5.8 we vary the probability channel 1 transitions to the busy state,  $\Pr\{X_{k+1}^1 = z | X_k^1 = v\}$ .  $\Pr\{X_{k+1}^1 = z | X_k^1 = z\} = 0.4$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = z\} = 0.8$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = v\} = 0.6$ .

In both cases, the SU utilization of our scheme is greater than the non-adaptive scheme. Our proposed scheme senses the surrounding environment to learn and adapt channel selection. However it takes several time slots for the policy to learn the system



**Fig. 5.9** Two Channel Scenario: Average distortion vs. channel 1 state transition matrix parameters (Case I: The probability of staying in the busy state).

state thus the performance of our scheme improves with slower transition dynamics. That is, our scheme approaches the perfect case as  $\Pr\{X_{k+1}^1 = z | X_k^1 = v\}$  approaches 0 as is shown in Fig. 5.8. Our scheme provides closer to optimal performance when there is a large difference in channel availability between the two channels as it becomes easier to distinguish the better channel. This is demonstrated in Fig. 5.7 where the performance of our scheme is more optimal at low  $\Pr\{X_{k+1}^1 = z | X_k^1 = z\}$  relative to  $\Pr\{X_{k+1}^2 = z | X_k^2 = z\}$ . In Fig. 5.9 and 5.10, we show the average distortion for the probability channel 1 stays busy and the probability channel 1 transitions to the busy state respectively. The average distortion of our scheme is better than the

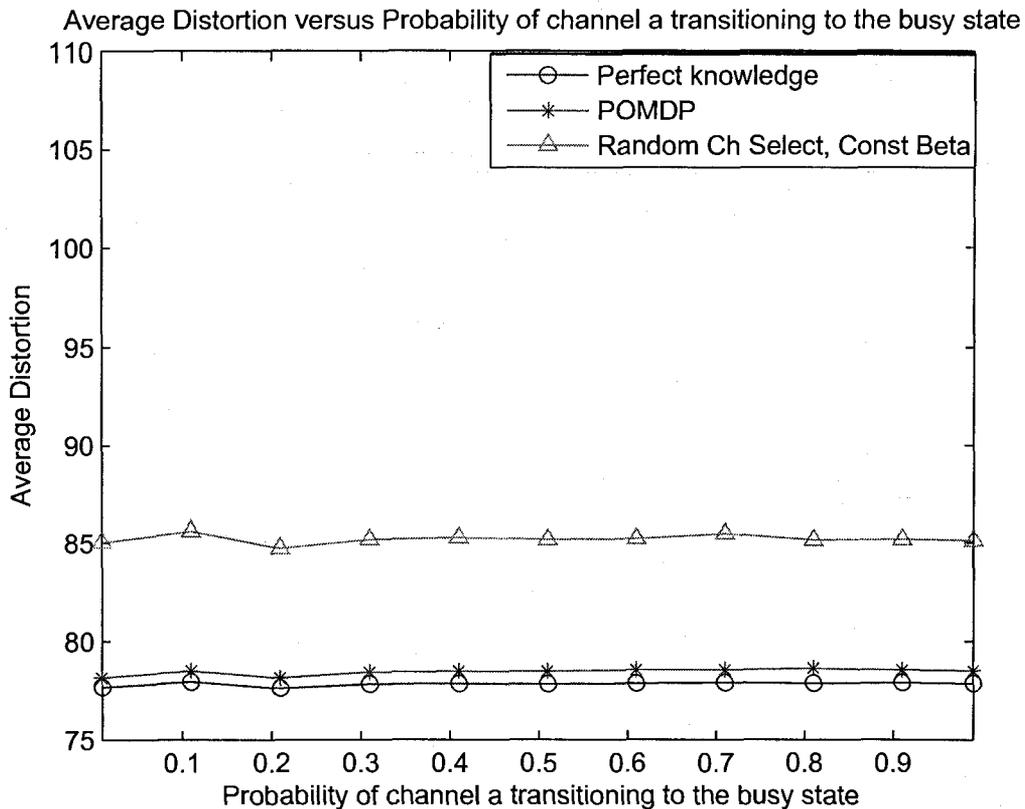


Fig. 5.10 Two Channel Scenario: Average distortion vs. channel 1 state transition matrix parameters (Case II: The probability of transitioning to the busy state).

non-adaptive scheme. Transition matrix parameters have little affect to the average distortion. Our scheme outperforms the non-adaptive scheme because our scheme will select the channel with the better channel gain and adapt the intra refreshing rate for the selected channel.

### 5.3.2 The Affect of Observation Matrix on Channel Selection Policy

SU with varying sensor operating point is shown in Fig. 5.11 and 5.12. In both cases,  $\Pr\{X_{k+1}^1 = z | X_k^1 = z\} = 0.4$ .  $\Pr\{X_{k+1}^1 = z | X_k^1 = v\} = 0.15$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = z\} = 0.6$ .  $\Pr\{X_{k+1}^2 = z | X_k^2 = v\} = 0.2$ . Observation parameters are derived

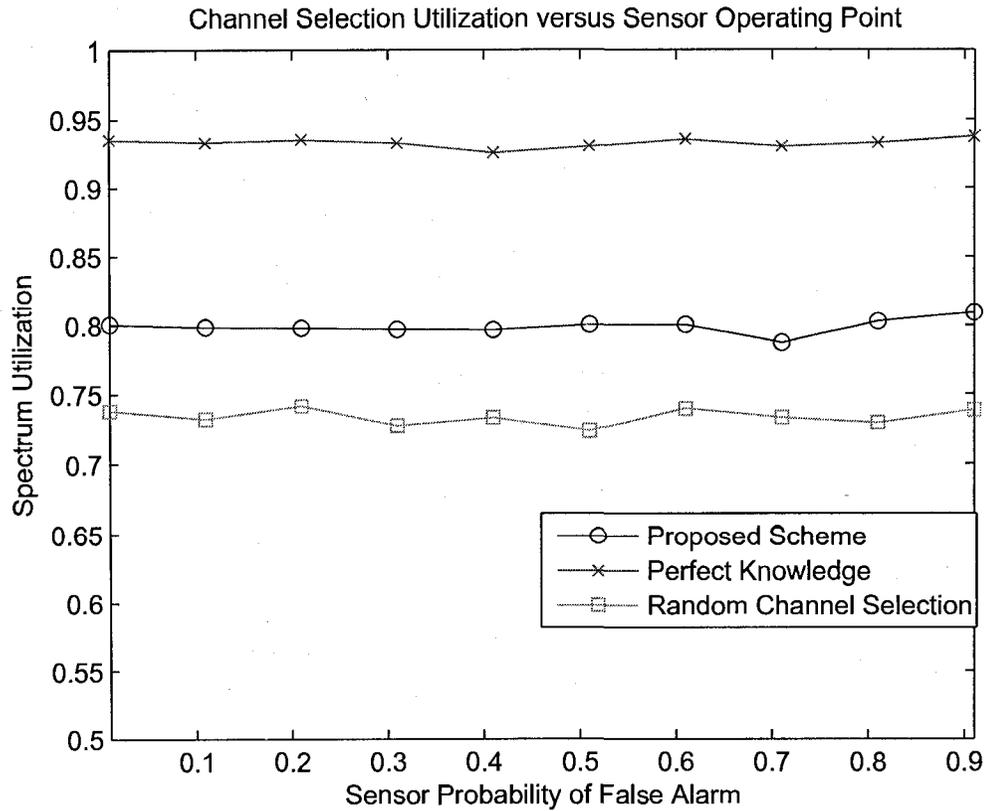


Fig. 5.11 Two Channel Scenario: Spectrum Utilization vs. channel 1 observation matrix parameters (Case I: Sensor operating point,  $\epsilon$ ).

by operating characteristics of the secondary users and are not likely to be different for each channel. Thus both channels are simulated with symmetrical observation parameters,  $\epsilon^1 = \epsilon^2 = \epsilon$  and  $\sigma^1 = \sigma^2 = \sigma$ . In Fig. 5.11 we vary the spectrum operating point  $\epsilon$ , and in Fig. 5.12 we vary the receiver channel estimation error  $\sigma$ .

In Fig. 5.13 and 5.14 we show the average distortion with varying  $\epsilon$  and  $\sigma$  respectively.

The observation parameters are shown to have little affect on the SU and average distortion performance of our proposed scheme.

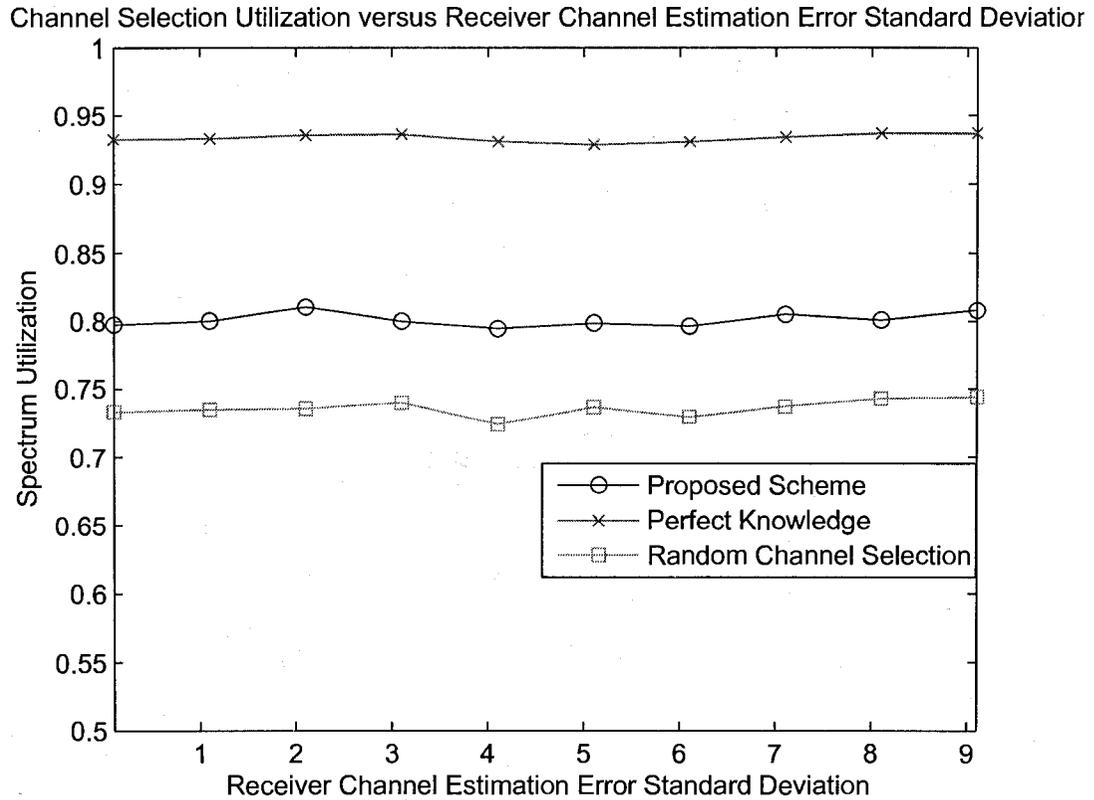


Fig. 5.12 Two Channel Scenario: Spectrum Utilization vs. channel 1 observation matrix parameters (Case II: Receiver channel estimation standard deviation,  $\sigma$ ).

## 5.4 Summary

In this chapter we presented numerical results of our proposed scheme. A single channel scenario is first presented to demonstrate the average distortion performance improvement by employing an adaptive intra-refresh strategy. When the channel is available, adapting the intra-refresh rate based on the channel gain shows a significant reduction in average distortion relative to non-adaptive schemes. The effect transition and observation matrix parameters have on average distortion was also presented in this chapter.

A two channel scenario was simulated to demonstrate the channel selection and

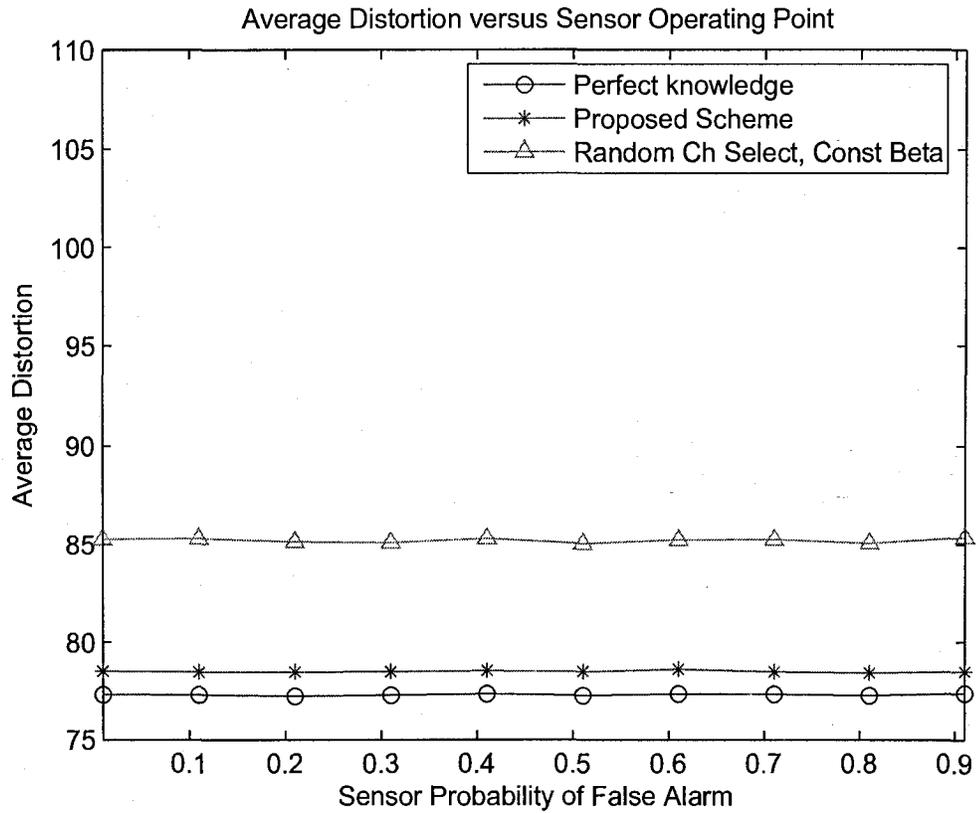


Fig. 5.13 Two Channel Scenario: Average distortion vs. channel 1 observation matrix parameters (Case I: Sensor operating point,  $\epsilon$ ).

adaptive intra-refresh rate scheme. Our channel selection scheme improves the spectrum utilization relative to a random channel selection scheme. The average distortion with our channel selection and intra-refresh rate scheme is lower relative to a non-adaptive random channel selection scheme.

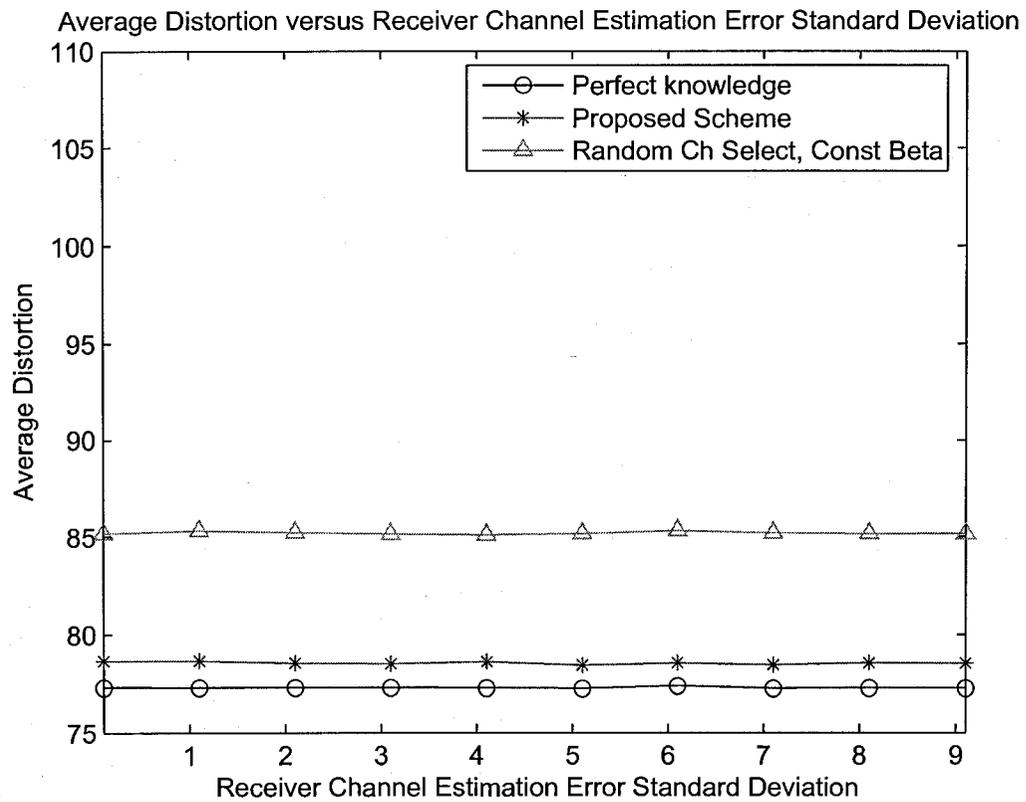


Fig. 5.14 Two Channel Scenario: Average distortion vs. channel 1 observation matrix parameters (Case II: Receiver channel estimation standard deviation,  $\sigma$ ).

# Chapter 6

## Conclusions and Future Work

### 6.1 Conclusions

In this thesis, a cross-layer multimedia transmission scheme was developed for cognitive radio. We investigated a hierarchical access spectrum overlay DSA CR system. An integrated CLD approach was used to adaptively select the intra-refresh rate to minimize distortion along with channel availability, channel gain, collision constraint, spectrum sensor design, and access strategy. We considered practical implementations by considering spectrum sensor error, CSI error, and hardware restrictions on the number of channels that can be sensed simultaneously. Our scheme is able to exploit both spectral and temporal opportunities.

To complete this study, we make the following concluding remarks.

- The problem of multimedia transmission over CR networks can be solved by employing an integrated CLD technique. An application layer QoS driven, channel adaptive scheme can significantly improve spectrum utilization and average distortion.
- The intra-refresh rate is an important application layer parameter to improve

error resilience when the channel is available and accessed. Due to the asymptotic nature of the channel distortion, the intra-refresh rate has little effect on distortion when the channel is unavailable or not accessed.

- With spectrum sensing and CSI error, the underlying system cannot be directly observed. The state of the system can be inferred from previous observations and actions. By formulating the entire system as a POMDP, the channel selection, access decision, spectrum sensor, and intra-refresh policies can be determined concurrently in the presence of sensing and CSI error.
- Solving a constrained POMDP requires randomized policies which is often intractable. It is also computationally prohibitive to solve unconstrained POMDPs with a large number of states. The number of states grows exponentially with the number of channels. By using a separation principle, myopic policies can be determined for the access and spectrum sensor policies. The information state can be used directly for the intra-refresh policy. The POMDP policy for channel selection can be computed offline. Employing these approaches significantly reduces the run-time computational complexity.
- With hardware limitations, not all channels can be sensed simultaneously. The channel selection policy allows the most likely available channel to be sensed which improves spectrum utilization relative to a random channel selection policy.
- The work presented in this thesis has limitations. The adaptive transmission scheme can only be used with peer to peer communication and does not support multicast scenarios as the channel conditions to each end point may be different. Solving POMDP problems becomes computationally prohibitive with

a large number of states. The number of states in our proposed scheme increases exponentially with the number of channels thus the computational complexity limits the number of channels that can be exploited. However researchers are investigating new methods to solve large scale POMDPs. The POMDP formulation requires a model of the transition and observation probabilities which may not be known. Reinforcement learning techniques can be employed to learn the models over time.

- The success of multimedia applications over CR networks will be limited in environments with high primary usage and/or a very low probability of collision required by the spectrum policy.

## 6.2 Future Research

CR and multimedia are two interesting and important fields of research. As such, they are both hot research areas. Future work can be done from several aspects.

- Future work could consider other QoS at the application layer.
- The work in this thesis can be readily extended to include joint source-channel coding, more channels, and channels with varying bit rate. More than one channel can be selected for channel sensing and channel correlation can be exploited to improve sensing while maximizing application layer QoS.
- Extending this work to include multiple secondary users using cooperative spectrum sharing techniques would be needed for a complete CR system.

## References

- [1] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, 2007.
- [2] FCC, "Report from the spectrum policy task force," <http://www.fcc.gov/sptf/reports.html>, Nov. 2002.
- [3] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, Royal Inst. Technol. (KTH), Stockholm, Sweden, 2000.
- [4] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Selected Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [5] DARPA XG WG, "The xg architectural framework v1.0," 2003.
- [6] —, "The xg vision rfc v2.0," 2004.
- [7] M. Van Der Schaar and N. Sai, Shankar, "Cross-layer wireless multimedia transmission: challenges, principles, and new paradigms," *IEEE Wireless Comm.*, vol. 12, no. 4, pp. 50–58, Aug. 2005.
- [8] S. Khan, Y. Peng, E. Steinbach, M. Sgroi, and W. Kellerer, "Application-driven cross-layer optimization for video streaming over wireless networks," *IEEE Commun. Mag.*, vol. 44, no. 1, pp. 122–130, Jan. 2006.

- [9] J. M. Chapin and W. H. Lehr, "The path to market success for dynamic spectrum access technology," *IEEE Commun. Mag.*, vol. 45, pp. 96–103, May 2007.
- [10] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 511–523, Jun. 2002.
- [11] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: A POMDP framework," *IEEE J. Selected Areas Commun.*, vol. 25, no. 3, pp. 589–600, Apr. 2007.
- [12] G. Ganesan and Y. Li, "Cooperative spectrum sensing in cognitive radio, part I: two user networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2204–2213, June 2007.
- [13] —, "Cooperative spectrum sensing in cognitive radio - part II: multiuser networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 6, pp. 2214–2222, June 2007.
- [14] A. R. Cassandra, "Exact and approximate algorithms for partially observed markov decision process," Ph.D. dissertation, Brown University, 1998.
- [15] J. Mitola, "The software radio architecture," *IEEE Commun. Mag.*, vol. 33, no. 5, pp. 26–38, May 1995.
- [16] D. N. Hatfield and P. J. Weiser, "Property rights in spectrum: taking the next step," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 43–55.
- [17] L. Xu, R. Tonjes, T. Paila, W. Hansmann, M. Frank, and M. Albrecht, "DRiVE-ing to the internet: Dynamic radio for IP services in vehicular environments," in *Proc. IEEE Conf. on Local Computer Networks*, Nov. 2000, pp. 281–289.

- [18] C. Raman, R. D. Yates, and N. B. Mandayam, "Scheduling variable rate links via a spectrum server," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 110–118.
- [19] O. Ileri, D. Samardzija, T. Sizer, and N. B. Mandayam, "Demand responsive pricing and competitive spectrum allocation via a spectrum server," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 194–202.
- [20] S. T. Chung, S. J. Kim, J. Lee, and J. M. Cioffi, "A game-theoretic approach to power allocation in frequency-selective gaussian interference channels," in *Proc. IEEE Int. Symp. on Information Theory*, Jun./Jul. 2003, pp. 316–316.
- [21] R. Etkin, A. Parekh, and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 517–528, Apr. 2007.
- [22] J. Huang, R. A. Berry, and M. L. Honig, "Spectrum sharing with distributed interference compensation," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 88–93.
- [23] Q. Zhao, "Spectrum opportunity and interference constraint in opportunistic spectrum access," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 3, Apr. 2007.
- [24] B. Wild and K. Ramchandran, "Detecting primary receivers for cognitive radio applications," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 124–130.

- [25] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. Asilomar Conf. on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 772–776.
- [26] W. A. Gardner, "Signal interception: a unifying theoretical framework for feature detection," *IEEE Trans. Commun.*, vol. 36, no. 8, pp. 897–906, Aug. 1988.
- [27] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits on cognitive radio," in *Proc. Allerton Conf. Communication, Control, Computing*, Oct. 2004.
- [28] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 131–136.
- [29] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in *Proc. IEEE Int. Conf. on Commun.*, vol. 4, Jun. 2006, pp. 1658–1663.
- [30] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. Wiley-Interscience, Sept. 2001.
- [31] Y. Chen, Q. Zhao, and A. Swami, "Joint design and separation principle for opportunistic spectrum access in the presence of sensing errors," *IEEE Trans. Inf. Theory*, Feb. 2007, submitted for publication. [Online]. Available: <http://arxiv.org/PS/cache/cs/pdf/0702/0702158.pdf>
- [32] Q. Zhao, L. Tong, and A. Swami, "Decentralized cognitive mac for dynamic spectrum access," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 224–232.

- [33] T. A. Weiss and F. K. Jondral, "Spectrum pooling: an innovative strategy for the enhancement of spectrum efficiency," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. 8–14, Mar. 2004.
- [34] U. Berthold and F. K. Jondral, "Guidelines for designing OFDM overlay systems," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 626–629.
- [35] H. Tang, "Some physical layer issues of wide-band cognitive radio systems," in *Proc. IEEE Int. Symp. on New Frontiers in Dynamic Spectrum Access Networks*, Nov. 2005, pp. 151–159.
- [36] S. Geirhofer, L. Tong, and B. M. Sadler, "Cognitive radios for dynamic spectrum access - dynamic spectrum access in the time domain: Modeling and exploiting white space," in *IEEE Commun. Mag.*, vol. 45, no. 5, May 2007, pp. 66–72.
- [37] P. Papadimitratos, S. Sankaranarayanan, and A. Mishra, "A bandwidth sharing approach to improve licensed spectrum utilization," *IEEE Commun. Mag.*, vol. 43, no. 12, Dec. 2005.
- [38] S. Geirhofer, L. Tong, and B. M. Sadler, "A measurement-based model for dynamic spectrum access in WLAN channels," in *Proc. Military Commun. Conf.*, Oct. 2006, pp. 1–7.
- [39] H. Zheng and C. Peng, "Collaboration and fairness in opportunistic spectrum access," in *Proc. IEEE Int. Conf. on Commun.*, vol. 5, May 2005, pp. 3132–3136.

- [40] W. Wang and X. Liu, "List-coloring based channel allocation for open-spectrum wireless networks," in *Proc. IEEE Veh. Technol. Conf.*, vol. 1, Sep. 2005, pp. 690–694.
- [41] Z. Ji and K. J. R. Liu, "Cognitive radios for dynamic spectrum access - dynamic spectrum sharing: A game theoretical overview," *IEEE Commun. Magazine*, vol. 45, no. 5, pp. 88–94, May 2007.
- [42] W. Hu, D. Willkomm, G. V. Lantis, M. Gerla, and A. Wolisz, "Dynamic frequency hopping communities for efficient IEEE 802.22 operation," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 80–87, May 2007.
- [43] H.-J. Lee, T. Chiang, and Y.-Q. Zhang, "Scalable rate control for MPEG-4 video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 878–894, Sep. 2000.
- [44] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 172–185, Feb. 1999.
- [45] Z. He, Y. K. Kim, and S. K. Mitra, "Object-level bit allocation and scalable rate control for MPEG-4 video coding," in *Proc. of Workshop and Exhibition on MPEG-4*, Jun. 2001, pp. 63–66.
- [46] K. Stuhlmüller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.
- [47] T. Sikora, "The mpeg-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 19–31, Feb 1997.

- [48] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.
- [49] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 952–965, Jun. 2000.
- [50] J. Y. Liao and J. Villasenor, "Adaptive intra block update for robust transmission of H.263," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 30–35, Feb. 2000.
- [51] A. C. Begen and Y. Altunbasak, "An adaptive media-aware retransmission timeout estimation method for low-delay packet video," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 332–347, Feb. 2007.
- [52] J. Chakareski and P. A. Chou, "Application layer error-correction coding for rate-distortion optimized streaming to wireless clients," *IEEE Trans. Commun.*, vol. 52, no. 10, pp. 1675–1687, Oct. 2004.
- [53] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [54] J. Costello, D. J., J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2531–2560, Oct. 1998.

- [55] P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 3, Mar. 1992, pp. 545–548.
- [56] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Indust. Appl. Math.*, June 1960.
- [57] Y. J. Liang, J. G. Apostolopoulos, and B. Girod, "Analysis of packet loss for compressed video: does burst-length matter?" in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 5, 2003, pp. 684–7.
- [58] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proc. IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.
- [59] K. Ngan and R. Steele, "Enhancement of PCM and DPCM images corrupted by transmission errors," *IEEE Trans. Commun.*, vol. 30, pp. 257–265, Jan. 1982.
- [60] K. M. Rose and A. Heiman, "Enhancement of one-dimensional variable-length DPCM images corrupted by transmission errors," *IEEE Trans. Commun.*, vol. 37, no. 4, pp. 373–379, Apr. 1989.
- [61] O. Mitchell and A. Tabatabai, "Channel error recovery for transform image coding," *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 1754–1762, Dec. 1981.
- [62] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 3, pp. 238–247, Jun. 1993.
- [63] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, no. 10, pp. 1544–1551, Oct. 1993.

- [64] Q. F. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 3, pp. 248–258, Jun. 1993.
- [65] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. Consum. Electron.*, vol. 39, no. 3, pp. 455–460, Aug. 1993.
- [66] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 470–477, Apr. 1995.
- [67] S. S. Hemami and T. H. Y. Meng, "Transform coded image reconstruction exploiting interblock correlation," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 1023–1027, Jul. 1995.
- [68] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *Proc. IEEE Int. Conf. on Commun.*, vol. 3, Jun. 1995, pp. 1778–1783.
- [69] H. Sun, K. Challapali, and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consum. Electron.*, vol. 38, no. 3, pp. 108–118, Aug. 1992.
- [70] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [71] Z. Wang, A. C. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002.

- [72] D. C. Lin and P. M. Chau, "Objective human visual system based video quality assessment metric for low bit-rate video communication systems," in *IEEE 8th Workshop on Multimedia Signal Processing*, Oct. 2006, pp. 320–323.
- [73] J. Guo, M. Van Dyke-Lewis, and H. R. Myler, "Gabor difference analysis of digital video quality," *IEEE Trans. Broadcast.*, vol. 50, no. 3, pp. 302–311, Sep. 2004.
- [74] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. Broadcast.*, vol. 50, no. 3, pp. 312–322, Sep. 2004.
- [75] J. Li, G. Chen, and Z. Chi, "A fuzzy image metric with application to fractal coding," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 636–643, Jun. 2002.
- [76] B. Girod and N. Farber, *Wireless video*. New York: Marcel Dekker, Inc., 2000, in Compressed Video Over Networks.
- [77] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *IEEE Wireless Commun.*, vol. 12, no. 1, pp. 3–11, Feb. 2005.
- [78] Z. Ji and K. J. R. Liu, "Dynamic spectrum sharing: a game theoretical overview," *IEEE Commun. Mag.*, vol. 45, no. 5, pp. 88–94, May 2007.
- [79] D. Zhang and Z. Tian, "Adaptive games for agile spectrum access based on extended kalman filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 1, pp. 79–90, Jun. 2007.
- [80] D. Wu, S. Ci, and H. Wang, "Cross-layer optimization for video summary transmission over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 4, pp. 841–850, May 2007.

- [81] M. Van Der Schaar, D. S. Turaga, and R. Wong, "Classification-based system for cross-layer optimized wireless video transmission," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1082–1095, Oct. 2006.
- [82] Y. S. Chan, J. W. Modestino, Q. Qu, and X. Fan, "An end-to-end embedded approach for multicast/broadcast of scalable video over multiuser CDMA wireless networks," *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 655–667, Apr. 2007.
- [83] C. H. Foh, Y. Zhang, Z. Ni, J. Cai, and K. N. Ngan, "Optimized cross-layer design for scalable video transmission over the IEEE 802.11e networks," *IEEE Trans. Circuits Syst. Video Technol.*, 2007, accepted for future publication.
- [84] A. Cassandra, "Tony's POMDP page,"  
<http://www.cs.brown.edu/research/ai/pomdp/index.html>.
- [85] R. Smallwood and E. Sondik, "Optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, pp. 1071–1088, 1973.
- [86] E. J. Sondik, "The optimal control of partially observable markov processes," Ph.D. dissertation, Stanford University, Stanford, California, 1971.
- [87] G. Monahan, "A survey of partially observable Markov decision processes: theory, models and algorithms," *Management Science*, vol. 28, no. 1, January 1982.
- [88] H. Cheng, "Algorithms for partially observable markov decision processes," Ph.D. dissertation, University of British Columbia, British Columbia, Canada, 1988.
- [89] N. L. Zhang and W. Liu, "Planning in stochastic domains: Problem characteristics and approximation," 1996, technical Report HKUSTCS96-31.

- 
- [90] H. Tong and T. Brown, "Adaptive call admission control under quality of service constraints: a reinforcement learning solution," *IEEE J. Selected Areas Commun.*, vol. 18, no. 2, pp. 209–221, 2000.
- [91] P. Bergamo, D. Maniezzo, A. Giovanardi, G. Mazini, and M. Zorzi, "Improved markov model for rayleigh fading envelope," *Electronics Letters*, vol. 38, pp. 477–478, May 2002.
- [92] A. T. Hoang and M. Motani, "Buffer and channel adaptive transmission over fading channels with imperfect channel state information," in *Proc. IEEE WCNC'04*, vol. 3, Mar. 2004.

# Appendix A: MATLAB Programs

## A.1 Single Channel Scenario

### A.1.1 Transition and Observation Matrix Generation

```

%Matrix Generation: generate T, quantized p,
%beta_opt, O, and D_matrix matrices

disp('-----')
disp('Generating Matrix Data')
%Build quantized p
%The length of p is NUM_STATES-1
p_tmp = zeros(1,NUM_STATES-1);
for p_index=1:NUM_STATES-1
    p_tmp(1,p_index) = ((p_index-1)/(NUM_STATES-2))*P_BOUND;
end
%set lowest p to nonzero value otherwise gamma goes to inf
p_tmp(1,1) = P_MIN;
p_len = length(p_tmp);
p = zeros(1,p_len);
p_index1 = NUM_STATES-1;
p_index2 = 1;
%reverse p_tmp and store in p
while (p_index1 >= 1)
    p(:,p_index2) = p_tmp(:,p_index1);
    p_index1 = p_index1 - 1;
    p_index2 = p_index2 + 1;
end

%Generate the optimum beta for quantized p
beta_opt = zeros(1,NUM_STATES);
beta_tmp = 0.01:0.01:0.99;
Ds = Ds_0 + ...
    beta_tmp.*(1-lambda + ...
    lambda.*beta_tmp) ...

```

```

.* (Ds_1 - Ds_0);
term1 = a ./ (1 - b + b .* beta_tmp);

for p_index=1:NUM_STATES
    if (p_index < NUM_STATES)
        p_tmp = p(:, p_index);
    else
        p_tmp = P_MAX;
    end
    term2 = p_tmp / (1 - p_tmp);
    Dc = term1 .* term2 .* Fd;
    [X,Y] = min(Ds + Dc, [], 2);
    beta_opt(:,p_index) = beta_tmp(1,Y);
end

%Calculate observation matrix based on SNR and Pce

%p = 1 - (1-Pb)^1, determine Pb (Prob bit error)
for p_index = 1:length(p)
    Pb = 1-(1-p(:,p_index))^(1/PKT_LEN);
    %Determine SNR required for this Pb assuming
    %binary modulation. Pb = qfunc(sqrt(gamma))
    gamma(:,p_index) = (qfuncinv(Pb)).^2;
end

%Determine Pce(s,k) (CSI error). Rows are s, columns k
den = 2 * sqrt(2) * STD;
for k = 1:NUM_STATES-1
    for s = 1:NUM_STATES-1
        if (s == 1)
            Pce(s,k) = 0.5 * (1 + erf((gamma(:,s)...
                + gamma(:,s+1) - 2*gamma(:,k))/den));
        elseif (s == NUM_STATES-1)
            Pce(s,k) = 0.5 * (1 - erf((gamma(:,s-1)...
                + gamma(:,s) - 2*gamma(:,k))/den));
        else
            Pce(s,k) = 0.5 * erf((gamma(:,s) + ...
                gamma(:,s+1) - 2*gamma(:,k))/den) - ...
                0.5 * erf((gamma(:,s) + gamma(:,s-1) - ...
                2*gamma(:,k))/den);
        end
    end
end

```

```

end

%Calculate Observation Matrix Rows are s, columns k
for s = 1:NUM_STATES
    for k = 1:NUM_STATES
        if (s < NUM_STATES)
            if (k < NUM_STATES)
                O(s,k) = Pce(s,k) * (1-epsilon);
            else
                O(s,k) = epsilon;
            end
        else
            if (k < NUM_STATES)
                O(s,k) = 0;
            else
                O(s,k) = 1;
            end
        end
    end
end

end

end

%Calculate Reward Matrix Rows are Beta columns are state
for beta_index = 1:NUM_STATES
    for s = 1:NUM_STATES
        if (s < NUM_STATES)
            p_tmp = p(:, s);
        else
            p_tmp = P_MAX;
        end

        Ds = Ds_0 + ...
            beta_opt(:,beta_index).*(1-lambda + ...
            lambda.*beta_opt(:,beta_index)) ...
            .* (Ds_1 - Ds_0);
        term1 = a ./ (1 - b + b .* beta_opt(:,beta_index));
        term2 = p_tmp / (1 - p_tmp);
        Dc = term1 .* term2 .* Fd;
        D(beta_index, s) = Ds + Dc;
    end
end

end

%Calculate constant beta matrix

```

```

for s=1:NUM_STATES
    if (s < NUM_STATES)
        p_tmp = p(:, s);
    else
        p_tmp = P_MAX;
    end
    Ds = Ds_0 + ...
        CONST_BETA.*(1-lambda + ...
        lambda.*CONST_BETA) ...
        .* (Ds_1 - Ds_0);
    term1 = a ./ (1 - b + b .* CONST_BETA);
    term2 = p_tmp / (1 - p_tmp);
    Dc = term1 .* term2 .* Fd;
    D_CONSTANT(1, s) = Ds + Dc;
end

%Calculate Transition Matrix, totally arbitrary
for s = 1:NUM_STATES
    for k = 1:NUM_STATES
        if (s < NUM_STATES)
            if (s == k)
                T(s,k) = DWELL(1,DWELL_INDEX);
            elseif (k < NUM_STATES)
                T(s,k) = (1-DWELL(1,DWELL_INDEX)-PRI_USAGE) / ...
                    (NUM_STATES-2);
            else
                T(s,k) = PRI_USAGE;
            end
        else
            if (s == k)
                T(s,k) = PRI_STAY_USED;
            else
                T(s,k) = (1-PRI_STAY_USED) / (NUM_STATES-1);
            end
        end
    end
end
end
end

```

### A.1.2 State Vector Generation

%State vector generation. State is the zero-based state number

```
t_slots = 1:1:NUM_T_SLOTS;
```

```

state(:,1) = START_STATE;
for i=1:length(t_slots)
    x = rand(1);
    cur_state = state(:,i)+1;

    LEVEL_LB = zeros(1,NUM_STATES);
    LEVEL_UB = zeros(1,NUM_STATES);
    for i2 = 1:NUM_STATES-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+T(cur_state,i2);
    end
    LEVEL_UB(:,NUM_STATES) = 1;
    for i2 = 1:NUM_STATES-1
        LEVEL_UB(:,NUM_STATES-i2) = LEVEL_UB(:,NUM_STATES-i2+1)- ...
            T(cur_state,NUM_STATES-i2+1);
    end

    for i2 = 1:NUM_STATES
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            state(:,i+1) = i2-1;
        end
    end
end

%Directly use observation matrix to generate errored
%view of the system state at each time slot
%This is essentially a vector of acknowledgements. Note
%that ACKs happen at the end of the time slot, so you need to
%USE this information for decisions at the beginning of
%the next time slot. That is, error_state is time aligned
%with state vector but represents information AFTER transmission.
error_state = zeros(1,NUM_T_SLOTS);
error_state(1,1) = START_STATE;
for i=1:NUM_T_SLOTS
    %Calculate our belief of the state
    x = rand(1);
    cur_state = state(:,i)+1;
    LEVEL_LB = zeros(1,NUM_STATES);
    LEVEL_UB = zeros(1,NUM_STATES);
    for i2 = 1:NUM_STATES-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+O(cur_state,i2);
    end
    LEVEL_UB(:,NUM_STATES) = 1;

```

```

for i2 = 1:NUM_STATES-1
    LEVEL_UB(:,NUM_STATES-i2) = LEVEL_UB(:,NUM_STATES-i2+1)- ...
        O(cur_state,NUM_STATES-i2+1);
end

for i2 = 1:NUM_STATES
    if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
        error_state(:,i) = i2-1;
    end
end
end

%Our belief state is independent of action thus we
%actually calculate the belief vector based solely
%on T and O. The ACKs in error_state occurred at the
%end of the time slot. So belief vector represents
%the belief information at the end of the time slot.
%updated_belief is the information assuming another
%transition took place and should only be used starting from
%time slot 2. It represents information at the beginning of a slot
%Give the belief a head start by assuming it knows the
%starting system state.
init_belief = zeros(1,NUM_STATES);
init_belief(1,START_STATE+1) = 1;
belief_t = zeros(NUM_T_SLOTS, NUM_STATES);
belief_t(1,:) = init_belief;
tmp_belief = zeros(1,NUM_STATES);
for i=1:length(t_slots)
    for s=1:NUM_STATES
        if (i == 1)
            tmp_belief = init_belief;
        else
            tmp_belief = belief_t(i-1,:);
        end

        %k is s' in belief update formula
        sum_num = 0;
        for k=1:NUM_STATES
            sum_num = sum_num + ...
                tmp_belief(1,k)*T(k,s)*O(s,error_state(:,i)+1);
        end
        sum_den = 0;
    end
end

```

```

        for s2=1:NUM_STATES
            for k2=1:NUM_STATES
                sum_den = sum_den + ...
                    tmp_belief(1,k2)*T(k2,s2)* ...
                    O(s2,error_state(:,i)+1);
            end
        end
        belief_t(i,s) = sum_num / sum_den;
    end
end

update_belief = zeros(NUM_T_SLOTS,NUM_STATES);
update_belief(1,:) = init_belief;
for i = 2:NUM_T_SLOTS
    for s=1:NUM_STATES
        sum = 0;
        for k=1:NUM_STATES
            sum = sum + (belief_t(i-1,k) * T(k,s));
        end
        update_belief(i,s) = sum;
    end
end
end

```

## A.2 Two Channel Scenario

### A.2.1 Transition Matrix Generation

```

%Transition matrix generator
%Creates a TOTAL_STATES X TOTAL_STATES transition
%matrix for two uncorrelated channels. Simulates
%the two channels separately and then computes the matrix.

%Calculate Reference Transition Matrix, totally arbitrary
%This is the transition matrix for a single channel
CH1_REF_T = zeros(NUM_STATES, NUM_STATES);
for s = 1:NUM_STATES
    for k = 1:NUM_STATES
        if (s < NUM_STATES)
            if (s == k)
                CH1_REF_T(s,k) = (1-CH1_PRI_USAGE)*CH1_DWELL;
            elseif (k < NUM_STATES)
                x = (1-CH1_PRI_USAGE)*CH1_DWELL;
                CH1_REF_T(s,k) = (1-x-CH1_PRI_USAGE) / ...

```

```

        (NUM_STATES-2);
    else
        CH1_REF_T(s,k) = CH1_PRI_USAGE;
    end
else
    if (s == k)
        CH1_REF_T(s,k) = CH1_PRI_STAY_USED(1,DWELL_INDEX);
    else
        CH1_REF_T(s,k) = (1-CH1_PRI_STAY_USED(1,DWELL_INDEX)) ...
            / (NUM_STATES-1);
    end
end
end
if (CH1_REF_T(s,k) < 0)
    error('trans_gen: Invalid channel 1 ref T element');
end
end
end
CH2_REF_T = zeros(NUM_STATES, NUM_STATES);
for s = 1:NUM_STATES
    for k = 1:NUM_STATES
        if (s < NUM_STATES)
            if (s == k)
                CH2_REF_T(s,k) = (1-CH2_PRI_USAGE)*CH2_DWELL;
            elseif (k < NUM_STATES)
                x = (1-CH2_PRI_USAGE)*CH2_DWELL;
                CH2_REF_T(s,k) = (1-x-CH2_PRI_USAGE) / ...
                    (NUM_STATES-2);
            else
                CH2_REF_T(s,k) = CH2_PRI_USAGE;
            end
        else
            if (s == k)
                CH2_REF_T(s,k) = CH2_PRI_STAY_USED;
            else
                CH2_REF_T(s,k) = (1-CH2_PRI_STAY_USED) / ...
                    (NUM_STATES-1);
            end
        end
        if (CH2_REF_T(s,k) < 0)
            error('trans_gen: Invalid channel 2 ref T element');
        end
    end
end
end

```

```

end

t_slots = 1:1:TRANS_GEN_SLOTS;
%trans_state stores the individual channel transitions into a
%NUM_CHANNELS x TIME SLOTS matrix. Below, the two channels are
%modelled as transitioning independently (i.e. use their own rand
%generator.
trans_state = zeros(NUM_CHANNELS, TRANS_GEN_SLOTS);
trans_state(:,1) = START_STATE;
for i=1:length(t_slots)
    x = rand(1);
    cur_state = trans_state(1,i)+1;

    LEVEL_LB = zeros(1,NUM_STATES);
    LEVEL_UB = zeros(1,NUM_STATES);
    for i2 = 1:NUM_STATES-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+CH1_REF_T(cur_state,i2);
    end
    LEVEL_UB(:,NUM_STATES) = 1;
    for i2 = 1:NUM_STATES-1
        LEVEL_UB(:,NUM_STATES-i2) = LEVEL_UB(:,NUM_STATES-i2+1)-...
            CH1_REF_T(cur_state,NUM_STATES-i2+1);
    end

    for i2 = 1:NUM_STATES
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            trans_state(1,i+1) = i2-1;
        end
    end
end

for i=1:length(t_slots)
    x = rand(1);
    cur_state = trans_state(2,i)+1;

    LEVEL_LB = zeros(1,NUM_STATES);
    LEVEL_UB = zeros(1,NUM_STATES);
    for i2 = 1:NUM_STATES-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+CH2_REF_T(cur_state,i2);
    end
    LEVEL_UB(:,NUM_STATES) = 1;
    for i2 = 1:NUM_STATES-1
        LEVEL_UB(:,NUM_STATES-i2) = LEVEL_UB(:,NUM_STATES-i2+1)-...

```

```

        CH2_REF_T(cur_state,NUM_STATES-i2+1);
    end

    for i2 = 1:NUM_STATES
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            trans_state(2,i+1) = i2-1;
        end
    end
end

mapped_state = zeros(1,TRANS_GEN_SLOTS);
for i=1:TRANS_GEN_SLOTS
    %search for unique state
    mapped_state(1,i) = ...
        find( (map_table(:,1) == trans_state(1,i)) & ...
            (map_table(:,2) == trans_state(2,i)) );
end

%Calculate the transition matrix
T = zeros(TOTAL_STATES, TOTAL_STATES);
for cur_state=1:TOTAL_STATES
    for next_state=1:TOTAL_STATES
        count = 0;
        for t_slots=1:TRANS_GEN_SLOTS-1
            if ((mapped_state(1,t_slots) == cur_state) && ...
                (mapped_state(1,t_slots+1) == next_state))
                count = count + 1;
            end
        end
        N = find(mapped_state(1,1:TRANS_GEN_SLOTS-1) == cur_state);
        if (length(N) == 0)
            T(cur_state,next_state) = 0;
        else
            T(cur_state,next_state) = count / length(N);
        end
    end
end

%Fudge transition matrix to sum to 1
T = roundn(T,-5);
for i=1:TOTAL_STATES

```

```

sum = 0;
for k=1:TOTAL_STATES
    if (T(i,k) < 0)
        disp('WARNING: Transition Matrix Element is less than 0\n');
    end
    sum = sum + T(i,k);
end
if (sum < 1)
    fudge = 1 - sum;
    for k=1:TOTAL_STATES
        if ( (T(i,k) > 0) & ...
            ((1-T(i,k)) > fudge) )
            T(i,k) = T(i,k)+fudge;
            break;
        end
    end
elseif (sum > 1)
    fudge = sum - 1;

    for k=1:TOTAL_STATES
        if ( (T(i,k) > fudge) )
            T(i,k) = T(i,k)-fudge;
            break;
        end
    end
end
end
end

T_ARRAY(DWELL_INDEX, :, :) = T;

```

### A.2.2 POMDP Matrix Generation

%POMDP Matrix Generation: generate T, quantized p,  
 %beta\_opt, 0, and D\_matrix matrixes to be used with  
 %pomdp\_solve

```

disp('-----')
disp('Generating POMDP Matrix Data')
%select NUM_STATES for p except 0 and 1
%The length of p is NUM_STATES-1
p_tmp = 0:1/(NUM_STATES):1;
p_len = length(p_tmp);
p = zeros(1,p_len-2);

```

```

for p_index=2:p_len-1
    p(:,p_len-p_index) = p_tmp(:,p_index);
end

%Generate the optimum beta for quantized p
beta_opt = zeros(1,NUM_STATES);
beta_tmp = 0.01:0.01:0.99;
Ds = Ds_0 + ...
    beta_tmp.*(1-lambda + ...
    lambda.*beta_tmp) ...
    .* (Ds_1 - Ds_0);
term1 = a ./ (1 - b + b .* beta_tmp);

for p_index=1:NUM_STATES
    if (p_index < NUM_STATES)
        p_tmp = p(:, p_index);
    else
        p_tmp = P_MAX;
    end
    term2 = p_tmp / (1 - p_tmp);
    Dc = term1 .* term2 .* Fd;
    [X,Y] = min(Ds + Dc, [], 2);
    beta_opt(:,p_index) = beta_tmp(1,Y);
end

%Calculate observation matrix based on SNR and Pce

%p = 1 - (1-Pb)^1, determine Pb (Prob bit error)
for p_index = 1:length(p)
    Pb = 1-(1-p(:,p_index))^(1/PKT_LEN);
    %Determine SNR required for this Pb assuming
    %binary modulation. Pb = qfunc(sqrt(gamma))
    gamma(:,p_index) = (qfuncinv(Pb)).^2;
end

%Determine Pce(s,k) (CSI error). Rows are s, columns k
den = 2 * sqrt(2) * STD;
for k = 1:NUM_STATES-1
    for s = 1:NUM_STATES-1
        if (s == 1)
            Pce(s,k) = 0.5 * (1 + erf((gamma(:,s)...
                + gamma(:,s+1) - 2*gamma(:,k))/den));
        end
    end
end

```

```

elseif (s == NUM_STATES-1)
    Pce(s,k) = 0.5 * (1 - erf((gamma(:,s-1)...
        + gamma(:,s) - 2*gamma(:,k))/den));
else
    Pce(s,k) = 0.5 * erf((gamma(:,s) + gamma(:,s+1) - ...
        2*gamma(:,k))/den) - ...
        0.5 * erf((gamma(:,s) + gamma(:,s-1) - ...
        2*gamma(:,k))/den);
end
end
end

end

%Calculate Observation Matrix Rows are s, columns k
for s = 1:NUM_STATES
    for k = 1:NUM_STATES
        if (s < NUM_STATES)
            if (k < NUM_STATES)
                O(s,k) = Pce(s,k) * (1-epsilon);
            else
                O(s,k) = epsilon;
            end
        else
            if (k < NUM_STATES)
                O(s,k) = 0;
            else
                O(s,k) = 1;
            end
        end
    end
end

end

end

%Calculate composite observation matrix
%There are NUM_CHANNELS*NUM_STATES possible
%observations. There are NUM_CHANNELS possible
%actions. We really have a different observation
%matrix for each action. There are TOTAL_STATES possible
%however the obs matrix breaks down like this:
%for the channel not accessed, the prob of observation
%is 0. For the channel accessed, the observation
%matrix is 0.
%Cols are observations, 3 per state. Rows are states.
O_1 = zeros(TOTAL_STATES, NUM_CHANNELS*NUM_STATES);

```

```

O_2 = zeros(TOTAL_STATES, NUM_CHANNELS*NUM_STATES);
for s=1:TOTAL_STATES
    for k=1:NUM_STATES
        O_1(s,k) = O(map_table(s,1)+1,k);
    end
    for k=NUM_STATES+1:NUM_CHANNELS*NUM_STATES
        O_2(s,k) = O(map_table(s,2)+1,k-NUM_STATES);
    end
end
end

%Calculate Reward Matrix Rows are Beta columns are state
%We need a deterministic reward structure for the POMDP
%but we also need a reward matrix for the simulation.
%For the POMDP, we need an action dependent reward where
%we assume optimum beta decision will be made afterwards.

for beta_index = 1:NUM_STATES
    for s = 1:NUM_STATES
        if (s < NUM_STATES)
            p_tmp = p(:, s);
        else
            p_tmp = P_MAX;
        end

        Ds = Ds_0 + ...
            beta_opt(:,beta_index).*(1-lambda + ...
            lambda.*beta_opt(:,beta_index)) ...
            .* (Ds_1 - Ds_0);
        term1 = a ./ (1 - b + b .* beta_opt(:,beta_index));
        term2 = p_tmp / (1 - p_tmp);
        Dc = term1 .* term2 .* Fd;
        D(beta_index, s) = Ds + Dc;
    end
end

%Calculate the PODMP Distortion.
%Rows are each state, cols are actions
%optimal decisions are along the diagonal of D

```

```

for k = 1:NUM_CHANNELS
    for s = 1:TOTAL_STATES
        D_POMDP(s,k) = D(map_table(s,k)+1,map_table(s,k)+1);
    end
end

%Calculate constant beta matrix
for s=1:NUM_STATES
    if (s < NUM_STATES)
        p_tmp = p(:, s);
    else
        p_tmp = P_MAX;
    end
    Ds = Ds_0 + ...
        CONST_BETA.*(1-lambda + ...
        lambda.*CONST_BETA) ...
        .* (Ds_1 - Ds_0);
    term1 = a ./ (1 - b + b .* CONST_BETA);
    term2 = p_tmp / (1 - p_tmp);
    Dc = term1 .* term2 .* Fd;
    D_CONSTANT(1, s) = Ds + Dc;
end

```

### A.2.3 State Vector Generation

```

%State vector generation. State is the zero-based state number

t_slots = 1:1:NUM_T_SLOTS;
state(:,1) = START_STATE;
for i=1:length(t_slots)
    x = rand(1);
    cur_state = state(:,i)+1;

    LEVEL_LB = zeros(1,TOTAL_STATES);
    LEVEL_UB = zeros(1,TOTAL_STATES);
    for i2 = 1:TOTAL_STATES-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+T(cur_state,i2);
    end
    LEVEL_UB(:,TOTAL_STATES) = 1;
    for i2 = 1:TOTAL_STATES-1
        LEVEL_UB(:,TOTAL_STATES-i2) = LEVEL_UB(:,TOTAL_STATES-i2+1)-...
            T(cur_state,TOTAL_STATES-i2+1);
    end
end

```

```

    for i2 = 1:TOTAL_STATES
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            state(:,i+1) = i2-1;
        end
    end
end
end

```

#### A.2.4 System Simulation

```

%Simulate system. For a state vector, select a channel based
%on POMDP value function and belief. Select optimum beta
%based on belief. Get acknowledge based on observation matrix.
%update belief vector based on acknowledge.

```

```

%The ACKs in error_state occurred at the
%end of the time slot. So belief vector represents
%the belief information at the end of the time slot.
%updated_belief is the information assuming another
%transition took place and should only be used starting from
%time slot 2. It represents information at the beginning of a slot
%Give the belief a head start by assuming it knows the
%starting system state.

```

```

init_belief = zeros(1,TOTAL_STATES);
init_belief(1,START_STATE+1) = 1;
belief_t = zeros(NUM_T_SLOTS, TOTAL_STATES);
belief_t(1,:) = init_belief;
tmp_belief = zeros(1,TOTAL_STATES);

for i=1:length(t_slots)
    %Iterate the finite POMDP solution over and over
    load_value_vector;

    if (i == 1)
        tmp_belief = init_belief;
    else
        tmp_belief = belief_t(i-1,:);
    end

    %Calculate action based on belief
    [VALUE_ROWS, VALUE_COLS] = size(value);
    for col=1:VALUE_COLS

```

```

        sum = 0;
        for row=2:VALUE_ROWS
            sum = sum + (value(row,col) * tmp_belief(1,row-1));
        end
        dot_product(:,col) = sum;
    end
    [dummy, max_dot] = max(dot_product(1,1:VALUE_COLS));
    action(:,i) = value(1,max_dot);

    %For this action, determine optimum beta based on belief
    %Always use the maximum available state, thus we need to
    %manually calculate the ch_likely_state (i.e. this is essentially
    %the beta used.
    max_tmp = 0;
    for comp_state=1:TOTAL_STATES
        ch_state = map_table(comp_state,action(:,i)+1);
        if (ch_state < (NUM_STATES-1))
            if (tmp_belief(1,comp_state) > max_tmp)
                max_tmp = tmp_belief(1,comp_state);
                ch_likely_state = ch_state;
            end
        end
    end
    end

    %Actual channel state used to calculate the distortion
    ch_state = map_table(state(1,i)+1,action(:,i)+1);

    POMDP_D(:,i) = D(ch_likely_state+1, ch_state+1);

    %Get an ACK:
    get_ack;

    %Based on ACK and transition, update belief vector
    update_belief;

end

```

### A.2.5 Get Acknowledgement

```

%Generate ack based on the current state, observation matrix
%and action

x = rand(1);

```

```

cur_state = state(:,i)+1;
LEVEL_LB = zeros(1,TOTAL_STATES);
LEVEL_UB = zeros(1,TOTAL_STATES);
if (action(:,i) == 0)
    num_obs = NUM_CHANNELS*NUM_STATES;
    for i2 = 1:num_obs-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+O_1(cur_state,i2);
    end
    LEVEL_UB(:,num_obs) = 1;
    for i2 = 1:num_obs-1
        LEVEL_UB(:,num_obs-i2) = LEVEL_UB(:,num_obs-i2+1)- ...
            O_1(cur_state,num_obs-i2+1);
    end
    for i2 = 1:num_obs
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            ack(:,i) = i2-1;
        end
    end
end
if (action(:,i) == 1)
    num_obs = NUM_CHANNELS*NUM_STATES;
    for i2 = 1:num_obs-1
        LEVEL_LB(:,i2+1) = LEVEL_LB(:,i2)+O_2(cur_state,i2);
    end
    LEVEL_UB(:,num_obs) = 1;
    for i2 = 1:num_obs-1
        LEVEL_UB(:,num_obs-i2) = LEVEL_UB(:,num_obs-i2+1)- ...
            O_2(cur_state,num_obs-i2+1);
    end
    for i2 = 1:num_obs
        if ( (x >= LEVEL_LB(:,i2)) && (x < LEVEL_UB(:,i2)) )
            ack(:,i) = i2-1;
        end
    end
end
end

```

### A.2.6 Update Information State

2

```

%Update per channel belief vector and composite belief vector
%This is done at the end of time slot i, thus the result is
%stored in i+1.

```

```

if (action(:,i) == 0)
    for s=1:TOTAL_STATES
        %k is s' in belief update formula
        sum_num = 0;
        for k=1:TOTAL_STATES
            sum_num = sum_num + ...
                tmp_belief(1,k)*T(k,s)*O_1(s,ack(:,i)+1);
        end
        sum_den = 0;
        for s2=1:TOTAL_STATES
            for k2=1:TOTAL_STATES
                sum_den = sum_den + ...
                    tmp_belief(1,k2)*T(k2,s2)*O_1(s2,ack(:,i)+1);
            end
        end
        belief_t(i,s) = sum_num / sum_den;
    end
end

if (action(:,i) == 1)
    for s=1:TOTAL_STATES
        %k is s' in belief update formula
        sum_num = 0;
        for k=1:TOTAL_STATES
            sum_num = sum_num + ...
                tmp_belief(1,k)*T(k,s)*O_2(s,ack(:,i)+1);
        end
        sum_den = 0;
        for s2=1:TOTAL_STATES
            for k2=1:TOTAL_STATES
                sum_den = sum_den + ...
                    tmp_belief(1,k2)*T(k2,s2)*O_2(s2,ack(:,i)+1);
            end
        end
        belief_t(i,s) = sum_num / sum_den;
    end
end

%We know that the system is going to do a transition next.
%Update with that transition.
for s=1:TOTAL_STATES

```

```
sum = 0;
for k=1:TOTAL_STATES
    sum = sum + (belief_t(i,k) * T(k,s));
end
post_trans_t(i,s) = sum;
end

belief_t(i,:) = post_trans_t(i,:);
```

## Appendix B: POMDP Files

### B.1 Example POMDP File for Two Channel Scenario

```
#Matlab Generated POMDP file
discount: 0.95
values: reward
states: 9
actions: 2
observations: 6
```

```
T: *
```

```
0.112590 0.070880 0.297690 0.081810 0.048000 0.191580 0.050220 0.030580 0.116650
0.078000 0.112220 0.280340 0.053520 0.082370 0.193390 0.030440 0.047950 0.121770
0.046680 0.048880 0.384820 0.031800 0.033330 0.256600 0.019420 0.020760 0.157710
0.073610 0.049930 0.183010 0.116570 0.075990 0.307540 0.042970 0.032620 0.117760
0.046380 0.071440 0.196020 0.075820 0.116220 0.293330 0.031440 0.049150 0.120200
0.033130 0.033900 0.253840 0.046810 0.047910 0.385390 0.018430 0.021360 0.159230
0.093830 0.066210 0.239240 0.089360 0.052800 0.246950 0.055240 0.033710 0.122660
0.062150 0.087410 0.240480 0.063330 0.101010 0.244370 0.034580 0.049340 0.117330
0.040250 0.042260 0.318980 0.043130 0.038790 0.308700 0.021830 0.019960 0.166100
```

```
O: 0
```

```
0.380000 0.000000 0.620000 0.000000 0.000000 0.000000
0.380000 0.000000 0.620000 0.000000 0.000000 0.000000
0.380000 0.000000 0.620000 0.000000 0.000000 0.000000
0.000000 0.380000 0.620000 0.000000 0.000000 0.000000
0.000000 0.380000 0.620000 0.000000 0.000000 0.000000
0.000000 0.380000 0.620000 0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 0.000000 0.000000 0.000000
0.000000 0.000000 1.000000 0.000000 0.000000 0.000000
```

O: 1

0.000000 0.000000 0.000000 0.380000 0.000000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.380000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000  
0.000000 0.000000 0.000000 0.380000 0.000000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.380000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000  
0.000000 0.000000 0.000000 0.380000 0.000000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.380000 0.620000  
0.000000 0.000000 0.000000 0.000000 0.000000 1.000000

R: 0 : 0 : \* : \* -80.968767  
R: 0 : 1 : \* : \* -80.968767  
R: 0 : 2 : \* : \* -80.968767  
R: 0 : 3 : \* : \* -75.262443  
R: 0 : 4 : \* : \* -75.262443  
R: 0 : 5 : \* : \* -75.262443  
R: 0 : 6 : \* : \* -101131.898009  
R: 0 : 7 : \* : \* -101131.898009  
R: 0 : 8 : \* : \* -101131.898009

R: 1 : 0 : \* : \* -80.968767  
R: 1 : 1 : \* : \* -75.262443  
R: 1 : 2 : \* : \* -101131.898009  
R: 1 : 3 : \* : \* -80.968767  
R: 1 : 4 : \* : \* -75.262443  
R: 1 : 5 : \* : \* -101131.898009  
R: 1 : 6 : \* : \* -80.968767  
R: 1 : 7 : \* : \* -75.262443  
R: 1 : 8 : \* : \* -101131.898009