REAL-TIME SPECTRAL PREDICTION AND METACOGNITION
FOR SPECTRUM SHARING RADAR

A Dissertation in
Electrical Engineering
by
Jacob A. Kovarskiy

© 2021 Jacob A. Kovarskiy

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2021
The dissertation of Jacob A. Kovarskiy was reviewed and approved by the following:

Ram M. Narayanan  
Professor of Electrical Engineering  
Dissertation Advisor  
Chair of Committee

Timothy J. Kane  
Professor of Electrical Engineering

Julio V. Urbina  
Associate Professor of Electrical Engineering

Karl M. Reichard  
Associate Research Professor of Acoustics

Anthony F. Martone  
Special Member  
The Army Research Laboratory

Kultegin Aydin  
Professor of Electrical Engineering  
Head of the Electrical Engineering Department
Abstract

The growing demand for radio frequency (RF) spectrum access poses new challenges for next-generation radar systems. Recent Federal Communications Commission (FCC) policies permit wireless communication networks to share the spectrum with incumbent radar systems. To operate in a crowded electromagnetic environment, radars must coexist with other RF emitters while maintaining system performance. The concept of cognitive RF provides robust and innovative solutions to efficiently share and build awareness of the spectrum. Cognition is actualized by the perception-action cycle (PAC) which iteratively senses RF interference (RFI), learns RFI behavior over time, and adapts the radar’s frequency band of operation. New developments in software defined radio (SDR) technology have enabled complex cognitive systems to be realized on hardware in real-time.

This work 1) presents a cognitive spectrum sharing implementation based on spectral prediction, 2) compares this implementation against radars employing alternative cognitive strategies, and 3) introduces a metacognition architecture to optimize a radar’s cognitive strategy with respect to the environment. The spectral prediction approach enables the radar to learn a stochastic model describing RF activity. Using this model, the radar adapts waveform parameters in anticipation of changes in the spectrum. Spectral prediction is demonstrated in conjunction with pulsed linear frequency modulated chirp waveforms as well as notched noise waveforms for coexistence. Additionally, this predictive implementation is compared to reactive and reinforcement learning-based spectrum sharing strategies. Experiments demonstrate that these different cognitive strategies are well suited to particular RFI scenarios. This indicates a need for radars to intelligently adapt cognitive strategies in changing environments. The bio-inspired concept of metacognition provides a framework for cognitive radar to achieve this via self-monitoring and regulation of the PAC. Here, we describe an algorithm selection process aided by metacognition theory.
To demonstrate the efficacy of spectral prediction and metacognition for radar, real-time SDR implementations are evaluated. A comprehensive set of synthetic RFI, emulated long-term evolution (LTE) RFI, and real measured RFI scenarios are used to characterize performance. These experiments measure the impact of RFI on radar processing and assess the relative performance improvements due to spectrum sharing. In measuring performance, a metric to characterize target detection quality is proposed based on the Jensen-Shannon divergence. Overall, this work presents a state-of-the-art review for cognitive RF, describes the theoretical background for each approach, details a real-time implementation for both predictive and metacognitive frameworks, and evaluates the performance of these implementations in a variety of RFI scenarios.
# Table of Contents

List of Figures viii
List of Tables xi
List of Symbols xii
Nomenclature xxii
Acknowledgments xxiv

## Chapter 1
**Introduction**
- 1.1 Dissertation Overview ........................................... 2
- 1.2 Motivation .......................................................... 3
- 1.3 Dissertation Objectives ........................................... 5

## Chapter 2
**Cognitive RF and Spectrum Sharing** 7
- 2.1 Introduction to Cognition for RF Systems ..................... 7
- 2.2 Review of Spectrum Sharing Architectures .................... 10

## Chapter 3
**Pulse-Doppler Radar** 14
- 3.1 Introduction to Radar ............................................. 14
- 3.2 Radar Processing .................................................. 15
  - 3.2.1 Matched Filtering ........................................... 15
  - 3.2.2 Doppler Processing .......................................... 17
  - 3.2.3 Target Detection ............................................ 18
3.3 Interference Impacts on Radar Processing ........................................... 20

Chapter 4
Sense-Predict-Avoid ................................................................. 28
4.1 Spectrum Sensing .............................................................. 29
4.2 Modeling RF Activity ......................................................... 30
  4.2.1 Parametric Model .................................................... 31
  4.2.2 Empirical Model .................................................... 33
4.3 Stochastic Model-based Prediction ........................................ 34
4.4 Software Defined Radar (SDRadar) Hardware ......................... 38

Chapter 5
Experimental Analysis of Sense-Predict-Avoid .............................. 42
5.1 Performance Metrics .......................................................... 42
5.2 Real-time Interference Scenarios .......................................... 46
  5.2.1 Synthetic Switching Tones ........................................ 47
  5.2.2 Synthetic Swept Tone & Random Frequency Hopper .......... 48
  5.2.3 Real Interference .................................................... 49
5.3 Real-time Experiments ........................................................ 51
  5.3.1 Synthetic RFI ......................................................... 51
    5.3.1.1 Random Single-Tone ....................................... 51
    5.3.1.2 Random Two-Tone ......................................... 54
  5.3.2 Synthetic Swept Tone & Random Frequency Hopper .......... 55
  5.3.3 Real Interference .................................................... 57

Chapter 6
Alternative Cognitive Radar Strategies ........................................ 59
6.1 Reinforcement Learning for Spectrum Sharing ......................... 60
6.2 Comparing to Sense-Learn-Avoid ......................................... 63
6.3 Notched FM Noise Waveforms ............................................ 68
6.4 Analysis of Sense-Predict-Notch .......................................... 71

Chapter 7
Metacognitive Radar ............................................................... 78
7.1 Introduction to Metacognition .............................................. 79
7.2 Metacognitive Radar Architecture ....................................... 81
7.3 Case-based Reasoner ........................................................ 85
7.4 The Restless Bandit Problem .............................................. 89
  7.4.1 Periodic Explore-first ............................................ 90
  7.4.2 Sliding Window UCB1-tuned ..................................... 92
7.4.3 Discounted Thompson Sampling ........................................ 94
7.4.4 REXP4 ................................................................. 96
7.5 Metacognitive Performance Self-evaluation ................................. 99
  7.5.1 Spectrum Sharing Performance ....................................... 99
  7.5.2 Jensen-Shannon Divergence .......................................... 101

Chapter 8
Experimental Analysis of Metacognition .................................. 105
  8.1 RFI Test Scenarios ...................................................... 105
  8.2 Comparing Restless Bandits .......................................... 111
    8.2.1 Spectrum Sharing Feedback .................................... 112
    8.2.2 Jensen-Shannon Divergence Feedback .......................... 115
  8.3 Classifier-aided Periodic Explore-first .............................. 118

Chapter 9
Conclusions and Future Work ................................................. 123
  9.1 Conclusions and Summary ............................................. 123
  9.2 Future Work ............................................................ 126

Appendix
Performance Metric Simulation Parameters ................................ 128

Bibliography ................................................................. 130
List of Figures

2.1 Visual representation of the perception-action cycle. ............... 8
3.1 Chirp Example .......................................................... 16
3.2 CA-CFAR Diagram ....................................................... 19
3.3 Chirp + RFI Spectrum .................................................... 21
3.4 Range Profile with RFI .................................................... 22
3.5 Range-Doppler Map with No RFI ......................................... 23
3.6 Range-Doppler Map with Constant Tone ................................. 24
3.7 Range-Doppler Map with Swept Tone ..................................... 25
3.8 Range-Doppler Map with Random Hopper ............................... 25
3.9 Range-Doppler Map with No RFI and Intra-CPI Adaptation .......... 26
4.1 Two-tone spectrum sensing over time example .......................... 31
4.2 Two-tone spectrum sensing with stochastic model parameter estimation 32
4.3 SPA High Level Block Diagram ............................................ 35
4.4 Picture of USRP-2944R .................................................... 39
4.5 Processing and data flow in SDRadar hardware ............................ 40
5.1 Test setup with processing flow in SDRadar hardware ..................... 46
5.2 Synthetic switching tone RFI examples .................................. 48
5.3 Synthetic swept and random hopping tone RFI examples .................. 49
5.4 Real measured RFI examples at 1750 MHz and 800 MHz center frequencies .................................................. 49
5.5 Real measured RFI examples at 1850 MHz and 2450 MHz center frequencies .................................................. 50
5.6 Real measured RFI examples at 1950 MHz and 2350 MHz center frequencies .................................................. 50
5.7 Synthetic single tone RFI results with SPA in log-normal CDF mode 52
5.8 Synthetic single tone RFI results with SPA in CHF mode .................. 52
5.9 Synthetic two tone RFI results with SPA .................................. 54
5.10 Synthetic swept tone and random hopping RFI results with SPA .... 56
8.15 Number of times best strategy selection for JS divergence feedback . 120
8.16 Spectrum sharing performance decomposed into individual metrics
   with notching . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
List of Tables

5.1 Coefficient of variation for each signal in the two-tone RFI scenario. The first column specifies the frequency of each sinusoid. The −15-MHz and 25-MHz signal have fixed on-times of 4.1 ms and 2.05 ms, respectively. "C1" through "C5" refers each case number. 55

5.2 The dwell time for each RFI scenario presented in Figure 5.10 is shown. "S1" through "S3" refers to swept tone cases while "H1" through "H3" refers to a random frequency hopper cases. 56

6.1 Coefficient of variation for each signal in the two-tone RFI scenario. The first column specifies the frequency of each sinusoid. The −15-MHz and 25-MHz signal have fixed on-times of 4.1 ms and 2.05 ms, respectively. "C1" through "C5" refers each case number. 64

6.2 Swept tone bandwidth improvement over notchless avoidance. 74

6.3 Two-tone case definition and bandwidth improvement over notchless avoidance. 76

A1 Simulation pulse pattern parameters 129
List of Symbols

Section 3.2.1

$t$  continuous time samples
$A$  amplitude of transmitted waveform
$u(t)$  LFM chirp waveform (time domain)
$f_s$  start frequency of LFM chirp
$f_e$  end frequency of LFM chirp
$\tau$  pulse width of LFM chirp
$R$  target range
$c$  wave velocity (speed of light)
$\Delta t$  time delay of reflected radar pulse
$v(t)$  range profile over time
$\mathcal{F}$  Fourier transform operator
$n$  discrete frequency samples
$Y[n]$  received radar signal (frequency domain)
$U[n]$  LFM chirp waveform (frequency domain)
$R_{\text{res}}$  radar range resolution
$B$  bandwidth of LFM chirp

Section 3.2.2
$t$  discrete time index

$T$  number of range cells in range profile

$M$  number of Doppler cells in range-Doppler map

$m$  radar pulse number (slow time)

$V$  matrix of range profiles

$F_m$  Fourier transform across the $m$-dimension

$D$  range-Doppler map matrix

d[$t, m$]  range-Doppler cell

$f_D$  Doppler frequency

$F_{PRI}$  pulse repetition frequency

$s_r$  target velocity

$\lambda$  radar carrier wavelength

**Section 3.2.3**

$D_R$  target component of range-Doppler matrix

$D_N$  noise component of range-Doppler matrix

$\Lambda$  CA-CFAR threshold

$z[t, m]$  range-Doppler cell after detection

$\alpha_{CA}$  CA-CFAR threshold constant

$\sigma_n^2$  local estimate of background noise

$N_{CA}$  number of background cells for noise estimation

$P_{FA}$  probability of false alarm

**Section 3.3**

$P_r$  radar receive power

$P_t$  radar transmit power

$G_r$  radar receive antenna gain

$G_t$  radar transmit antenna gain

$\sigma_r$  radar cross-section

$L$  radar range equation loss multiplier
\[X[n]\] RFI signal (frequency domain)
\[N_f[n]\] receiver noise (frequency domain)
\[v_R[t]\] target component of range profile
\[v_I[t]\] interference component of range profile
\[v_N[t]\] noise component of range profile
\[D_I\] interference component of range-Doppler matrix

**Section 4.1**

\[Y_{SS}[n]\] received FFT for spectrum sensing
\[S_i\] occupancy state of the \(i^{th}\) sub-band
\[N_{s_i}\] start index of the \(i^{th}\) sub-band
\[N_{e_i}\] end index of the \(i^{th}\) sub-band
\[\lambda_D\] energy detection threshold for spectrum sensing
\[N_d\] number of indices in each sub-band
\[M\] number of sub-bands
\[P_{FA}\] spectrum sensing false alarm rate
\[\sigma_W^2\] receiver average noise power

**Section 4.2**

\[p_{B_i}(t_{B_i})\] probability the \(i^{th}\) sub-band remains in a busy state for \(t_{B_i}\) timesteps
\[p_{I_i}(t_{I_i})\] probability the \(i^{th}\) sub-band remains in an idle state for \(t_{I_i}\) timesteps
\[B_{i,j}\] \(j^{th}\) busy duration occurrence for the \(i^{th}\) sub-band
\[I_{i,j}\] \(j^{th}\) idle duration occurrence for the \(i^{th}\) sub-band
\[\mu_{B_i}\] mean busy duration for the \(i^{th}\) sub-band
\[\mu_{I_i}\] mean idle duration for the \(i^{th}\) sub-band
\[\sigma_{B_i}\] standard deviation of busy durations for the \(i^{th}\) sub-band
\( \sigma_{l_i} \) standard deviation of idle durations for the \( i^{th} \) sub-band

\( \mu_{L,B_i} \) busy distribution log-normal \( \mu \) parameter for the \( i^{th} \) sub-band

\( \sigma_{L,B_i} \) busy distribution log-normal \( \sigma \) parameter for the \( i^{th} \) sub-band

\( \mu_{L,I_i} \) idle distribution log-normal \( \mu \) parameter for the \( i^{th} \) sub-band

\( \sigma_{L,I_i} \) idle distribution log-normal \( \sigma \) parameter for the \( i^{th} \) sub-band

\( T_0 \) length of time in a timestep or FFT acquisition

\( \theta_B \) prediction threshold for busy states

\( \theta_I \) prediction threshold for idle states

\( A_i \) predicted availability for the \( i^{th} \) sub-band

\( A_{TX} \) set of selected transmit sub-bands based on avoidance or notching

\( C_s \) simulated collision rate during threshold optimization

\( D_s \) simulated missed opportunity rate during threshold optimization

\( \rho \) prediction error cost function for threshold optimization

\( \alpha \) spectrum sharing trade-off for threshold optimization

\( t_0 \) waveform adaptation latency

**Section 5.1**

\( C_r \) actual observed collision rate during radar operation

\( D_r \) actual observed missed opportunity rate during radar operation
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_{RX}$</td>
<td>actual available sub-bands in the spectrum during radar operation</td>
</tr>
<tr>
<td>$Q$</td>
<td>number of pulses in a test set for measuring SPA performance</td>
</tr>
<tr>
<td>$P_{IN}$</td>
<td>average interference plus noise power estimate</td>
</tr>
<tr>
<td>$n_{t1}$</td>
<td>start index of test target mainlobe</td>
</tr>
<tr>
<td>$n_{t2}$</td>
<td>end index of test target mainlobe</td>
</tr>
<tr>
<td>SINR</td>
<td>peak-to-average SINR</td>
</tr>
<tr>
<td>SINR$^F$</td>
<td>peak-to-average SINR from full-bandwidth transmission</td>
</tr>
<tr>
<td>SINR$^p$</td>
<td>peak-to-average SINR from waveform due to prediction</td>
</tr>
<tr>
<td>SINR$^{\min}$</td>
<td>minimum SINR required for target detection</td>
</tr>
<tr>
<td>$P^{\min}$</td>
<td>minimum power required for target detection</td>
</tr>
<tr>
<td>$R_D$</td>
<td>maximum detection range based on minimum SINR for detection</td>
</tr>
<tr>
<td>$R_{D,P}$</td>
<td>maximum detection range for predicted waveform</td>
</tr>
<tr>
<td>$R_{D,F}$</td>
<td>maximum detection range for full-bandwidth waveform</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>detectable range improvement metric</td>
</tr>
</tbody>
</table>

**Section 5.2**

$cv_{V,I}$ | coefficient of variation for off-times of test RFI

**Section 6.1**

$A$ | action space of RL agent |
$S$ | state space of RL agent |
$T$ | state transition probability function |
$R$ | reward function of RL agent |
$N$ | number of frequency sub-bands |
number of actions for a spectrum sharing radar
optimal policy in an environment
policy selected by radar
RL discount factor weight on immediate and future rewards
utility function depending on current state $s$

Section 6.3

iteration of PRO-FM optimization
total number of PRO-FM iterations
time domain PRO-FM waveform projection at $k^{th}$ iteration
frequency domain PRO-FM waveform projection at $k^{th}$ iteration
discretized spectral shape requirements for PRO-FM notching
discretized rectangular window
operator extracting phase from a complex value
Hadamard product or element-wise multiply operator
Gaussian spectral template with notches
vector of notch locations
lower notch taper location
upper notch taper location
lower notch taper frequency template
upper notch taper frequency template
vector of waveform phase values
number of waveform samples
zero-order hold period of SDR DAC
discrete frequency index of ZOROW signal representation
\( S(f_m, \phi) \)

ZOROW DAC signal reconstruction representation

**Section 7.3**

\( \Omega(Y_{SS}) \)

generalized classifier function with spectrogram \( Y_{SS} \) as input

\( \omega_c \)

class or case of RF spectrum

\( P \)

number of FFTs obtained for spectrogram

\( N \)

number of frequency samples in each FFT

\( M \)

number of sub-bands for spectrum sensing

\( C_G \)

average congestion of spectrogram

\( P_0 \)

number of FFT acquisitions with no RF activity

\( C_X \)

average complexity of spectrogram

**Section 7.4.1**

\( t \)

CPI number or MAB round

\( T \)

number of CPIs or rounds during metacognitive radar operation

\( a_k \)

\( k^{th} \) CR strategy

\( K \)

total number of CR strategies

\( z_k \)

performance feedback from \( k^{th} \) strategy

\( \bar{z}_k \)

average performance for \( k^{th} \) strategy

\( T_{SEI} \)

number of CPIs in a SEI

\( T_{exp} \)

number of CPIs during exploration for Periodic Explore-first

\( G(t) \)

history of performance feedback

\( H(t) \)

history of of \( k \)-values for selected CR strategies

\( 1_{\{arg\}} \)

indicator function that returns 1 if \( arg \) is true and 0 otherwise
$G^*(t)$  
maximum attainable performance if the best strategy is selected over time

$R(t)$  
cumulative regret over time

**Section 7.4.2**

$C$  
UCB1 constant controlling likelihood of exploration

$T_W$  
sliding window size in CPIs for SW UCB1-tuned

$u_k$  
confidence index for $k^{th}$

$\sigma_k^2$  
variance of performance feedback for $k^{th}$

$G(t)$  
history of performance feedback

**Section 7.4.3**

$\gamma$  
discounting or forgetting factor for DTS

$\theta_k$  
random sample from Beta distribution for $k^{th}$ strategy

$Beta(\alpha, \beta)$  
Beta distribution with parameters $\alpha$ and $\beta$

$\alpha_0$  
Beta prior initialization parameter for successes

$\beta_0$  
Beta prior initialization parameter for failures

$\bar{s}$  
set of successes for each strategy

$s_k$  
success for $k^{th}$ strategy

$\bar{f}$  
set of failure for each strategy

$f_k$  
failure for $k^{th}$ strategy

$Ber(p)$  
Bernoulli trial with likelihood $p$

**Section 7.4.4**

$p_k$  
probability distribution assignment for $k^{th}$ strategy
\( \xi_{m,k} \) \( m \)th expert’s recommendation score for the \( k \)th strategy

\( M \) number of expert recommendations or sub-bands

\( c_{X,m} \) complexity for the \( m \)th sub-band

\( w_{m} \) weight for the \( m \)th expert

\( \hat{z}_{k} \) \( k \)th pseudo-reward with IPS applied

\( \hat{y}_{m} \) pseudo-reward weighed with \( m \)th expert

\( U(0,1) \) uniform distribution on the interval \([0,1]\)

\( \theta \) sample from uniform distribution

\( r \) EXP4 constant (controls regret bound/adaptation rate)

\( T_{R} \) number of CPIs to run REXP4 before resetting

**Section 7.5.1**

\( c_{\text{est.},(i,m)} \) collisions in the \( i \)th sub-band for the \( m \)th pulse

\( Q \) number of pulses in a test set for measuring SPA performance

\( M \) number of sub-bands for spectrum sensing

\( I_{\text{est}} \) estimated spectral interference power

\( \eta_{t} \) estimated spectral SINR

\( \hat{\eta}_{1} \) normalized spectral SINR in a CPI

\( \hat{\eta}_{2} \) normalized bandwidth utilization in a CPI

\( \hat{\alpha} \) spectrum sharing performance feedback weighing

**Section 7.5.2**

\( P_{0} \) noise probability distribution noise (no target)

\( P_{1} \) target probability distribution noise (target)

\( D_{\text{KL}}(P_{0}||P_{1}) \) Kullback-Leibler divergence between \( P_{0} \) and \( P_{1} \)

\( D_{\text{JS}}(P_{0}||P_{1}) \) Jensen-Shannon divergence between \( P_{0} \) and \( P_{1} \)
# Nomenclature

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARL</td>
<td>U.S. Army Research Laboratory</td>
</tr>
<tr>
<td>ARP</td>
<td>Alternating Renewal Process</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>CA-CFAR</td>
<td>Cell-Averaging Constant False Alarm Rate</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CE</td>
<td>Cognitive Engine</td>
</tr>
<tr>
<td>CFAR</td>
<td>Constant False Alarm Rate</td>
</tr>
<tr>
<td>CHF</td>
<td>Cumulative Hazard Function</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-The-Shelf</td>
</tr>
<tr>
<td>CoV</td>
<td>Coefficient of Variation</td>
</tr>
<tr>
<td>CPI</td>
<td>Coherent Processing Interval</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CR</td>
<td>Cognitive Radar</td>
</tr>
<tr>
<td>CUT</td>
<td>Cell Under Test</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital-to-Analog Converter</td>
</tr>
<tr>
<td>DSA</td>
<td>Dynamic Spectrum Access</td>
</tr>
<tr>
<td>DTS</td>
<td>Discounted Thompson Sampling</td>
</tr>
<tr>
<td>EME</td>
<td>Electromagnetic Environment</td>
</tr>
<tr>
<td>EXP4</td>
<td>Exponential-weight algorithm for Exploration and</td>
</tr>
<tr>
<td></td>
<td>Exploitation using Expert advice</td>
</tr>
<tr>
<td>FCC</td>
<td>Federal Communications Commission</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>FM</td>
<td>Frequency Modulated</td>
</tr>
<tr>
<td>FPGA</td>
<td>Field-Programmable Gate Array</td>
</tr>
<tr>
<td>FSS</td>
<td>Fast Spectral Sensing</td>
</tr>
<tr>
<td>GPU</td>
<td>Graphics Processing Unit</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>IID</td>
<td>Independent and Identically Distributed</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IQ</td>
<td>In-phase and Quadrature</td>
</tr>
<tr>
<td>JS Divergence</td>
<td>Jensen-Shannon Divergence</td>
</tr>
<tr>
<td>KL Divergence</td>
<td>Kullback-Leibler Divergence</td>
</tr>
<tr>
<td>LFM</td>
<td>Linear Frequency Modulated</td>
</tr>
<tr>
<td>LTE</td>
<td>Long-Term Evolution</td>
</tr>
<tr>
<td>MAB</td>
<td>Multi-Armed Bandit</td>
</tr>
<tr>
<td>MCR</td>
<td>Metacognitive Radar</td>
</tr>
<tr>
<td>MDP</td>
<td>Markov Decision Process</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-Input and Multiple-Output</td>
</tr>
<tr>
<td>MSK</td>
<td>Meta-Strategic Knowledge</td>
</tr>
<tr>
<td>OFDM</td>
<td>Orthogonal Frequency-Divison Multiplexing</td>
</tr>
<tr>
<td>PAC</td>
<td>Perception-Action Cycle</td>
</tr>
<tr>
<td>PCIe</td>
<td>Peripheral Component Interconnect Express</td>
</tr>
<tr>
<td>POMDP</td>
<td>Partially Observable Markov Decision Process</td>
</tr>
<tr>
<td>PRF</td>
<td>Pulse Repetition Frequency</td>
</tr>
<tr>
<td>PRI</td>
<td>Pulse Repetition Interval</td>
</tr>
<tr>
<td>PRO-FM</td>
<td>Pseudo-Random Optimized FM</td>
</tr>
<tr>
<td>PSU</td>
<td>The Pennsylvania State University</td>
</tr>
<tr>
<td>RCS</td>
<td>Radar Cross-Section</td>
</tr>
<tr>
<td>REXP4</td>
<td>Restarting EXP4 algorithm</td>
</tr>
<tr>
<td>RF</td>
<td>Radio Frequency</td>
</tr>
<tr>
<td>RFI</td>
<td>Radio Frequency Interference</td>
</tr>
<tr>
<td>RNN</td>
<td>Recurrent Neural Network</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>SDR</td>
<td>Software Defined Radio</td>
</tr>
<tr>
<td><strong>SDRadar</strong></td>
<td>Software Defined Radar</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td><strong>SEI</strong></td>
<td>Spectrum Evaluation Interval</td>
</tr>
<tr>
<td><strong>SINR</strong></td>
<td>Signal-to-Interference-plus-Noise Ratio</td>
</tr>
<tr>
<td><strong>SLA</strong></td>
<td>Sense-Learn-Avoid</td>
</tr>
<tr>
<td><strong>SLN</strong></td>
<td>Sense-Learn-Notch</td>
</tr>
<tr>
<td><strong>SNR</strong></td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td><strong>SPA</strong></td>
<td>Sense-Predict-Avoid</td>
</tr>
<tr>
<td><strong>SPN</strong></td>
<td>Sense-Predict-Notch</td>
</tr>
<tr>
<td><strong>SRA</strong></td>
<td>Sense-React-Avoid</td>
</tr>
<tr>
<td><strong>SRN</strong></td>
<td>Sense-React-Notch</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td><strong>SW UCB1-tuned</strong></td>
<td>Sliding Window UCB1-tuned</td>
</tr>
<tr>
<td><strong>TS</strong></td>
<td>Thompson Sampling</td>
</tr>
<tr>
<td><strong>UCB1</strong></td>
<td>Upper Confidence Bound 1</td>
</tr>
<tr>
<td><strong>USRP</strong></td>
<td>Universal Software Radio Peripheral</td>
</tr>
<tr>
<td><strong>UWB</strong></td>
<td>Ultra-wideband</td>
</tr>
<tr>
<td><strong>VST</strong></td>
<td>Vector Signal Transceiver</td>
</tr>
<tr>
<td><strong>Virginia Tech</strong></td>
<td>Virginia Polytechnic Institute and State University</td>
</tr>
<tr>
<td><strong>ZOROW</strong></td>
<td>Zero-Order Reconstruction of Waveforms</td>
</tr>
</tbody>
</table>
Acknowledgments

The presented work was from collaborative effort between The Pennsylvania State University and the U.S. Army Research Laboratory (ARL) in Adelphi, MD. The research was supported by U.S. Army Research Office (ARO) grant W911NF-15-2-0066 and the Research Associateship Program with Oak Ridge Associated Universities (ORAU) through Cooperative Agreement (CA) W911NF-16-2-0008. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of ARO or ORAU.

I would like to thank my advisor, Dr. Ram Narayanan, for his invaluable support throughout this journey. During my time at Penn State, Dr. Narayanan has taught me about remote sensing, informed me of conventions for performing research in the real-world, and continued to remind me to "go, go, go!" I am glad that I was lucky enough to stumble into his lab. Similarly, I would like to thank Dr. Anthony Martone for serving as my mentor and encouraging me to explore new and exciting research ideas. All of the interesting work done on the SDRadar project would not have been possible without his coordination. Thank you to Kelly Sherbondy for teaching me the importance of effectively communicating complex ideas in simple terms. I am grateful that Kelly was able to arrange for me to conduct research at ARL. I would also like to thank Dr. Timothy Kane, Dr. Julio Urbina, and Dr. Karl Reichard for serving as members of my committee and providing valuable insight into this work.

Thank you to Benjamin Kirk for the many technical discussions that contributed to this research as well as the many nonsensical discussions that kept me sane through this process. I would like to thank the team of students and contractors on the SDRadar project: Jon Owen, Jason Zsak, Charles Thornton, Austin Egbert, Jordan DeVault, and Brandon Ravenscroft. The SDRadar project would not have been possible without their collaboration. Similarly, thank you to the professors from the collaborating universities: Dr. Shannon Blunt, Dr. Mike Buehrer, and Dr. Charles Baylis.

I thank Dr. Kyle Gallagher and Dr. Brian Phelan for their support during
my time at ARL. Last but not least, thank you to Colin Kelly, David Alexander, and the many members of the CATO lab at Penn State for the useful technical discussions and laughs over the years.
Dedication

This dissertation is dedicated to my parents, Yefim Kovarskiy and Alexandra Kovarskaya. Your guidance and encouragement has provided me with a solid foundation in life. You have not only led me to complete a PhD but you also continue to shape my future. For this, I’m grateful.
Chapter 1

Introduction

This dissertation analyzes a model and real-time implementation of predictive spectrum sharing for cognitive radar, followed by the integration of this system into a metacognition model for radar. These radars were developed as a result of collaborative efforts from The Pennsylvania State University, the U.S. Army Research Laboratory, and other universities to address the growing problem of radio frequency (RF) spectrum crowding. The goal of this effort was to develop spectrum sharing capabilities for pulsed radar systems to maintain performance in congested RF environments. Development priorities were guided by the objective of implementing these spectrum sharing radar capabilities in real-time using commercial off-the-shelf (COTS) software-defined radio (SDR) hardware.

Spectral prediction for radar coexistence provides an improvement over methods which adapt to interfering emitters via reaction. This configures the radar to adapt in anticipation of changes to the environment and build knowledge of the spectrum for use by other cognitive RF processes. Through PSU/ARL collaboration, an initial cognitive software-defined radar (SDRadar) with spectral prediction was implemented with real-time learning and processing. In parallel, similar SDRadars employing other cognitive radar (CR) strategies were developed by ARL collaborating universities including Virginia Tech, University of Kansas, and Baylor University. These CR strategies included both alternative methods to spectral prediction and a variety of waveforms to be used in conjunction with prediction. Experiments across these groups showed that performance benefits and drawbacks of the various CR strategies were dependent on the RF environment.
Subsequently, spectrum sharing SDRadar development led to integrating these disparate CR strategies, including prediction, into a single merged platform. Theories of metacognition influenced a processing architecture that enables cognitive radar to combine and switch between CR strategies, thereby further optimizing performance. Drawing from the field of reinforcement learning and neuroscience, this architecture expands Joe Mitola’s and Simon Haykin’s ideas of cognitive RF to metacognition. This concept was used to combine the aforementioned predictive SDRadar with the merged metacognitive radar to adapt to changing RF environments. Here, we investigate the design and performance of both spectral prediction and metacognition for spectrum sharing radar implemented in real-time on SDR hardware.

1.1 Dissertation Overview

This dissertation is organized as follows. Chapter 1 introduces the motivation and contributions of the work. Chapter 2 introduces the concept of cognition, discusses cognitive spectrum sharing architectures, and presents a state-of-the-art review. Chapter 3 introduces basic radar phenomenology, discusses Doppler processing, and analyzes the consequences of radar spectrum sharing in the presence of interference. Chapter 4 presents the implementation of spectral prediction for cognitive radar on SDRadar hardware while Chapter 5 analyzes the cognitive SDRadar system’s performance in a variety of interference scenarios. Chapter 6 analyzes the spectral prediction implementation against other cognitive SDRadar implementations including a reinforcement learning-based approach as well as spectral prediction combined with notched waveforms. Chapter 7 introduces the notion of metacognition, presents the merged metacognitive SDRadar architecture, and discusses the restless bandit problem for optimally adapting CR strategy with respect to a changing spectral environment. Chapter 8 analyzes the performance of the merged metacognitive SDRadar architecture by comparing a variety of restless bandit-based algorithms and configurations. Chapter 9 provides concluding statements and suggests areas for future work.
1.2 Motivation

The radio frequency spectrum has become a scarce resource due to increased demands for information throughput [1]. With the advent of technologies such as autonomous vehicles and Internet of Things (IoT), a rising number of everyday objects are embedded with wireless capabilities. Additionally, next-generation 5G cellular networks require additional bandwidth to meet these demands. Traditional Federal Communications Commission (FCC) regulations reserve portions of the RF spectrum by statically allocating frequency bands to licensed users. To satisfy the growing spectrum demands, more recent policies propose communication systems share the spectrum with incumbent radar systems at the 3.5 GHz and 5 GHz bands [2–4]. This poses new challenges for radar systems that must coexist with unlicensed emitters.

For radar systems, unlicensed emitters may act as RF interference (RFI) which has been shown to degrade performance for several radar applications [5,6]. Some radars rely on their allocated frequencies to penetrate specific materials during operation [7]. Additionally, advances in radar are increasing system bandwidth requirements [8]. Dynamic spectrum access (DSA) provides a solution to this by adaptively reallocating spectrum which undergoes periods of inactivity [9,10].

While numerous works have applied DSA to communication networks, the notion of a spectrum sharing radar is still an emerging idea. Spectral coexistence puts radar systems at risk for harmful RFI which degrades target detection performance [5,6,11,12]. Similarly, radars can negatively impact the bit error rate (BER) of communication systems [3,13]. Wang et al [6] considers the case of a high-power radar coexisting with a lower power long-term evolution (LTE) system. Despite the power difference, communications still negatively impact radar target detection. While LTE utilizes error correction to tolerate RFI, many existing radar systems do not have such capability. This affirms the need for radar DSA to mitigate mutual interference effects while maintaining system performance.

For spectrum sharing, the cognitive radar must respond to avoid measured RFI activity. This response consists of either a reaction or prediction of changes in the environment. A predictive decision infers changes in RFI to anticipate future availability [10,14]. A reactive system only adapts after environmental changes are observed. While reaction is computationally efficient, reaction time becomes
a limiting factor. Reaction has a delay in adjusting to changes in RF activity which results in errors [15]. While spectrum sharing for radar is still an emerging application space, prior work that applies spectral prediction to cognitive radar is sparse. Similarly, only a few have documented real-time hardware implementations of spectrum sharing radars. This dissertation seeks to explore spectral prediction for radar spectrum sharing via a stochastic model-based approach. This approach draws from similar methods applied to spectrum sharing for communications [16,17] and expands it to a real-time radar implementation on SDR hardware.

To emphasize the practicality of this spectral prediction approach, cognitive SDRadar performance was assessed using a comprehensive set of RFI scenarios emulated in real-time. Further analysis in this dissertation compares stochastic model-based prediction to a reinforcement learning approach for predictive spectrum sharing. In addition, prediction is paired with different waveform types to demonstrate flexibility of the approach. The notion of the perception-action cycle (PAC) provides a biological framework to inspire the design of cognitive systems [18]. This framework is a closed-loop model that actualizes cognitive processing. All the various implementations of prediction, reaction, and waveform design for cognitive radar present adaptations to the components of this PAC model.

Prior work [15] and analysis in this dissertation show that certain CR strategies are better suited to particular RFI scenarios and conditions. In a dynamic electromagnetic environment (EME), these conditions can change rapidly. Furthermore, technological innovations are increasing spectral congestion which make the dynamics of EMEs more volatile. This creates a need for more robust cognitive radar solutions with the ability to adapt to changes in RFI conditions. As a response, we invoke the bio-inspired concept of metacognition to monitor the EME and optimally adapt the PAC to mitigate performance degradation. The prospective best-of-breed framework conducts self-monitoring and self-regulation to switch CR strategies over time. In this dissertation, the presented metacognitive radar implementation maintains spectrum sharing performance while adapting to changing RFI scenarios in real-time.
1.3 Dissertation Objectives

The goals of this work pertain to the design and analysis of spectral prediction for radar spectrum sharing and metacognitive radar implemented on SDR hardware. The analysis of spectral prediction is directed by the following objectives:

- Leverage a stochastic model-based prediction approach for radar spectrum sharing.
- Present a scheme to optimize the stochastic model’s decision rule to further improve performance.
- Demonstrate a real-time SDR implementation of the cognitive radar with online stochastic model learning and RFI prediction.
- Define a detectable range improvement metric to quantify the benefit of radar spectrum sharing.
- Characterize predictive spectrum sharing performance over a comprehensive set of RFI scenarios.
- Compare the stochastic model for prediction to other predictive models and combine prediction with notched radar waveforms for RFI avoidance.

Influenced by the comparison of prediction to other CR strategies, the objectives relating to metacognitive radar are as follows:

- Expand the state-of-the-art for cognitive radar by introducing the concept of metacognition for spectrum sharing radar.
- Leverage the aforementioned stochastic model to obtain metrics to characterize the EME.
- Present a metacognitive control architecture based on case-based reasoning and the restless bandit problem to optimize CR strategy selection.
- Compare the performance of various restless bandit-based algorithms for CR strategy selection.
• Analyze metacognition performance with spectrum sharing objectives compared to radar target detection objectives.

• Present the Jensen-Shannon divergence as a metacognition metric to describe target detection quality as self-evaluation feedback.

• Characterize performance with a real-time SDR implementation in a dynamic EME.
Cognitive RF and Spectrum Sharing

2.1 Introduction to Cognition for RF Systems

The fields of neuroscience and psychology present a broad set of theories to define the cognitive processes of humans and living organisms. More specifically, cognition describes the underlying mechanisms for sensing, perceiving, and acting in response to acquired information. These processes include learning, reasoning, formation of knowledge, and numerous other brain functions [19]. The overarching theories describe how humans function as intelligent autonomous agents in a complex real-world environment. In the book *Cortex and Mind: Unifying Cognition* [18, 19], Fuster decomposes the principles of cognition into four components: the perception-action cycle (PAC), memory, attention, and intelligence. The PAC (shown in Figure 2.1) describes the process of sampling information in an environment, processing this information, and performing an action based on the information. Once an action is performed, the cycle repeats as the effects of the action on the environment are observed. Information obtained by the PAC can be stored as memory to inform future perceptive processing and decision making. Attention refers to the allocation of resources for sampling, processing, and storing information. While difficult to define, intelligence refers to the capacity to learn and optimize the performance of desired objectives [18]. With continually increasing computing power and advances in machine learning, many have turned to these bio-inspired cognition theories for inspiration in improving autonomous technologies.

Cognition provides a comprehensive architecture to aid the development of
Figure 2.1: Visual representation of the perception-action cycle. Cognition involves cyclically observing the environment and its response to actions performed within the environment. Perceptions stores processed information in memory and allocates attention resources for executing cognition.

intelligent machines such as autonomous vehicles, natural language processors, and next-generation RF systems. These technologies exist in problem spaces containing complex dynamic environments with numerous unknowns. In [20], Haykin defines cognitive dynamic systems that "build up rules of behavior over time through learning from continuous experiential interactions with the environment, and thereby deal with environmental uncertainties." Similarly, RF technologies exists in a complex problem space with many uncertainties making them opportune candidates to utilize cognition. Mitola, in [21], was the first to conceptualize a fully cognitive radio that autonomously optimizes all possible parameters for wireless communication based on network demands. The cognitive radio concept employs SDRs to fully reconfigure hardware and processing parameters with full awareness of the geospatial, temporal, and spectral environment. Fully cognitive radio is considered a natural path for SDRs to evolve.

In an effort to address spectral congestion (as described in Section 1.2), the
wireless communications community proposed DSA which leverages aspects of
Mitola’s concept for cognitive radio. Spectral crowding paired with increasing
demand for data throughput has created a need for cognitive spectrum sharing.
This demand is driven by an increasing number of technologies which require
RF spectrum resources. The technologies range from mobile handheld devices
and objects embedded with IoT capabilities to a variety of radars for defense,
autonomous vehicle, and weather applications. Spectrum sharing enables these
devices to coexist while maintaining performance. In developing DSA systems for
communication networks, elements of the cognitive PAC have been realized using
SDR hardware. Spectrum sharing research places large emphasis on algorithms
for reliable sensing and detection of spectrum opportunity, building knowledge-
bases to aid spectrum allocation, and control schemes for accessing spectrum
opportunity [10,22]. These spectrum sharing aspects compose the elements of a
PAC to realize cognitive radio.

Over time, Haykin drew inspiration from innovations in cognitive radio and
extended the concept to cognitive dynamic systems with an emphasis on radar.
Beyond wireless transmission of information and spectrum sharing, cognitive radar
focuses on applying the PAC to all possible radar objectives. These objectives
include target detection, tracking, imaging, and even autonomous control. The
application of cognition to radar presents a natural parallel to biology. Bats
demonstrate an example of a fully cognitive sonar system for locating and identifying
insects. These bats adapt their emitted waveform frequency and pulse repetition
interval (PRI) to optimally search and track prey [23]. Their PAC also allows them
to distinguish desired targets from clutter while rapidly maneuvering through a
forest. By replacing sound waves with electromagnetic waves, this biological model
for processing could be applied to autonomous aircraft as a realization of cognitive
radar.

Given radar’s broad application space, cognition could apply to a variety of
objectives and elements of radar processing. Some previous works have considered
cognitive radars which adapt waveform parameters to optimize target tracking
and recognition similar to a bat’s biosonar [24–26]. Those experiments optimized
parameters such as waveform envelope, duration, bandwidth, and PRI. Others
have explored cognitive radar aspects such as automatic target recognition and
RF hardware adaptation [27–29]. In [30], a knowledge-aided cognitive approach
is applied to optimal clutter suppression for multiple-input and multiple-output (MIMO) radar with space-time adaptive processing. While these radar implementations present intelligent learning-based adaptation, they leave the problem of spectral congestion unaddressed.

As cognitive radio technology began to develop, the radar community was slower to address the problems of spectral congestion. Evolving FCC policies have begun permitting wireless communication networks to share the spectrum with incumbent radar systems [2–4]. This places further burden on radars to adopt spectrum sharing solutions. Investigations have shown that both radars and communication networks can mutually degrade performance during spectral coexistence [6,12]. In response to this problem, radar technologies began to adopt cognition for spectrum sharing. Similar to DSA for communications, spectrum sharing cognitive radar uses the PAC to adapt its transmit waveform to optimally avoid RFI while maximizing target detection objectives. Examples of cognitive schemes for radar spectrum sharing include [8,15,31]. Stinco et al [8] presents a general architecture for spectrum sharing cognitive radar which describes a PAC that pairs spectrum sensing and channel characterization. [31] demonstrates a DSA implementation based on compressive sensing and radar operation over disjoint frequency bands. [15] evaluates a real-time hardware implementation of cognitive SDRadar for reactive DSA in a variety of RFI scenarios. Since work regarding spectrum sharing radar is sparse, we intend to further develop the state-of-the-art for this cognitive radar variant. This dissertation presents a real-time SDRadar implementation focused on prediction as opposed to the reactive scheme described in [15]. The predictive spectrum sharing radar implementation will later develop into the topic of metacognition in Chapter 7. The rest of this chapter presents a literature review of cognitive spectrum sharing schemes and techniques.

2.2 Review of Spectrum Sharing Architectures

A variety of PAC architectures and algorithms have been proposed for spectrum sharing. The PAC for DSA can be decomposed to sensing of the RF environment (spectrum sensing), perception of spectral opportunities, and action in the form of transmit adaptation. The dimensions defining the environment of the cognitive system vary depending on the application and hardware capabilities. Spectrum
sharing is conducted across a combination of frequency, temporal, and spatial domains [9]. Simpler baseline implementations consider spectrum sensing and waveform adaptation in the frequency and time dimensions. Spatial domain awareness may require antenna arrays with adaptive beam steering and fully reconfigurable RF front-ends [32]. This requires significantly more computing resources and exponentially increases the dimensionality of sensing data.

To accurately assess the RF environment and avoid colliding with emitters, robust spectrum sensing methods are necessary. RFI sensing methods include detection theoretic approaches such as energy detection, cyclostationary feature detection, eigenvalue-based methods, as well as others. Energy detection acts as a low complexity baseline detector compared to more processing intensive approaches such as cyclostationary and eigenvalue-based detection which require spectral correlation [9]. Additionally, energy detection requires no prior knowledge of coexisting RF emitters to perform sensing, making this approach desirable for high speed DSA applications. Traditionally, receiver operating characteristic (ROC) curves evaluate the performance of sensing methods [9]. ROC curves describe the trade-off between a method's false alarm and missed detection rate. Robust spectrum sensing relies on maintaining a good detection error trade-off in low signal-to-noise ratio conditions.

When coexisting with RF emitters in the environment, the cognitive system may share the spectrum cooperatively or non-cooperatively. Cooperative DSA allows devices to maximize spectral efficiency by making collaborative decisions on spectrum availability. This requires an environment consisting of only cognitive radars/radios with allocated resources and protocols for cooperative handshaking [33]. In non-cooperative sharing, the cognitive system assesses the spectrum without any communication with other RF emitters. Non-cooperation requires less resources, has less latency, and is more robust when interacting with other RF emitters. While cooperative devices may have better spectrum efficiency [34], performance may degrade in an environment with non-cooperative or statically allocated RF systems.

In deciding how to access the available spectrum, the cognitive radar must respond to the environment based on the perception of sensed RFI activity. This response to avoid RFI consists of either a perceived reaction to or prediction of changes in the environment. A predictive decision infers changes in RFI before they occur which allows the system to adapt in anticipation of these changes [10,14]. A
reactive system only adapts after a change in the environment is observed. While reaction is computationally efficient, this approach’s performance is limited by reaction time. Reaction has a delay in adjusting to environmental changes which result in errors during reaction time [15]. Prediction mitigates these errors at the cost of additional memory and processing resources.

Predictive channel access has been widely explored by the communications community with a few examples extended to radar. With the rising popularity of machine learning, approaches such as neural networks, Support Vector Machines (SVMs), and Hidden Markov Models (HMMs) have been applied to DSA for communications [35–38]. Others have explored model-based approaches in the form of binary time series analysis [39] and stochastic methods [16, 17, 40]. These machine learning approaches are computationally intensive during training and risk training bias. The model-based approaches require less computation with stochastic methods directly describing random RF activity. Most of these communications applications only consider a single channel for prediction while a radar system will typically consider multiple frequency channels simultaneously. For predictive channel access in radars, [8] presents a single channel HMM approach but does not discuss the impact of RFI on radar processing. Similarly, [41] discusses multi-channel reinforcement learning approaches (Deep-Q Networks and Markov Decision Processes) but does not describe the effects of collisions on radar performance. While these reinforcement learning approaches perform well for deterministic RFI, performance degrades for random switching interference [42]. Simulations presented in [38, 43] consider Recurrent Neural Network (RNN) and SVM based prediction models for radar without considering the real-time implications of offline training. Offline training for such models may become unsuitable in rapidly changing RF environments. This dissertation will present a real-time stochastic approach based on [44–46] with the ability to tolerate random RFI and consideration of radar processing performance.

While models for prediction and reaction define the perception component of the PAC, action methods determine the cognitive system’s response to the perceived environment. Specific methods for action depend on the application’s waveform requirements. In DSA for communications, systems that use Orthogonal Frequency-Division Multiplexing (OFDM) waveforms can fill available spectral holes over multiple discontinuous sub-bands [10]. RF systems with antenna arrays and
reconfigurable front-ends may have action methods that involve beam steering and retuning the RF hardware respectively. Radars may require much wider bandwidth than communication networks and have different waveform operating characteristics. Pulsed radar systems require periodic access to the spectrum for transmission. For such radars, the classic linear frequency modulated (LFM) chirp waveform requires access to continuous uninterrupted blocks of the spectrum. DSA with an LFM chirp requires using an avoid method for action. Some have suggested using frequency modulated (FM) noise waveforms with notches for coexistence with communication systems [47,48]. Others suggest implementing a notch action method using stepped frequency waveforms [49]. Here, we consider frequency domain avoidance and notching as action methods for a pulsed radar with spectrum sharing.
Pulse-Doppler Radar

3.1 Introduction to Radar

RAdio Detection And Ranging (RADAR) is the measurement of reflected radio waves to determine physical properties of an object. These physical properties include position, velocity, dimensions, material characteristics, and more. Basic radars determine distance to an object by measuring the time taken for a transmitted signal to reflect back to the receiver. Velocity is measured via the Doppler effect or the frequency shift of the reflected wave over time. The origins of radar date back to the 19th century when Heinrich Hertz observed radio waves reflected by metallic objects [50]. Over the years several individuals would develop and experiment with radar-like devices. In 1903, Christian Hülsmeye would apply for a patent describing a device that transmits and receives reflected radio waves for detecting the presence of ships. This device called the telemobiloskop, could detect the presence of a ship without the ability to determine range. Later incarnations of the apparatus could estimate range via a trigonometric approach.

Several nations would continue to develop radar capabilities prior to World War II. During the 1930s, Albert H. Taylor, Leo C. Young, and Lawrence A. Hyland would conduct early radar development for the United States Navy Research Laboratory, leading to the coining of the term RADAR in 1939 [50]. Through the 1940s, radars would adopt the magnetron for generating higher power and higher frequency RF transmissions. This wartime effort would spark significant radar development resulting in the technology becoming a major field of research [50].
Initially, radars were designed for military applications relating to air defense such as early warning, air-to-surface, and navigation. Eventually, radars would extend to fields such as meteorology, astronomy, and even vital sign monitoring [50]. Further developing RF technologies have led to an interest in systems capable of operating over ultra-widebands (UWBs). This enables pulse compression radars to achieve fine spatial resolution which is desired for imaging and the detection of buried targets [51]. Similarly, designs for multifunction systems that embed radar with communication waveforms require UWBs [8]. This progression further exacerbates the spectrum congestion problem. Here, we consider the problem of spectrum sharing using a pulse-Doppler radar system with the objective of maximizing bandwidth while also maximizing signal-to-interference-plus-noise ratio (SINR). The next section continues by describing standard range-Doppler processing with an LFM chirp waveform and then considers the impacts of spectrum sharing with RFI on radar processing.

3.2 Radar Processing

3.2.1 Matched Filtering

This section considers a baseline pulse compression radar that transmits an LFM chirp waveform. The entire radar processing cycle is defined starting with extraction of range profiles from each received pulse, construction of a range-Doppler map, and automatic target detection. The presented radar processing follows the descriptions given by Richards et al [52]. This work considers a pulsed LFM chirp radar for proof-of-concept with spectrum sharing. The presented processing could easily be extended to other waveform types and more advanced techniques such as noise waveforms and mismatched filtering.

An LFM chirp consists of a sinusoid which linearly sweeps over a frequency range defined as follows

\[ u(t) = Ae^{-j\theta(t)}, \]  

\[ \theta(t) = \pi \left( \frac{f_e - f_s}{\tau} \right) t^2 + 2\pi f_s t. \]
Figure 3.1: Example of a chirp signal over time with linearly increasing frequency. Only the imaginary part is shown.

\( u(t) \) (shown in Figure 3.1) describes the radar transmit pulse as a complex sinusoid with amplitude \( A \) over time \( t \). The time-dependent quadratic phase \( \theta(t) \) results in a linear frequency sweep starting at frequency \( f_s \) and ending at \( f_e \). \( \tau \) describes the pulse width in time and the term \( \left( \frac{f_e - f_s}{\tau} \right) \) determines the sweep rate of the chirp. The quadratic phase progression corresponds to linear frequency sweep by the relationship to angular frequency \( \omega(t) = \frac{d\theta(t)}{dt} \). A pulsed radar system periodically transmits \( u(t) \) and receives RF signals in between transmissions to measure a return pulse or echo. The delay of this return pulse corresponds to target range \( R \),

\[
R = \frac{c\Delta t}{2}.
\]

(3.3)

Above, \( c \) describes the velocity of the transmitted electromagnetic wave while \( \Delta t \) is the time delay between transmitting and receiving the echo. After receiving a target echo \( y(t) \), pulse compression applies a matched filter for recovering a range profile \( v(t) \),

\[
v(t) = \mathcal{F}^{-1}\{Y[n]U^*[n]\}.
\]

(3.4)

A matched filter involves a cross-correlation between the received pulse \( y(t) \) and a
copy of the original transmitted chirp $u(t)$ [52]. Equation (3.4) describes a frequency
domain multiply for a more efficient matched filter implementation. $Y[n]$ and $U^*[n]$
represent the Fourier transforms of the received echo and transmit chirp complex
conjugate respectively. Let $n$ define the index of the frequency bin. The $F^{-1}$
operator is the inverse Fourier transform, used here for recovering the range profile
$v(t)$. In the presence of a target, $v(t)$ contains a peak which can be used to recover
$\Delta t$. This $\Delta t$ is defined by the difference in time from when the last pulse was
transmitted and the echo peak at some $t$ was received. Using this relationship, the
range of the target can be recovered via Equation (3.3). The target peak main-lobe
width or range resolution [52] for pulse compressed LFM chirps is defined by

$$R_{\text{res}} = \frac{c}{2B}. \quad (3.5)$$

Larger bandwidth $B = f_e - f_s$ results in a finer range resolution. This makes
wider bandwidths more desirable for detecting closely spaced targets. To achieve a
fine $R_{\text{res}}$, we will approach spectrum sharing with a joint objective of maximizing
bandwidth while avoiding RFI.

### 3.2.2 Doppler Processing

The radar continually transmits pulses at some pulse repetition frequency (PRF)
denoted by $F_{\text{PRI}}$. After obtaining a collection of $M$ range profiles, Doppler processing
extracts the velocity of the target. Doppler processing measures the Doppler shift
from the phase progression of the target during the collection of $M$ echos [52]. The
collection of $M$ pulses is known as a coherent processing interval (CPI). For now,
we consider a standard radar with no spectrum sharing or interference present. This
means that $f_s$ and $f_e$ remain constant throughout the CPI. To obtain information
from the $m^{th}$ pulse of a CPI, the range data $v[t, m]$ now has two-dimensional indices.
In this section, $t$ is redefined as a discrete time index $t \in \{1, 2, \ldots T\}$ and $m$ is the
pulse number $m \in \{1, 2, \ldots M\}$. Let $V$ be a $T \times M$ matrix containing $T$ range
cells (fast-time) and $M$ range profiles (slow-time)

$$V = \begin{bmatrix}
v[1, 1] & \ldots & v[1, M] \\
\vdots & \ddots & \vdots \\
v[T, 1] & \ldots & v[T, M]
\end{bmatrix} = \begin{bmatrix}
v_1[1] & \ldots & v_1[M] \\
\vdots & \ddots & \vdots \\
v_T[1] & \ldots & v_T[M]
\end{bmatrix} = \begin{bmatrix}
v_1^T \\
\vdots \\
v_T^T
\end{bmatrix}. \quad (3.6)$$
Through abuse of notation, $v_t$ defines the slow-time signal for the $t^{th}$ range cell or a range cell’s signal over the collection of $M$ pulses. A range-Doppler map $D$ is formed by taking a Fourier transform of $V$ across the $m$ dimension or $\mathcal{F}_m\{V\}$ defined by

$$D = \mathcal{F}_m\{V\} = \left[ \mathcal{F}\{v_T^\top\} \right] = \begin{bmatrix} d[1,1] & \ldots & d[1,M] \\ \vdots & \ddots & \vdots \\ d[T,1] & \ldots & d[T,M] \end{bmatrix}. \quad (3.7)$$

The range-Doppler map contains $d[t,m]$ where the $m$ maps to a Doppler frequency $f_D$ bounded by the PRF which is subsequently used to compute target velocity $s_r$,

$$f_D = \left( \frac{2(m-1)}{M} - 1 \right) \left( \frac{F_{PRF}}{2} \right), \quad (3.8)$$

$$s_r = \frac{f_D \lambda}{2}. \quad (3.9)$$

$\lambda$ describes the wavelength of the radar’s carrier frequency. Now that the range-Doppler map is constructed, the radar performs target detection to recover velocity and range information.

### 3.2.3 Target Detection

To detect a target signal, the received target echo must be distinguished from background noise and clutter. We first consider the case with only background noise. The range-Doppler map can be decomposed into target signal and noise components: $D = D_R + D_N$. To determine the location of target signals, a binary hypothesis test is performed on each range-Doppler cell $d[t,m]$,

$$H_0 : \quad d[t,m] = d_N[t,m] \quad \text{(No Target)}$$

$$H_1 : \quad d[t,m] = d_T[t,m] + d_N[t,m] \quad \text{(Target)}. \quad (3.10)$$

The subscripts R and N denote the target and noise components of the range-Doppler map respectively. Determining the hypothesis requires placing a threshold
on the power of each cell

\[ z[t, m] = \left| d[t, m] \right|^2 \frac{H_1}{H_0} \Lambda. \tag{3.11} \]

This work considers the cell-averaging constant false alarm rate (CA-CFAR) detector for computing an adaptive threshold \( \Lambda \) based on a local estimate of the background noise [52]. \( z[t, m] \) describes the state of each cell under test (CUT) according to the threshold

\[ \Lambda = \alpha_{CA} \sigma_N^2. \tag{3.12} \]

The threshold is computed and Equation (3.11) is applied to each cell to detect the presence of targets. \( \alpha_{CA} \) is a CFAR threshold constant and \( \sigma_N^2 \) is a local estimate of background noise in the cells surrounding the CUT. While estimating the background noise, guard cells are defined around the CUT (Figure 3.2) to avoid any target signal leaking into the estimate. The threshold \( \Lambda \) is recalculated for each cell based on Equation (3.12). Let the range-Doppler CUT indices be \( t_T \) and \( m_T \) while \( g \) defines a guard cell radius. Define the indices for background noise

![Figure 3.2: Example of indexing for CA-CFAR. Cells for background noise estimation highlighted in red. The CUT labeled and colored gray. Guard cells assigned based on radius \( g = 1 \) and labeled GD. The CUT and guard window is moved across the range-Doppler map to perform detection on each cell.](image-url)
estimation as \( t_N = \{ t : |t_T - t| > g \} \) and \( m_N = \{ m : |m_T - m| > g \} \),

\[
\sigma_N^2 = \sum_{t_N} \sum_{m_N} |d(t_N, m_N)|^2.
\] (3.13)

The CFAR constant \( \alpha_{CA} \) is defined by the number of background cells for estimation \( N_{CA} \) and an operator defined probability of false alarm \( P_{FA} \) given by

\[
\alpha_{CA} = N_{CA}(P_{FA}^{-1/N_{CA}} - 1).
\] (3.14)

Under ideal conditions where the background noise is complex Gaussian with zero-mean, the selected false alarm rate \( P_{FA} \) remains constant across all signal-to-noise ratios (SNRs). As a result, the probability of target detection largely depends on the SNR [52]. Non-idealities such as the presence of RFI have negative impacts on radar target detection. RFI impacts must be carefully considered when radars operate in congested EMEs.

### 3.3 Interference Impacts on Radar Processing

To operate in a congested spectrum, a pulsed radar must adapt the transmitted pulses to avoid colliding with other RF emitters. Coexisting with no adaptation can result in reduced ability to detect targets and poor range-Doppler SINR. During operation, a transmitted radar waveform travels to a target, scatters off the target, and travels back to the receiver. This process results in possibly significant signal attenuation due to a number of factors. Due to this, matched filtering and Doppler processing become necessary to attain an acceptable SNR for target detection. The radar range equation models the loss in received power \( P_r \) due to factors such as path loss, antenna gain, and target properties

\[
P_r = \frac{P_t G_t G_r \lambda^2 \sigma_r}{(4\pi)^3 R^4} = P_t |L|^2.
\] (3.15)

\( P_t \) describes the transmitted power while \( G_t \) and \( G_r \) are transmit and receive antenna gains. The \( \sigma_r \) term describes the radar cross-section of the target and the target range is \( R \). Many variations of the radar range equation exist to account for factors such as atmospheric attenuation. To account for various factors that impact
\( P_r \), we define the frequency dependent loss term \( L \). In the case of a radar with RFI, consider a received signal \( Y[n] \) decomposed into target \( U[n] \), interference \( X[n] \), and noise \( N_f[n] \) components in the frequency domain

\[
Y[n] = L[n]U[n] + X[n] + N_f[n]. \tag{3.16}
\]

Applying the matched filter to \( Y[n] \) using Equation (3.4) gives

\[
v[t] = \mathcal{F}^{-1}\left\{ (L[n]U[n] + X[n] + N_f[n])U^*[n] \right\} = v_R[t] + v_I[t] + v_N[t]. \tag{3.17}
\]

This operation returns \( v[t] \) to the time domain where \( t \) is a discrete time index \( t \in \{1, 2, \ldots, T\} \) and \( n \) is a discrete frequency index of the same size. Here, the subscripts R, I, and N describe the target, RFI, and noise contributions, respectively. The contribution of the RFI to the range profile is the cross-correlation of the transmitted LFM chirp with the received RFI \( v_I[t] = \mathcal{F}^{-1}\{X[n]U^*[n]\} \).

To evaluate the impact of RFI on a range profile, consider a pulsed radar with in-phase and quadrature (IQ) sampling at 100 MSamples/s at some arbitrary center

![Spectrum of Chirps + Interference](image)

**Figure 3.3:** Plot shows the spectrum of an interference source (blue), a full bandwidth chirp (black), and a smaller bandwidth chirp avoiding the interference (red).
frequency. These parameters would correspond to a full bandwidth LFM chirp waveform close to 95 MHz to account for filter roll-off (from -47.5 to +47.5 MHz baseband). Consider sinusoidal interference at -10 MHz baseband. If the radar reduces bandwidth to avoid the RFI, $v_I[t]$ is minimized. Figure 3.3 shows a full bandwidth LFM chirp, the sinusoidal RFI, and a predicted chirp avoiding the RFI. Figure 3.4 shows the range profile that results from the full bandwidth chirp colliding with the sinusoid while the predicted chirp avoids the RFI. This shows a $\sim$20 dB improvement in the average SINR due to $v_I[t]$. By colliding with RFI, the radar has a reduced ability to detect weak targets below -40 dB. The various factors described in the radar range Equation (3.15) could result in weaker targets such as target cross-section $\sigma_r$ and range $R$.

Now, considering the effects of RFI on Doppler, $v[t]$ from Equation (3.17) is fed into Equation (3.7) for Doppler processing. Similar to $v[t]$, due to linearity, the range-Doppler map can be decomposed into target $D_R$, RFI $D_I$, and noise $D_N$ components $D = D_R + D_I + D_N$. The contribution of RFI to the range-Doppler map is the result of the Fourier transform of the slow-time dimension for each
Consider a pulsed radar system transmitting full bandwidth chirp waveforms with the same 100 MSamples/s sampling rate from earlier. This system generates range-Doppler maps with 400 pulses per CPI and a 9.8 kHz PRF. First, we consider the case with no RFI $D_t = 0_{T \times M}$ and four point targets in the scene (Figure 3.5). Each target has a different loss factor with the leftmost target having $L = 1$, the target at range 325 and Doppler 0.25 has a loss $L = 0.7$, range 225 and Doppler 0.25 has loss $L = 0.4$, and the rightmost target has $L = 0.2$. This radar processing shown is simulated in MATLAB with a -90 dB noise floor (not visible in plots) and the radar transmit power at 0 dB. CA-CFAR target detection is run on each target scene with a $P_{FA} = 10^{-7}$ and a guard cell radius of $g = 1$. In the case with no RFI, all targets are successfully detected with minimal false alarms.

Next, a constant -30 dB tone at -10 MHz baseband is included in the radar processing. This result shown in Figure 3.6, shows zero-Doppler artifacts which produce false targets when run through the detector. An RF emitter with constant phase during the radar’s CPI will result in similar zero-Doppler artifacts as a result.

**Figure 3.5:** Range-Doppler map with no RFI present and four point targets marked by the red arrows.
Figure 3.6: Range-Doppler map with a constant tone present as RFI. The constant phase of the RFI creates artifacts across zero-Doppler which results in false alarms when processed by a CA-CFAR detector.

of processing. Figure 3.7 considers a -30 dB tone linearly sweeping from -10 MHz to 10 MHz baseband. The one full sweep occurs every 100 pulses which results in four full sweeps during a single CPI. This phase progression results in structured RFI artifacts in the range-Doppler map. This produces additional false alarms where the artifacts are visible and causes a missed detection on the target that overlaps with RFI distortions. Lastly, Figure 3.8 considers a tone randomly hopping between -10 MHz and 10 MHz baseband with a random location at each radar pulse. The random phase structure results in distributed lower power distortion over the entire range-Doppler map. For CA-CFAR processing, two missed detections occur on the weaker targets located at the bottom-right. Due to the distributed RFI artifacts, few false alarms occur as the background noise estimation increases the adaptive threshold. These results suggest that RFI with more structured phase that matches the LFM chirp produces more discrete high-powered RFI artifacts. This type of structure results in more false alarms compared to random RFI with incoherent phase structure compared to the LFM chirp. This incoherent RFI results in reduced average SINR which, in turn, reduces the probability of detection. Many
Figure 3.7: Range-Doppler map with a tone sweeping from -10 MHz to 10 MHz baseband every 100 pulses within a CPI. The RFI’s phase progression results in false alarms and missed detection of one target.

Figure 3.8: Range-Doppler map with a tone randomly hopping between -10 MHz to 10 MHz baseband every pulse within a CPI. The RFI’s phase progression results in missed detections on the two weaker targets but few false alarms.
communications and LTE waveforms use OFDM with some phase coding which may present itself as incoherent RFI to an LFM chirp radar waveform [10].

As shown in these simulations, radars coexisting with other emitters result in detection errors and range-Doppler artifacts. This demonstrates a need for spectrum sharing radars and the ability to adapt transmit waveform bandwidth to avoid interference. Combining the cognitive PAC with spectrum sharing allows for robust coexistence. In the previous section, this dissertation defines a constant start $f_s$ and end $f_e$ frequency for a LFM chirp transmit pulse. To achieve coexistence, this work considers adapting $f_s$ and $f_e$ between pulse to maneuver around RFI.

Since the PAC is defined by a spectrum perception method and a waveform action method, we refer to adapting the bandwidth of an LFM chirp as an avoid action method. This work considers adapting pulse-to-pulse or performing intra-CPI adaptation. While intra-CPI adaptation provides agile RFI avoidance, it results in targets becoming modulated and creates ghost targets across Doppler [53, 54].

**Figure 3.9:** Range-Doppler map with the LFM chirp $f_s$ alternating between -35 MHz and -15 MHz baseband and $f_e$ alternating between 5 MHz and 25 MHz baseband every two pulses (according to "Pulse Pattern 6" in the Appendix). This results in detection false alarms due to the target signal becoming modulated.
Consider the same radar simulation with no RFI, except the LFM chirp parameters change every pulse. Let the start frequency $f_s$ alternate between -35 MHz and -15 MHz baseband and the end frequency $f_e$ alternate between 5 MHz and 25 MHz baseband. Figure 3.9 shows that this intra-CPI adaptation results in the target signal becoming modulated. This results weaker copies of the target appearing across Doppler. These ghost targets are detected by CA-CFAR which creates false alarms.

While this dissertation mainly focuses on cognitive RFI avoidance, the effects of intra-CPI adaptation should be noted. Some consideration to these distortion effects will be analyzed in Chapter 8. In the context of an avoid waveform action with a pulsed LFM chirp radar system, the next chapter begins to introduce aspects of spectrum perception and modeling the RF spectrum. The detection and prediction of RFI becomes necessary to inform radar waveform adaptation for spectrum sharing. Through this process, the effects of LFM chirp waveforms and RFI colliding will be minimized.
Sense-Predict-Avoid

In the context of a pulsed radar system, this dissertation considers spectrum sharing that adapts transmit waveform bandwidth for each pulse. The last section described the PAC for radar spectrum sharing including a spectrum perception method and waveform action method. An avoid action refers to adapting LFM chirp frequencies to occupy the widest available contiguous portion of the RF spectrum. In order to decide optimal parameters for avoidance, the PAC must sense and form a perception of the EME. First, the sense stage detects the presence of RFI in the spectral environment. After detecting the RFI, the cognitive radar processes the sensed information to determine available spectrum. This work considers a predict perception method based on a stochastic model of RF activity. Prediction provides advantages over reaction by anticipating changes in the RF spectrum and enables the cognitive radar to store memory of the EME in the form of a descriptive model. Combined spectrum sensing, prediction, and avoidance describes a sense-predict-avoid (SPA) CR strategy for spectrum sharing. This chapter introduces this SPA CR strategy and an implementation on a SDRadar system.

To perform predictive spectrum sharing, the cognitive radar must first monitor and learn RF activity over multiple frequency sub-bands. Once the model is learned, the radar can proceed with monitoring, predicting, and accessing the widest available bandwidth. This spectrum sharing implementation trades off maximizing bandwidth while maintaining a high SINR. While a wider bandwidth improves range resolution, occupying more spectrum increases the chances of interference which reduces SINR and target detectability. The construction of SPA
is described by how the PAC detects RFI for each individual sub-band, formulates the stochastic model describing RF activity, and utilizes the model to predict and occupy the spectrum.

4.1 Spectrum Sensing

To perform spectrum sensing, the radar obtains continual RF spectrum measurements for sensing the presence of interference. These measurements consist of \( N \)-point frequency domain observations denoted by \( Y_{SS}[n] \). The frequency spectrum is partitioned into \( M \) sub-bands to monitor RF activity in each sub-band individually. The start and end indices for each sub-band are described by \( N_s = \{N_{s1}, N_{s2}, ..., N_{sM}\} \) and \( N_e = \{N_{e1}, N_{e2}, ..., N_{eM}\} \) respectively. This system uses energy detection in each sub-band to detect the presence of an unknown deterministic signal in a noisy spectrum \([55,56]\). Energy detection was chosen for low computational complexity and its position as a baseline detector. This low computational complexity is well suited toward real-time implementation on SDR hardware. The SDRadar implementation described in Section 4.4 discusses the implementation of energy detection aided by the Fast Spectral Sensing (FSS) algorithm for dimensionality reduction \([57]\). A binary hypothesis test models each sub-band’s busy and idle state

\[
H_0 : \quad Y_{SS}[n] = N_{f}[n] \quad \text{(Idle State)}
\]

\[
H_1 : \quad Y_{SS}[n] = X[n] + N_{f}[n] \quad \text{(Busy State).}
\]

White noise is denoted by \( N_{f}[n] \) and \( X[n] \) is received interference. Available frequency sub-bands are denoted by an idle state, while a busy sub-band contains RFI which could harm radar performance. To obtain the set of binary channel states \( S = \{S_1, S_2, ..., S_M\} \), energy estimates are computed from \( Y_{SS}[n] \) and thresholded by \( \lambda_D \)

\[
S_i = \sum_{n=N_{s_i}}^{N_{e_i}} |Y_{SS}[n]|^2 \overset{H_1}{\gtrless} \lambda_D \quad \forall i \in \{1, ..., M\}.
\]

This work assumes each sub-band has the same length given by \( N_d = N_{e_i} - N_{s_i} + 1 \). \( \lambda_D \) is calculated using calibrated radar receiver noise power measurements and a set false alarm rate \( P_{FA} = 10^{-4} \) according to \([55]\). Here, the process for spectrum
sensing is another Neyman-Pearson detector similar to the CA-CFAR algorithm presented in Chapter 3. With a sufficiently large $N_d$, the energy estimates for each state are normally distributed which allows for the estimation of the detector’s ROC [55]. Probability of false alarm $P_{FA} = P(H_1|H_0)$ and probability of missed detection $P_{MD} = P(H_0|H_1)$ determine the ROC of the detector. Since the $P_{FA}$ only requires an estimate of the receiver’s noise variance $\sigma^2_W$, selecting a desired $P_{FA}$ determines the detection threshold

$$\lambda_D = \sigma^2_W (Q^{-1}(P_{FA})\sqrt{2N_d + N_d}). \tag{4.3}$$

Here, $Q^{-1}(P_{FA})$ is the inverse of the right-sided Gaussian distribution tail integral. Calibrated noise measurements of the radar’s receiver determines $\sigma^2_W$. This work uses $P_{FA} = 10^{-4}$ since false alarms severely impact the estimation of stochastic model parameters. In this case, missed detections will consist of low power RFI which only has a minor impact on radar processing [15].

### 4.2 Modeling RF Activity

This work uses an alternating renewal process (ARP) to model the random arrival and departure of RFI. In this model, each sub-band alternates between busy and idle states with independently random time intervals between state transitions. Here, we compare a parametric and an empirical model to describe these random busy and idle time distributions. The parametric model assumes a log-normal distribution for each state interval while the empirical model uses survival analysis to estimate the distributions. To accurately estimate these distributions, the system must perform spectrum sensing and maintain memory of RF channel states $S$ over time by performing energy detection on $Y_{SS}[n]$ continually. An example of this is shown for a switching two-tone RFI scenario in Figure 4.1 which shows two interferers alternating states over time. From these observed RFI states in $S$, $M$ sets of busy $B_t$ and idle $I_t$ intervals are counted. These intervals describe the time between state transitions for each sub-band in $S$ and allow the cognitive radar to estimate a collection of cumulative distribution functions (CDFs) for busy $p_B = \{p_{B_1}(t_{B_1}), p_{B_2}(t_{B_2}), ..., p_{B_M}(t_{B_M})\}$ and idle $p_I = \{p_{I_1}(t_{I_1}), p_{I_2}(t_{I_2}), ..., p_{I_M}(t_{I_M})\}$ times for each sub-band. This process of estimating $p_B$ and $p_I$ for both models is
Two-tone RFI Scenario

Two-tone RFI Scenario

Figure 4.1: Example of spectrum sensing over time for a switching two-tone RFI scenario. The left plot shows a single spectrum measurement $Y_{SS}[n]$ with two tones as interference in the 4th and 8th channels. The right plot shows the accumulated $S$ after spectrum sensing where yellow indicates RFI and blue indicates no RFI.

outlined in Algorithm 1 as well as Sections 4.2.1 and 4.2.2. The stochastic RFI model requires probability threshold selection which is discussed further in Section 4.3 and Algorithm 3.

4.2.1 Parametric Model

The parametric model assumes busy and idle time distributions have independent log-normal distributions. Log-normal distributions are widely used in reliability and time-to-event analysis [58]. Since this distribution draws from only positive values, it has been used to model ARPs [59]. Given a sufficient observation interval, the system computes a sample mean estimate $\mu$ and variance estimate $\sigma^2$ for both busy and idle times in all $M$ sub-bands

$$\mu_{B_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} B_{i,j}$$

$$\sigma_{B_i}^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (B_{i,j} - \mu_{B_i})^2.$$ (4.4)

The index of the frequency sub-band is denoted by $i \in \{1, 2, ..., M\}$ and $n_i$ describes the sample size of busy or idle intervals observed in that sub-band. Similarly for
idle times

\[ \mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} I_{i,j} \]
\[ \sigma_i^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (I_{i,j} - \mu_i)^2. \] (4.5)

Log-normal distribution parameters for busy times \( \mu_{L,B_i} \) and \( \sigma_{L,B_i} \) are computed from the sample mean and variance estimates

\[ \mu_{L,B_i} = \ln \left( \frac{\mu_{B_i}^2}{\sigma_{B_i}^2 + \mu_{B_i}^2} \right) \]
\[ \sigma_{L,B_i} = \sqrt{\ln \left( \frac{\sigma_{B_i}^2}{\mu_{B_i}^2} + 1 \right)}. \] (4.6)

Substituting subscript \( B_i \) with \( I_i \) in (4.6) yields the idle distribution parameters \( \mu_{L,I_i} \) and \( \sigma_{L,I_i} \). A visual example of the parameter estimation process is shown in Figure 4.2. Given a sub-band is in a particular state, a CDF yields the probability of the sub-band remaining in that state for some specified time interval. This allows

\[ \text{RF Channel States} \]

\[ \pm \sigma_B \]

\[ \pm \sigma_I \]

\[ \mu_B \]

\[ \mu_I \]

\[ \pm \sigma_B \]

\[ \pm \sigma_I \]

\[ \text{idle} \]

\[ \text{busy} \]

\[ \text{= busy} \]

\[ \text{= idle} \]

**Figure 4.2:** Example of spectrum sensing over time for a switching two-tone RFI scenario with \( M = 10 \) sub-bands. Two interferers alternate states over time in the 4th and 8th sub-bands. The plot depicts the estimate of mean (black) and standard deviation (gray) parameters for busy and idle times.
the log-normal CDF to describe the probability of the $i^{th}$ sub-band remaining in a busy or idle state for $t_{B_i}$ and $t_{I_i}$ timesteps, respectively

$$p_{B_i}(t_{B_i}) = \frac{1}{\sigma_{L,B_i}\sqrt{2\pi}} \int_0^{t_{B_i}} \frac{1}{t_{B_i}} \exp \left( -\frac{(\ln(t) - \mu_{B_i})^2}{2\sigma_{L,B_i}^2} \right) dt$$

$$p_{I_i}(t_{I_i}) = \frac{1}{\sigma_{L,I_i}\sqrt{2\pi}} \int_0^{t_{I_i}} \frac{1}{t_{I_i}} \exp \left( -\frac{(\ln(t) - \mu_{I_i})^2}{2\sigma_{L,I_i}^2} \right) dt. \quad (4.7)$$

In this implementation, we discretize $t_{B_i}$ and $t_{I_i}$ into timesteps where each timestep is $T_0 = 41 \mu s$. This is determined by the SDRadar hardware which continually performs spectrum sensing on $N = 4096$ point blocks of RF data at 100 MSamples/s sampling rate. The length of a timestep $T_0$ is set by the block size of the received data (described in more detail in Section 4.4).

**Algorithm 1: Stochastic Model Estimation**

input : \{Y_{SS,j}[n]\} - set of spectra over all timesteps $j \in \{1, 2, ..., P\}$
$N_s, N_e, \lambda_D$ - sensing parameters

output: $p_{B_i}$ - set of busy time distributions over all sub-bands $i \in \{1, 2, ..., M\}$
$p_{I_i}$ - set of idle time distributions over all sub-bands
\{S_{i,j}\} - set of detected RFI states over all sub-bands and timesteps
$\theta_B, \theta_I$ - busy and idle state prediction thresholds

1) Obtain RFI states $S_{i,j}$ via energy detection (Eqn 4.2)
2) Count busy $B_{t_i}$ and idle time $I_{t_i}$ intervals using $S_{i,j}$
3) Estimate busy $p_{B_i}$ and idle time $p_{I_i}$ distributions $\forall i$:
   if using parametric model then
      i) Compute busy and idle time statistics (Eqn 4.4 and 4.5)
      ii) Compute log-normal CDFs for $p_{B_i}$ and $p_{I_i}$ using statistics (Eqn 4.7)
   else
      iii) Compute CHFs for $p_{B_i}$ and $p_{I_i}$ using $B_{t_i}$ and $I_{t_i}$ (Eqn 4.8)
end
4) Tune prediction thresholds $\theta_B$ and $\theta_I$ using Algorithm 3

4.2.2 Empirical Model

Alternatively, survival analysis could be used to obtain probabilities $p_I$ and $p_B$ without assuming a distribution or estimating model parameters. Survival analysis
is traditionally implemented to analyze statistical properties of the time between events, such as failure in a mechanical system or infant mortality [16]. Having observed a large sample of \( n_i \) busy times \( B_{(i,1)}, B_{(i,2)}, \ldots, B_{(i,n_i)} \) and \( m_i \) idle times \( I_{(i,1)}, I_{(i,2)}, \ldots, I_{(i,m_i)} \) in sub-band \( i \), an empirical cumulative distribution function or cumulative hazard function (CHF) \( H(t) \) can be formed

\[
\begin{align*}
p_I_i(t_{I_i}) &= \sum_{j: I_{(i,j)} \leq t_{I_i}} \frac{1}{n_i - j + 1} \\
p_B_i(t_{B_i}) &= \sum_{j: B_{(i,j)} \leq t_{B_i}} \frac{1}{m_i - j + 1}.
\end{align*}
\] (4.8)

(4.8) uses a form of the Nelson-Aalen estimator (derived in [16]) to approximate each distribution. For the CHF, each distribution is normalized to ensure \( 0 \leq p_{I_i} \leq 1 \) and \( 0 \leq p_{B_i} \leq 1 \).

### 4.3 Stochastic Model-based Prediction

The cognitive radar system’s operation with SPA consists of an initial online learning period where the system observes the RF spectrum to compute the model distributions as described in Algorithm 1. The operator specifies the desired model type (parametric or empirical) to use for learning. During this learning period, no radar transmission occurs. Once learning is complete, new estimates only need to be completed if the RF statistics significantly change. Periodic retraining of the model could offset the impacts of non-stationary statistics [60]. After estimating the stochastic model with thresholds in real-time, the cognitive radar begins periodically predicting RFI according to the model and transmits according to the prediction scheme. During operation, the cognitive radar continues to sense the spectrum with energy detection to obtain \( S \) and track the duration of each sub-band’s state. Figure 4.3 describes this sequence of estimating the stochastic model, tracking RFI states, and performing predictive spectrum sharing. Prior to transmitting a radar pulse, the probability of channel availability \( p_{a_i} \) is computed for all sub-bands \( i \in \{1, 2, \ldots, M\} \),

\[
p_{a_i} = \begin{cases} p_I_i(t_{I_i} + t_0), & S_i = 0 \\ 1 - p_B_i(t_{B_i} + t_0), & S_i = 1. \end{cases}
\] (4.9)
Figure 4.3: Block Diagram describing the spectrum sensing, stochastic model training, and prediction sequence of operations for spectrum sharing cognitive radar. This diagram depicts where Algorithms 1 and 3 are used for model training and Algorithm 2 for prediction.

This work compares both the log-normal CDF and CHF models as alternate modes of operation for calculating $p_{a_i}$. Equation (4.9) uses $p_I(t_I)$ and $p_B(t_B)$ as lookup tables to compute the probability of each sub-band becoming available in the next $t_0$ timesteps. $t_B$ and $t_I$ represent the timesteps sub-band $i$ has spent in the current busy or idle state, respectively. As described in Figure 4.3, continuous spectrum sensing allows the cognitive radar to track $t_B$ and $t_I$. This system performs prediction and determines the next transmit waveform every $t_0$ timesteps before the next PRI ends. $t_0$ describes the waveform adaptation latency in timesteps where $t_0 = N_0T_0 = 451 \mu s$ for the hardware described in Section 4.4. Due to SDRadar hardware latency, $N_0 = 11$ timesteps where a PRI is set by $N_0 - 1 = 10$ timesteps. After computing a set of channel probabilities $p_{a_i}$, the system determines a set of available sub-bands $A = \{A_1, A_2, ..., A_M\}$ by applying a threshold to each $p_{a_i}$:

$$A_i = \begin{cases} p_{a_i} \geq \theta_I, & S_i = 0 \\ p_{a_i} \geq \theta_B, & S_i = 1 \end{cases}$$

(4.10)

$A_i = 1$ describes a sub-band predicted to be unoccupied and available. This work proposes using two separate thresholds $\theta_B$ and $\theta_I$ for each state. These thresholds are tuned to optimize performance at the end of stochastic model training. The cognitive radar selects the widest contiguous cluster of sub-bands in $A$ for its transmit center frequency and bandwidth. The widest contiguous group of sub-bands determines $f_s$ and $f_e$, the start and end frequencies of the transmit waveform. These start and end frequencies allow the system to track the sub-bands to be
accessed in $A_{\text{TX}} = \{A_{\text{TX}_1}, A_{\text{TX}_2}, ..., A_{\text{TX}_M}\}$.

This process of performing prediction and transmit bandwidth selection is described in Algorithm 2. After transmitting the selected pulse, the SDRadar repeats SPA or the spectrum sensing, prediction, and radar transmit cycle as described in the right half of Figure 4.3. Previous work in [44], presents results from this approach with a fixed threshold for both states regardless of the RF environment. Compared to this method, the addition of threshold optimization and unique thresholds for each state seeks to improve performance.

Algorithm 2: Predict and Avoid RFI

**input**: $\{S_i\}$ - Set of current RF channel states $i \in \{1, 2, ..., M\}$

$t_{B_i}, t_{I_i}$ - current busy or idle time

$t_0$ - timesteps to predict ahead before next pulse

$p_{B_i}, p_{I_i}, \theta_B, \theta_I$ - stochastic model distributions and parameters

**output**: $f_1, f_2$ - transmit start and end frequency

$A_{\text{TX}}$ - set of selected TX sub-bands

1) Compute channel availability $p_{a_i} \forall i$ using $S_i$ (Eqn 4.9)
2) Predict available sub-bands $A_i$ by thresholding $p_{a_i}$ (Eqn 4.10)
3) Get $f_1$ and $f_2$ from the widest contiguous available sub-bands in $A$
4) $A_{\text{TX}}$ is the set of sub-bands the radar accesses based on step (3)

After performing stochastic model estimation (described in Section 4.2), $\theta_B$ and $\theta_I$ are tuned to optimize a spectrum sharing performance trade-off as the last stage of the online learning process. By considering the prediction model as a detector, the average training performance can be characterized by type I error (false alarm) rate and type II error (missed prediction) rate. Unlike energy detection, these thresholds are applied to the output likelihood values of the prediction model rather than estimated energy. An analysis of the predictor’s detection error trade-off [61] allows for the optimized tuning of thresholds $\theta_B$ and $\theta_I$. A detection cost function weighs and sums the observed error types to achieve a scalar performance measure [62]. Based on the notion of detection error trade-off or cost, we construct a mapping of thresholds $\theta_B$ and $\theta_I$ to a prediction error cost function for optimizing these thresholds.

The system uses the collection of detected states $S$ from learning $p_{B_i}(t_{B_i})$ and $p_{I_i}(t_{I_i})$ to obtain a set of model probabilities $p_{a_i}$ over time. Applying Equation (4.10) using this set of $p_{a_i}$ produces a set of simulated $A$ over time. Comparing
the set of $A$ to the true detected states $S$ with Boolean logic allows the system to measure the simulated prediction errors using the training data. In the ideal $t_0$ step ahead prediction case, $A_{i,j} = \neg S_{i,j+t_0}$. Spectrum sharing radar performance is measured by collisions (radar interfering with coexisting emitters) and missed opportunities (unused open frequency sub-bands during radar operation). The rate of simulated collisions $C_s$ over $P$ timesteps is given by

$$C_s = \frac{\sum_{j=1}^{P-t_0} \sum_{i=1}^{M} (A_{i,j} \land S_{i,j+t_0})}{\sum_{j=1}^{P-t_0} \sum_{i=1}^{M} S_{i,j+t_0}}.$$  \tag{4.11}

In this case, collisions are equivalent to a prediction missed detection or type II error. The total rate of missed opportunities or prediction type I errors are given by

$$D_s = \frac{\sum_{j=1}^{P-t_0} \sum_{i=1}^{M} (\neg A_{i,j} \land \neg S_{i,j+t_0})}{\sum_{j=1}^{P-t_0} \sum_{i=1}^{M} \neg S_{i,j+t_0}}.$$  \tag{4.12}

The prediction error cost function $\rho$ assigns weighting $0 \leq \alpha \leq 1$ to both error types for achieving some target spectrum sharing trade-off

$$\rho = \alpha C_s + (1 - \alpha) D_s.$$  \tag{4.13}

Given a desired performance weighting $\alpha$ combined with a measured collection of $S$ and $p_a$ over time, the system selects $\theta_B$ and $\theta_I$ by minimizing $\rho$. If $A(\theta_B, \theta_I) : \theta_B \times \theta_I \rightarrow A$, then it follows that $C_s(A(\theta_B, \theta_I)) : A(\theta_B, \theta_I) \rightarrow C_s$ and $D_s(A(\theta_B, \theta_I)) : A(\theta_B, \theta_I) \rightarrow D_s$. Here, the $\times$ operator denotes a Cartesian product. Similar to minimum detection error cost [62], this relationship describes the threshold optimization function as

$$(\theta_B, \theta_I) = \arg \min_{\theta_B, \theta_I} \alpha C_s(A(\theta_B, \theta_I)) + (1 - \alpha) D_s(A(\theta_B, \theta_I)).$$  \tag{4.14}

This system optimizes $\theta_B$ and $\theta_I$ via a grid search. This prediction simulation and grid search process is described by Algorithm 3 in detail. In this SDRadar implementation, the grid search computes $\rho$ by exhaustively testing 100 possible threshold values for $\theta_B$ and $\theta_I$ between 0.05 and 0.99 to satisfy Equation (4.14). Based on previous results in [44], the prediction threshold directly controls a trade-off between collisions and missed opportunities. Since this trade-off is based on a rough performance heuristic, the search size has only minor impacts on performance.
A higher threshold results in less collisions and more missed opportunities. Careful selection of $\alpha$, allows for optimized tuning of these thresholds with consideration for some desired spectrum sharing performance trade-off.

**Algorithm 3:** Threshold Tuning

**input:**

- $\{S_{i,j}\}$ - set of detected RFI states over all sub-bands and timesteps
- $p_B$ - set of busy time distributions over all sub-bands $i \in \{1, 2, ..., M\}$
- $p_I$ - set of idle time distributions over all sub-bands
- $t_0$ - timesteps to predict ahead before next pulse

**output:** $\theta_B, \theta_I$ - busy and idle state prediction thresholds

1) Simulate Algorithm 2 step 1) over time $j \in \{1, 2, ..., P\}$:
   i) Step through $S_{i,j}$ over time $j$ to count $t_{B_i}$ and $t_{I_i}, \forall i$
   ii) Compute channel availability $p_{a_{i,j}}$ using output of step (1i) (Eqn 4.9)

2) Perform grid search of $\theta_B$ and $\theta_I$ to minimize $\rho$ using $p_{a_{i,j}}$ (Eqn. 4.14):
   for $\theta_B = m, \forall m \in \{0.05, 0.06, ..., 0.99\}$ AND $\theta_I = n, \forall n \in \{0.05, 0.06, ..., 0.99\}$
   iii) Test ($\theta_B, \theta_I$), predict availability $A_{i,j}$ by thresholding $p_{a_{i,j}}$ (Eqn. 4.10)
   iv) Compute performance error $\rho$ using $S_{i,j}$ and $A_{i,j}$ (Eqn. 4.13)

v) Select ($\theta_B, \theta_I$) pair which minimized $\rho$

### 4.4 Software Defined Radar (SDRadar) Hardware

The predictive mode cognitive radar was implemented via SPA using a USRP-2944R SDR with a UBX-160 daughterboard containing two receive and two transmit channels (shown in Figure 4.4). The SDR platform is a reconfigurable commercial off-the-shelf device which allows for rapid prototyping and real-time evaluation of cognitive functionality. This SDR samples and performs initial pre-processing of the RF environment. The device sends this RF data to a host controller over a 4x PCIe interface to perform real-time cognitive PAC processing. The real-time host processes include online stochastic model learning, prediction threshold optimization, and combined prediction with waveform generation. This host processing is implemented in LabVIEW 2017. This predictive spectrum sharing scheme builds upon the system architecture for a reactive cognitive radar described
in [63]. The reactive system’s host controller was modified to perform prediction and develop memory of the RF environment.

To perform spectrum sharing, the receiver samples in-phase and quadrature (I and Q) channels at 100 MSamples/s before performing digital down-conversion to baseband frequencies. This allows the system to monitor 100-MHz portions of the RF spectrum for activity. The SDR uses a field-programmable gate array (FPGA) for rapidly performing a fast Fourier Transform (FFT) on input signals to obtain $Y_{SS}[n]$ as per Equation (3.10). During operation, the FPGA performs spectrum sensing on a continuous stream of 4096-sample FFT blocks $Y_{SS}[n]$. An FPGA implementation of FSS [15,57] reduces the dimensionality of the data before energy detection is performed to obtain the occupancy state for each sub-band $S$. Under standard operating conditions, FSS reduces dimensionality by identifying busy and idle sub-bands in a spectrum and grouping each sub-band into a 4-tuple [15,57]. This 4-tuple contains the start and end frequency of the sub-band, the power in the sub-band, and the busy or idle state of the sub-band. Here, we consider a special case of FSS where the start and end frequencies are preset by $N_s$ and $N_e$. Modifying FSS allows for efficient computation of the power within each sub-band to perform energy detection for the SPA architecture. In this SDRadar implementation, FSS reduces the dimensionality of the RF spectrum data from 4096 to $4M$. Later chapters in the dissertation evaluate $M = 10$ and $M = 20$ sub-bands.

After obtaining the RF occupancy states, the data is sent to the host controller to perform prediction or other perception processing. At 100 MSamples/s, each
Figure 4.5: Hardware configuration shows the data flow for the cognitive SDRadar system. Stochastic model estimation and prediction is performed in the RFI Learning & Perception block.

A block or timestep contains a 40.96-µs observation of the RF spectrum. The host performs spectrum sensing and prediction (Figure 4.3) to select a transmit center frequency and bandwidth. These waveform parameters are periodically sent from the host to the SDR, and then the SDR’s FPGA performs transmit waveform generation and controls the timing of the radar PRI. This radar uses a LFM chirp waveform $u(t)$ (defined in Equation (3.1)) for target detection. The LFM chirp bandwidth is characterized by start frequency $f_s$ and end frequency $f_e$. The host chooses $f_s$ and $f_e$ to occupy the widest predicted available bandwidth based on $A_{TX}$ (defined in Algorithm 2). These waveform parameters are determined $t_0$ timesteps before each radar pulse using the stochastic model. This system uses a PRI of 409.6 µs or 10 spectrum sensing timesteps. During radar operation, the system performs prediction and waveform adaptation between each pulse. A visualization of the data flow and processing in the SDRadar hardware is shown in Figure 4.5.

For real-time radar processing, the FPGA performs the initial matched filtering stage of Doppler processing. The FPGA keeps track of waveforms selected by the host and performs matched filters on incoming RF acquisitions containing radar data. With 10 spectrum sensing timesteps in a PRI, each 10th FFT acquisition is matched filtered and sent as radar data to the host. After the host accumulates a full CPI of matched filtered RF acquisitions, the GPU is utilized to perform the 2D-FFT for Doppler processing and detect targets with CA-CFAR (described in Chapter 3). The host GPU uses a Nvidia GeForce GTX 1060 for hardware-accelerated processing. A similar radar processing implementation for a reactive SDRadar
system is introduced in [63]. This dissertation extends the reactive implementation to a cognitive SDRadar which perceives the spectrum via spectral prediction.
Here, we introduce a comprehensive set of RFI test cases and performance metrics to analyze the performance of SPA implemented on the SDRadar according to Chapter 4. These test cases and metrics evaluate the performance of SPA compared to a full bandwidth pulsed radar with no spectrum sharing. The metrics consider impacts on the range profile and spectrum sharing efficiency. The test set includes synthetic RFI with varying levels of predictability and some real measured RF data.

5.1 Performance Metrics

This chapter considers prediction performance metrics such as collision and missed opportunities, in addition to radar metrics post-matched filtering. These metrics apply to any CR strategy using an avoid waveform action. The simulated collisions $C_s$ and simulated missed opportunities $D_s$ from Equations (4.11) and (4.12) describe type I and II prediction errors. For radar operation, this work considers a radar collision $C_r$ and missed opportunity $D_r$ metric based on the widest selected bandwidth $A_{TX(i,k)}$ (Algorithm 2). $i$ describes each sub-band $i \in \{1, 2, ..., M\}$ and $k$ describes the timestep of each radar transmit pulse $k \in \{1t_{PRI}, 2t_{PRI}, ..., Qt_{PRI}\}$ where $t_{PRI}$ is the number of timesteps in a PRI and $Q$ is the number of pulses in a test set. Let $S_{TX(i,k+t)}$ define the RF sub-band states when the predicted waveform
given by $A_{TX(i,k)}$ is transmitted $t_0$ timesteps later. This slight modification of Equation (4.11) yields

\[
C_r = \frac{\sum_{m=1}^{Q} \sum_{i=1}^{M}(A_{TX(i,m*PRI)} \land S_{TX(i,m*PRI+t_0)})}{\sum_{m=1}^{Q} \sum_{i=1}^{M} S_{TX(i,m*PRI+t_0)}}.
\]  

(5.1)

To calculate missed opportunities $D_r$, the system must consider the actual widest available bandwidth at time $k + t_0$. Let this ground truth widest bandwidth be $A_{RX(i,k+t_0)}$. This value can be obtained by counting the widest contiguous cluster of sub-bands where $\neg S_{TX(i,k+t_0)} = 1$ (similar to Algorithm 2 step 3). From $A_{RX(i,k+t_0)}$, the measured radar missed opportunities are described by

\[
D_r = \frac{\sum_{m=1}^{Q} \sum_{i=1}^{M} (\neg A_{TX(i,m*PRI)} \land A_{RX(i,m*PRI+t_0)})}{\sum_{m=1}^{Q} \sum_{i=1}^{M} A_{RX(i,m*PRI+t_0)}}.
\]  

(5.2)

The trade-off between radar collisions $C_r$ and missed opportunities $D_r$ is directly controlled by the $\alpha$ parameter weighting during training (described in Section 4.3). As variability in RF activity increases, this trade-off degrades. Lower collisions correspond to less interference power which improves target SINR while lower missed opportunities corresponds to increased bandwidth utilization. This increase in bandwidth results in finer range resolution while improved SINR results in better target detectability.

For further analysis, this work considers the relationship between spectrum efficiency and radar processing by analyzing range profiles in the presence of interference. Compared to full bandwidth operation, the spectrum sharing radar sacrifices some range resolution to reduce mutual interference with coexisting RF emitters. The impact of this interference on a pulsed radar’s range profile can be modeled via matched filter analysis. In Chapter 3, the impact of RFI on the radar’s matched filter is analyzed. Equation 3.17 shows the contribution of RFI to a range profile $v[t]$ as a linear combination of target signal $v_R[t]$, RFI $v_I[t]$, and noise $v_N[t]$. To evaluate SINR, this work considers the peak target echo power due to $v_R[t]$ after pulse compression as the target signal power. Then, the average interference plus noise power is evaluated as the matched filter response of the $v_I[t] + v_N[t]$ component. Given a target at index $n_t$ with the mainlobe’s respective start and end indices at $n_{t1}$ and $n_{t2}$, the average interference plus noise power is

\[
43
approximated by

\[ P_{IN} = \frac{\left( \sum_{n=0}^{n_{t1}} |v[n]|^2 + \sum_{n=n_{t2}}^{N} |v[n]|^2 \right)}{N - (n_{t2} - n_{t1} + 1)}. \]  \hfill (5.3)

\( N \) describes the number of points in the range profile measurement. Equation (5.3) represents an experimental formulation for the average interference plus noise power \( P_{IN} \) to measure the impact of RFI on the radar range profile. This \( P_{IN} \) combined with the peak target power at \( n_t \) approximates the peak-to-average SINR

\[ SINR = \frac{P_S}{P_{IN}} = \frac{|v[n_t]|^2}{P_{IN}}. \]  \hfill (5.4)

This work evaluates the difference of SINR during spectrum sharing (\( \text{SINR}_P \)) compared to full bandwidth operation (\( \text{SINR}_F \))

\[ G_{\text{SINR}} = \frac{\text{SINR}_P}{\text{SINR}_F}. \]  \hfill (5.5)

\( G_{\text{SINR}} \) describes the SINR gain from spectrum sharing. While target power \( P_S \) should be similar for both scenarios, spectrum sharing reduces interference power \( P_{IN} \) which increases SINR. Target detection requires some minimum detectable SINR \( \text{SINR}_{\text{min}} \) to reliably detect a signal

\[ \text{SINR}_{\text{min}} = \frac{P_{\text{min}}}{P_{IN}}. \]  \hfill (5.6)

This \( \text{SINR}_{\text{min}} \) relates to some minimum detectable signal power \( P_{\text{min}} \) to detect a target [64].

For a target with constant radar cross-section (RCS) \( \sigma_R \), the basic radar range equation relates this minimum detectable signal \( P_{\text{min}} \) to some maximum detection range \( R_D \) [64]

\[ P_{\text{min}} = \frac{P_T G_T G_R \lambda^2 \sigma_R}{(4\pi)^3 R_D^4}. \]  \hfill (5.7)

\( P_T \) describes the transmitted power, \( G_T \) and \( G_R \) are transmit and receive antenna gains, while \( \lambda \) is the wavelength of the carrier frequency. When performing spectrum sharing, the system reduces the interference power \( P_{IN} \) which improves \( R_D \) compared to full bandwidth operation. This improves the ability of the radar to detect weak targets over longer ranges while coexisting with potential RFI sources. To evaluate
the impact of $P_{IN}$ on $R_D$, this work considers the maximum detection range improvement as a ratio of $R_{D,P}$ and $R_{D,F}$,

$$\frac{R_{D,P}}{R_{D,F}} = \sqrt{\frac{P_{\text{min,F}}}{P_{\text{min,P}}}} = \sqrt{\frac{P_{IN,F}}{P_{IN,P}}},$$ (5.8)

$$\gamma = \left( \frac{R_{D,P} - R_{D,F}}{R_{D,F}} \right) (100)$$

$$= \left( \sqrt{\frac{P_{IN,F}}{P_{IN,P}}} - 1 \right) (100).$$ (5.9)

$R_{D,P}$ is the maximum detection range after spectrum sharing while $R_{D,F}$ describes the maximum detection range from full bandwidth operation (no spectrum sharing). Evaluating the radar range equation (5.7) in terms of minimum detectable SINR, Equation (5.6) yields Equation (5.8). This describes the direct relationship between maximum detection range improving from reduced interference power $P_{IN}$. During full bandwidth operation, the lack of spectrum sharing results in higher interference plus noise power $P_{IN,F}$ compared to the sharing case $P_{IN,P}$. The resultant percentage of maximum detection range improvement is described by $\gamma$ in Equation (5.9). Values from Equations (5.5) and (5.9) are computed experimentally using range profiles containing a target with a controlled known range index $n_t$. This work evaluates the average cognitive radar performance in terms of spectrum efficiency metrics such as collisions $C_r$ and missed opportunities $D_r$, as well as radar metrics including SINR gain $G_{\text{SINR}}$ and maximum detection range improvement $\gamma$. These presented metrics apply to spectrum sharing radars using an avoid waveform action for their cognitive PAC. Modifications to these metrics should be made for other waveform action types.

In addition to spectrum efficiency and radar performance metrics, we analyze the number of waveform adaptations which occur during radar operation. The count of waveform adaptations describes resulting predictor behavior with respect to prediction parameters (such as thresholds) and RFI properties. This metric may explain sources of error such as a low number of waveform adaptations in the presence of rapidly changing RFI resulting in high missed opportunities. Waveform adaptations describe the cognitive radar’s behavior not captured by errors in spectrum sharing.
To test and evaluate predictive spectrum sharing, a vector signal transceiver (VST) acts as an arbitrary waveform generator to transmit RFI. The SDR and VST transmitters feed into a combiner, which is then received by the SDR (Figure 5.1). These received signals are processed according to Figure 4.3 to carry out cognitive spectrum sharing. The cognitive radar estimates the stochastic model using 410 ms of RF measurement before prediction and radar operation is performed on the subsequent 410 ms time period. SPA partitions the spectrum into ten evenly spaced 10 MHz sub-bands for detecting RFI and tracking statistics. Since the SDR and VST are capable of arbitrary center frequencies from 10 MHz to 6 GHz, these experiments were conducted at 1 GHz center frequency and 100 MSamples/s sampling rate. A set of test RFI spectra consist of synthetic and real measured signal datasets.

The synthetic RFI dataset includes time-varying single-tone and two-tone signals which switch on and off over time. The single-tone signals evaluate baseline prediction functionality while the two-tone cases are included to verify similar
performance for multiple signals. These sinusoids are controlled to randomly switch states according to the ARP described in Chapter 4. Each tone is characterized by the on-time mean $\mu_B$ and standard deviation $\sigma_B$ as well as its off-time mean $\mu_I$ and standard deviation $\sigma_I$.

The single-tone RFI signals consist of a sinusoid occupying a single sub-band with a deterministic on-time ($\sigma_B = 0$) and a random off-time ($\sigma_I \neq 0$). This choice for $\sigma_B$ and $\sigma_I$ emulates a communication system with a fixed length on-time burst and random time between requests or transmissions of data. Each switching tone scenario exists in a 50% traffic intensity case where the on-time and off-time are equal $\mu_B = \mu_I$. The off-time is generated from a log-normal random variable characterized by $\mu_{L,I}$ and $\sigma_{L,I}$, similar to the parametric model described in Section 4.2.1. This work considers the coefficient of variation as a measure of variability for the off-time

$$c_{VI} = \frac{\sigma_I}{\mu_I}.$$  \hspace{1cm} (5.10)

To evaluate prediction performance as variability increases, the cognitive radar is tested on a set of single-tone signals with gradually increasing $c_{VI}$ between 0 and 1. Similarly, the RFI set includes a variety of mean on-times to test this parameter’s effect on prediction.

## 5.2.1 Synthetic Switching Tones

For each $c_{VI}$ value, the single-tone scenarios consider 3 mean on-times $\mu_B$: 410 $\mu$s, 2.05 ms, and 4.1 ms. Different length $\mu_B$ evaluates predictor performance as the RFI transitions states more rapidly. For these tests, we consider performance as a function of $c_{VI}$ and $\mu_B$ using metrics such as the number of waveform adaptations paired with SINR gain $G_{\text{SINR}}$ and the corresponding maximum detection range improvement $\gamma$.

For the two-tone RFI tests, the RF spectrum consists of two sinusoids occupying a single sub-band with $\mu_B = \mu_I$ and a deterministic on-time $\sigma_B = 0$. Figure 5.2 shows examples of these RFI scenarios during cognitive radar operation where the horizontal lines are the radar pulse. A few combinations of $c_{VI}$ and $\mu_B$ for the pair of tones are considered for evaluation. In addition, spectrum efficiency metrics such as collisions $C_r$ and missed opportunities $D_r$ are evaluated. The VST generates these signals in real-time with random non-repeating off-times according to the
Figure 5.2: Spectrograms from cognitive radar operation for (a) the single-tone RFI scenario and (b) the two-tone RFI scenario. The single-tone RFI scenario consists of a single sinusoid at $-25$ MHz baseband while the two-tone scenario includes a sinusoid at $-15$ MHz and 25 MHz baseband, respectively. The horizontal lines shown are the radar’s LFM chirp waveforms avoiding the RFI.

5.2.2 Synthetic Swept Tone & Random Frequency Hopper

The swept tone and random frequency hopper RFI scenarios evaluate the ability of the system to predict signals moving between different frequency sub-bands. Swept tone RFI consists of a deterministic pattern where a tone continuously sweeps from 45 MHz baseband to $-45$ MHz baseband at 10-MHz intervals. Baseband frequency refers to spectrum measurements after digital down-conversion from some arbitrary carrier frequency. Similar to the single-tone scenarios, the test set includes dwell times of 410 µs, 2.05 ms, and 4.1 ms. Random frequency hopping RFI consists of a tone hopping randomly between the 10 frequency sub-bands during operation. This random hopping set also considers dwell times of 410 µs, 2.05 ms, and 4.1 ms. Figure 5.3 shows examples of these synthetic test scenarios with LFM radar pulses avoiding the RFI shown by horizontal lines.
Figure 5.3: Spectrograms from cognitive radar operation for (a) swept tone RFI with a 410-µs dwell time and (b) frequency hopping RFI with a 4.1-ms dwell time. The horizontal lines shown are the radar’s LFM chirp waveforms avoiding the RFI.

Figure 5.4: Spectrograms from cognitive radar operation with (a) measured RF data from 1750 MHz center frequency and (b) 800 MHz center frequency. The horizontal lines shown are the radar’s LFM chirp waveforms avoiding the RFI.

5.2.3 Real Interference

For emulating real RFI scenarios, a set of real measured RF data is employed to evaluate cognitive radar performance. Real data captures were collected using a USRP X310 SDR measuring continuous time-varying 100-MHz portions of the RF spectrum in State College, Pennsylvania.
Figure 5.5: Spectrograms from cognitive radar operation with (a) measured RF data from 1850 MHz center frequency and (b) 2450 MHz center frequency. The horizontal lines shown are the radar’s LFM chirp waveforms avoiding the RFI.

Figure 5.6: Spectrograms from cognitive radar operation with (a) measured RF data from 1950 MHz center frequency and (b) 2350 MHz center frequency. The horizontal lines shown are the radar’s LFM chirp waveforms avoiding the RFI.
These measurements are replayed on the VST (according to Figure 5.1) with the same model estimation and testing times (410 ms each) as the synthetic RFI scenarios. The real RF data consists of various cellular, LTE, and WiFi signals measured at the following center frequencies: 800 MHz, 1750 MHz, 1850 MHz, 1950 MHz, 2350 MHz, and 2450 MHz. Figure 5.4 show examples of the real RF environments with LFM radar pulses (shown by horizontal lines) avoiding the LTE uplink signals at 1750 MHz and cellular emissions at 800 MHz center frequency. Figure 5.5 shows LTE uplink at 1850 MHz and the ISM band at 2450 MHz center frequencies which includes WiFi and bluetooth waveforms from the measurement. Figure 5.6 shows LTE downlink and some uplink at 1950 MHz and with some light LTE activity at 2350 MHz center frequencies.

5.3 Real-time Experiments

To demonstrate the impact of predictive spectrum sharing, this cognitive radar system was implemented according to Chapter 4. Using a VST to emulate an RFI environment in controlled conditions (Figure 5.1), we evaluate metrics such as number of waveform adaptations, collision rate $C_r$, missed opportunity rate $D_r$, SINR gain $G_{SINR}$, and the corresponding maximum detection range improvement $\gamma$.

The system obtains a continuous stream of RF spectrum measurements with 40.96 µs long timesteps per observation. The stochastic model is estimated on 410 ms of RF measurement and then performs radar operation (according to Chapter 4) on the next 410 ms. Using a PRI of 410 µs, this work evaluates average performance over 1000 range profiles within the 410-ms test interval. While spectrum sharing and radar operation occurs in real-time, the system computes performance metrics after radar operation is complete.

5.3.1 Synthetic RFI

5.3.1.1 Random Single-Tone

For random single-tone interference described in Section 5.2.1, Figure 5.7 shows the parametric log-normal CDF model performance and Figure 5.8 shows the empirical CHF model performance. The prediction threshold tuning weight was set to $\alpha = 0.95$ to heavily favor avoiding collisions and improving radar performance.
Figure 5.7: Radar performance for the parametric log-normal CDF model for the single tone set as variation increases. As shown in Figure 5.8c, dashed lines represent constant threshold results while solid lines represent results using optimized thresholds. (a) Processing gain (b) Number of Waveform Adaptations

Figure 5.8: Radar performance for the empirical CHF model for the single tone set as variation increases. As shown in (c), dashed lines represent constant threshold results while solid lines represent results using optimized thresholds. (a) Processing Gain (b) Number of Waveform Adaptations
The results compare using a constant threshold $\theta_B = \theta_I = 0.5$ to using threshold optimization (legend given Figure 5.8c). Three collections of on-times $\mu_B$ were evaluated for this set at 410 µs, 2.05 ms, and 4.1 ms (shown in legend Figure 5.8c). Figure 5.7a and 5.8a show the processing gain from predictive spectrum sharing in terms of $\gamma$ and $G_{\text{SINR}}$ on the left and right y-axes, respectively. For both models, the optimized threshold yields higher processing gain than the constant threshold case (Figure 5.7a and Figure 5.8a). Higher processing gain corresponds to fewer collisions during spectrum sharing. While the constant threshold results degrade as variability $c_{V,I}$ increases, the optimized threshold maintains and even shows slight processing gain improvement with higher variability.

Figure 5.7b and Figure 5.8b show the number of waveform adaptations or number of times the system adapted center frequency or bandwidth between pulses. As $c_{V,I}$ increases, the system has less waveform adaptations, especially for the optimized threshold log-normal CDF cases (Figure 5.7b). With a higher $c_{V,I}$, the environment often has lower probability of availability based on the stochastic model. This causes the optimized threshold approach with a high $\alpha = 0.95$ to tend towards less waveforms adaptations and, in turn, less collisions. This explains the reduction in waveform adaptations as SINR gain or collision rate improves. A reduction in waveform adaptations to avoid interference corresponds to an increase in missed opportunity rate $D_r$. For optimized threshold selection, the choice of $\alpha = 0.95$ favors lower collisions and results in higher missed opportunities than the constant threshold case due to an inherent trade-off.

Comparing the log-normal CDF to the CHF model, the CHF results (Figure 5.8) were highly variable in terms of $\gamma$ for the constant threshold case. Since the RFI is randomly generated, an empirical model may yield different distributions between runs. An adaptive threshold appears to compensate for this slight difference in CHFs to maintain the performance trade-off. This variation in results is particularly pronounced for RFI with lower $c_{V,I}$. A lower $c_{V,I}$ results in less diverse off-time interval values which makes estimating the hazard function more sensitive to outliers.

Considering three sets of on-times, the predictive sharing scheme’s performance in terms of $\gamma$ degraded as the mean on-time $\mu_B$ grew shorter. This effect of $\mu_B$ on performance is reduced when using an optimized threshold compared to a constant prediction threshold. A shorter mean on-time resulted in more waveform
adaptations which has been shown to increase potential for distortion during Doppler processing [53]. In general, these results suggest predictive spectrum sharing performance degrades as the coefficient of variation $c_{V,1}$ increases and mean on-time decreases. Despite this trend, the performance reduction can be mitigated with threshold optimization.

5.3.1.2 Random Two-Tone

Random two-tone RFI scenarios evaluate performance for two sinusoids with 5 different combinations of off-time statistics. The VST randomly switches two sinusoidal signals at $-15$ MHz and $25$ MHz (baseband frequency with some arbitrary carrier frequency) on and off. Both of these signals have fixed on-times at 4.1 ms.

![Figure 5.9](image)

**Figure 5.9:** Performance results for both parametric and empirical stochastic models in each two-tone RFI scenario. Table 5.1 describes each two-tone case. (a) Processing Gain (b) Collision Rate (c) Missed Opportunity Rate
Table 5.1: Coefficient of variation for each signal in the two-tone RFI scenario. The first column specifies the frequency of each sinusoid. The −15-MHz and 25-MHz signal have fixed on-times of 4.1 ms and 2.05 ms, respectively. "C1" through "C5" refers each case number.

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>−15 MHz $c_{v,i}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.225</td>
<td>0.45</td>
</tr>
<tr>
<td>25 MHz $c_{v,i}$</td>
<td>0.32</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.225</td>
</tr>
</tbody>
</table>

and 2.05 ms, respectively, with the off-time statistics varying based on some desired $c_{v,i}$ (see Table 5.1 for details). Optimized prediction threshold selection ($\alpha = 0.95$) is used for the results in this section. Case C1 is deterministic over time (for both tones $c_{v,i} = 0$) and shows lower missed opportunity rates (Figure 5.9c) for the log-normal CDF model. As the average $c_{v,i}$ increases, the missed opportunity rate rises for the CDF model. On the other hand, the CHF model shows some improvement at moderate $c_{v,i}$ levels before degrading at higher $c_{v,i}$. As discussed in section 5.3.1.1, this effect is caused by the CHF being sensitive to outliers [65] present in data with a low $c_{v,i}$. While both models have consistent collision avoidance and radar processing gain (Figure 5.9a and 5.9b), the log-normal CDF approach achieves largely better missed opportunity rates and only marginally worse collision rates compared to the CHF. Similar to the single-tone results, the CDF model achieves better overall spectrum sharing efficiency for deterministic switching RFI patterns.

5.3.2 Synthetic Swept Tone & Random Frequency Hopper

This section presents performance for interference which migrates frequency sub-bands while persisting over time. The two groups of such scenarios include swept tone RFI and random frequency hopping RFI (described in Section 5.2.2). Performance is evaluated using optimized prediction threshold selection with the interference cases defined in Table 5.2. Figure 5.10a shows consistent radar processing gain with increasing performance for longer dwell times.
Table 5.2: The dwell time for each RFI scenario presented in Figure 5.10 is shown. 'S1' through 'S3' refers to swept tone cases while 'H1' through 'H3' refers to a random frequency hopper cases.

<table>
<thead>
<tr>
<th></th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dwell Time</td>
<td>410 µs</td>
<td>2.05 ms</td>
<td>4.1 ms</td>
<td>410 µs</td>
<td>2.05 ms</td>
<td>4.1 ms</td>
</tr>
</tbody>
</table>

Figure 5.10: Performance results for both parametric and empirical stochastic models in the swept tone and random frequency hopper scenarios. 'S#' refers to a swept tone case while 'H#' refers to a random frequency hopper case. Table 5.2 describes the dwell time for each case. (a) Processing Gain (b) Collision Rate (c) Missed Opportunity Rate

Similarly, Figure 5.10b shows lower collisions for longer dwell times. A similar trend with mean on-time and processing gain was observed for the single-tone scenarios in Section 5.3.1.1. The log-normal CDF model achieves lower collisions
in the deterministic swept tone cases. For these swept tone scenarios, the system maintained a low missed opportunity rate which corresponds to a high bandwidth utilization (Figure 5.10c). In the random hopping cases, the radar maintained radar performance at the cost of a high missed opportunity rate. The choice of $\alpha = 0.95$ for threshold selection results in the system reducing bandwidth utilization to maintain a low collision rate. Across all synthetic RFI scenarios, the performance decreases with larger variation (high $c_{V_1}$) and short mean time intervals (low $\mu_B$).

### 5.3.3 Real Interference

To demonstrate the feasibility of this approach on real-world RF environments, we evaluate performance on the set of measured RF data as described in Section 5.2.3. With a variety of RF environments, this scheme achieved maximum detection range improvements of $\gamma \sim 20\%$ and greater (Figure 5.11a). Processing gain is dependent on the RFI power and the RFI’s correlation with the transmitted LFM chirp (given by Equation (5.5)). The 1950-MHz and 2350-MHz RF cases contained higher power signals which were avoided by the radar to achieve high processing gains around 100% and 40%. Figure 5.11b shows the system maintaining a $C_r < 0.25$. Similar to the synthetic RFI cases, the prediction threshold constant of $\alpha = 0.95$ results in increased $D_r$ (Figure 5.11c) to minimize collisions.

High missed opportunity rates for the 2350-MHz and 2450-MHz cases (Figure 5.11c) are the result of highly varying RF activity over the entire bandwidth. These frequency ranges are subject to a high volume of RF traffic from WiFi devices. Most of the remaining real RFI cases contain LTE uplink and downlink signals as interference. LTE uplink signals typically have a deterministic on-time with a varying off-time, similar to the synthetic single-tone scenarios. Lower missed opportunity and collision rates at the 1850-MHz and 1950-MHz bands are associated with less variation in the RF signals over time.

Comparing the CDF and CHF models, they have relatively comparable performance in terms of detection range improvement. The collision and missed opportunity rates for each model vary depending on the content of the RF spectrum. The 1950 MHz case contains highly congested cellular downlink signals with fewer opportunities for spectrum access. The CDF compensates for this by restricting adaptation which reduces the missed opportunity rates. In contrast, the
CHF better accesses available spectrum at the cost of high collisions. The 1850 MHz case contains low variation LTE uplink with occasional outliers. These occasional outliers worsen the accuracy of the CHF model which results in significantly increased collisions. The 2450 MHz case contains highly random RFI with higher spectrum occupancies. In this case, the CHF models the variations better than the CDF which reduces the collision rate at the cost of increased missed opportunities. The CDF achieves better model accuracy for lower variation RFI cases while the CHF shows comparably improved performance with high variation.

![Comparative Processing Gain](image1)

![Comparative Collision Rate](image2)

![Comparative Missed Opportunity Rate](image3)

**Figure 5.11:** Performance results for real measured RF data captured at 6 different center frequencies. (a) Processing Gain (b) Collision Rate (c) Missed Opportunity Rate
Alternative Cognitive Radar Strategies

In our cognitive SDRadar implementation, SPA presents one CR strategy for spectrum sharing. SPA describes the components of the cognitive PAC with a *predict* spectrum perception method and an *avoid* waveform action. Predict refers to using a stochastic model for predicting the spectrum based on the sensed RF activity (described in Chapter 4). Avoid refers to occupying the widest available contiguous bandwidth within the RF spectrum. Previous chapters consider avoid with a pulsed LFM chirp waveform. Others have developed alternative CR strategies to SPA with real-time SDRadar implementations on the same USRP platform. This chapter compares some of these alternative CR strategies to SPA and even combines them in various modes.

As shown in Chapter 5, SPA performs well in EMEs with lower variability over time. Based on these results, predictive spectrum sharing provides advantages over reactive techniques when the EME is predictable. The results establish a relationship where a *react* perception method may become more desirable as variability increases compared to prediction. While this provides some intuition to compare SPA to a sense-react-avoid (SRA) CR strategy [15], additional predictive DSA models may be considered as alternatives to SPA. Others have implemented a reinforcement learning model for predictive spectrum sharing on the SDRadar in the form of a Markov Decision Process (MDP) [41,66]. Here, the implementation using a MDP for spectral perception along with an avoid waveform action is referred to as sense-learn-avoid (SLA). The next section will compare these predictive CR strategy implementations (SPA and SLA) with some real-time RFI scenarios. This
will provide further distinction between the performance of stochastic models (SPA) and an MDP for prediction. Since some intuition exists between SRA and predictive CR strategies in general, an analysis including SRA will later be presented as a merged metacognition implementation in Chapter 7.

In addition to perception methods, this chapter will introduce alternatives to avoidance as a waveform action. Another cognitive SDRadar implementation was developed that employed pulsed FM noise waveforms with notching for RFI avoidance [67]. Instead of avoiding the widest contiguous portion of the RF spectrum, the radar occupies the entire operating bandwidth with notches placed at RFI locations. The work in [67] expands the SRA implementation to develop a sense-react-notch (SRN) CR strategy. This chapter seeks to merge the predict perception method with a notch action to produce a sense-predict-notch (SPN) method. The efficacy of this SPN cognitive radar implementation is confirmed through real-time experiments with a variety of RFI scenarios.

6.1 Reinforcement Learning for Spectrum Sharing

Here, a reinforcement learning-based alternative to the spectral perception stage of the PAC is introduced. SLA is composed of a MDP model presented in [68] to predict the RF environment and optimize future transmit waveforms for a spectrum sharing cognitive tracking radar. This method optimizes the radar waveform to mitigate mutual interference between the radar and coexisting communications systems. The MDP models possible RF system decisions as a finite state machine and learns the optimal sequence of decisions based on prior observed training data. The spectrum sharing components of this MDP model were implemented on the SDRadar as SLA in [41]. This section introduces the theoretical construction of the model and details of the SDRadar implementation.

The MDP approach applies an autonomous state machine to identify future available RF sub-bands. MDPs are desirable for developing a system or agent which operates for long periods of time in uncertain environments [69]. In this model, the environment is represented by a set of states and possible actions to be performed within this environment. The environment along with the corresponding states and
actions are defined by the RF spectrum which contains other emitters. While the MDP may model tracking radar parameters such as future target location, this work considers the problem of avoiding RFI in a congested spectral environment. Reinforcement learning allows the agent, in this case the radar, to take actions (select frequency sub-bands) in certain states and measure a reward as feedback. Here, the reward corresponds to the SINR of the radar and the number of frequency bands or bandwidth used. This state selection and reward feedback allows the agent to learn the optimal action to mitigate RFI in every state or scenario. The agent optimizes a reward function over time which predicts the optimal combinations of RF channels to transmit within.

A MDP is specified by the following tuple: $< S, A, T, R, \gamma, \pi^* >$. $S$ is the set of all possible states in the model defined by the possible locations of RFI in the spectrum. The action space $A$ is the set of all actions that can be taken by the system [70] or, in this case, the radar to adapt its transmit waveform. Possible actions taken by the radar consist of selecting sub-bands where the radar will transmit an LFM chirp. The radar can operate only in contiguous bands and the number of actions the radar can take has been shown to be $N_a = N(N + 1)/2$ [68] where $N$ is the number of frequency sub-bands. The transition probability function $T(s, a, s')$ defines the probability that an agent in state $s \in S$ will transition to another state $s' \in S$ when taking action $a \in A$. In this case, the transition function describes the probability of multiple channels changing states between time-steps. The reward function $R(s, a, s')$ describes the average reward accumulated by the agent by performing action $a$ while in state $s$ and transitioning to state $s'$. This function drives the behavior of the system or which radar transmit waveforms are deemed optimal. This implementation maximizes rewards for high SINR and high bandwidths in order to balance range resolution and target detectability. $\gamma$ defines the discount factor which weighs future rewards and immediate rewards [68]. $\pi^*$ describes the optimal policy determined by measuring rewards associated with all observed state transitions. The optimal policy is given by

$$\pi^* = \arg \max_{\pi} E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \right].$$  \hspace{1cm} (6.1)

An example of a Markov Decision Process is shown in Figure 6.1. The $s_1$, $s_2$, $s_3$ describe states with the transition probabilities shown on the arrows. The
transition probabilities denote the probability of arriving in new state $s_0$ when action $a$ is performed in state $s$. When the agent arrives in a new state, it receives a reward ($R_1$, $R_2$, $R_3$ in Figure 6.1) depending on the action that was taken. A reward can be positive or negative, but it must be bounded \([69]\). Determining the optimal policy requires the random selection of some initial policy $\pi_0$ and then iterative improvement or adaptation of this policy. The iterative process repeatedly computes the reward or utility of each policy and then adapts the policy to maximize utility until $\pi^*$ is achieved. The utility function $U(s)$ dictates total reward given by a sequence of states or some policy $\pi_i$,

$$U(s) = E \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t) \middle| \pi_i \right]. \quad (6.2)$$

During iterative policy adaptation $U(s)$ is used as input to compute the next policy $\pi_{i+1}$,

$$\pi_{i+1}(s) = \arg \max_a \sum_{s'} T(s, s', a) U(s'). \quad (6.3)$$

Equation (6.3) uses the utility function $U(s)$ to choose the action $a$ for the current state $s$ to maximize the expected utility of the subsequent predicted state $s'$. This expected next state $s'$ should correspond with the predicted RFI states. The predicted action or transmit waveform is based on the reward structure which trades off maximizing bandwidth and SINR to avoid future interference.
6.2 Comparing to Sense-Learn-Avoid

In the SDRadar implementation with SLA, the same USRP-2944R SDR platform was used while following the architecture laid out in Chapter 5. While the SPA implementation considered ten evenly spaced 10 MHz sub-bands for sensing RFI, SLA uses five 20 MHz sub-bands for sensing RFI and constructing the MDP model. This results in an action space of $N_a = 15$ continuous bandwidths for avoidance with an LFM chirp. The reduction in sub-bands was required to allow the SDRadar to train the MDP model in real-time. The state space with 10 sub-bands takes significantly longer for the MDP to complete policy iteration and converge to an optimal $\pi^*$. Given some number of sub-bands, the cognitive radar performs spectrum sensing (as described in Chapter 4) to determine the current spectral states $s$. As each set of states $s$ is determined, the radar predicts the next future set of state $s'$ and the action or waveform to be transmitted by the radar $a$. An initial random policy $\pi_0$ is selected and the radar iteratively optimizes the action set based on the process described in Section 6.1. By predicting future states $s'$ based on the policy for the current state, the radar infers the behavior of future interference. The MDP approach has the benefit of performing either online or offline learning. Online learning refers to acting or transmitting during training while offline learning optimizes the policy without actively transmitting. The stochastic model-based implementation is unable to act or transmit while training distribution parameters. Since each combination of RF sub-band states is an individual MDP state, performing this processing becomes memory and resource intensive for a large number of sub-bands. The stochastic-based prediction approach implements the probability distributions as look-up tables which reduces processing resources.

Here, we present experiments comparing SLA and SPA. Both follow a similar processing architecture and experiment setup as the one described in Chapter 5. Each experiment runs for 0.82 s, with offline training performed on the first 0.41 s and active radar operation paired with prediction in the second 0.41 s. During offline learning, the MDP learns the optimal policy by selecting random actions throughout and observing the reward. These experiments were performed before prediction threshold optimization for SPA was developed (described in Section 4.3). In lieu of automatic threshold tuning, these results use hand-tuned thresholds where
Table 6.1: Coefficient of variation for each signal in the two-tone RFI scenario. The first column specifies the frequency of each sinusoid. The $-15$-MHz and $25$-MHz signal have fixed on-times of $4.1$ ms and $2.05$ ms, respectively. "C1" through "C5" refers each case number.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$-15$ MHz $c_{V,1}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.225</td>
<td>0.45</td>
</tr>
<tr>
<td>$25$ MHz $c_{V,1}$</td>
<td>0</td>
<td>0.32</td>
<td>0.45</td>
<td>0.45</td>
<td>0.225</td>
</tr>
</tbody>
</table>

the idle and busy time thresholds are equivalent $\theta_I = \theta_B$. Here, the prediction threshold is defined by $\theta$.

Similar to prior experiments, a VST is used to generate real RFI and feeds into a combiner with the radar’s transmitter (see Chapter 5). The test RFI scenarios are generated by the VST looping 0.41 s recordings of RFI data. The test scenarios of interest include synthetic interference in the form of randomly switching sinusoids in time. This includes randomized two-tone dataset consisting of two randomly switching sinusoids at $-20$ MHz and $15$ MHz baseband (see Section 5.2.1). Both had fixed deterministic on times at $100$ FFT acquisitions or $4.1$ ms for the $-20$ MHz tone and $50$ FFT acquisitions or $2.05$ ms for the $15$ MHz tone. The off times were based on a log-normal distribution with $4.1$ ms and $2.05$ ms mean durations and different standard deviations respectively. The off time parameters are described in detail in Table 6.1 (repeated from Section 5.2.1). Additionally, the approaches are tested on a set of 4 real measured RFI spectra collected over 0.41 s. These measurements were collected using the SDR over a 100-MHz bandwidth in a rural environment RF spectrum in State College, Pennsylvania. The realistic test spectra include 2 measurements at 950 MHz center frequency, 1 measurement at 2350 MHz, and 1 measurement at 2450 MHz (see Section 5.2.3).

Each approach was evaluated in terms of collision rate, missed opportunity rate, and number of waveform adaptations (see Section 5.1). Collisions refer to transmitting in channels with interference present. A higher collision rate corresponds with a lower SINR during radar processing and therefore lower target detectability. Missed opportunities describe the system avoiding available bandwidth for transmission. High missed opportunity rates correspond to lower bandwidth utilization which worsens radar range resolution. Poor range res ultion may cause two closely spaced targets to be indistinguishable to the system. Waveform adaptations refers to the
number of times the systems adapts the transmit center frequency or bandwidth during operation. Since range-Doppler analysis requires coherently processing over a large number of pulses, more waveform adaptions may result in artifacts which reduce Doppler SINR and target detectability [53].

For the randomized two-tone scenario, Figures 6.2 and 6.3 show the performance of the stochastic-model based prediction using a log-normal CDF and cumulative hazard function in addition to the MDP approach. For these tests, fixed thresholds were used for SPA where the CDF threshold was $\theta = 0.2$ the hazard function threshold was $\theta = 0.3$. These thresholds were manually tuned to the two-tone scenario which should mimic the behavior of threshold optimization. For most of the cases given by Table 6.1, the stochastic model-based approaches outperform the MDP. Case 4 consisted of higher variation on the shortest duration tone. Since this creates the highest variability case, the collision rate for the MDP worsens and the missed opportunity rates for all approaches degrade.

Generally, the MDP adapts its waveform less frequently than the stochastic model-based approaches which may benefit Doppler processing in a radar (Figure 6.3). Since the MDP weighs SINR in deciding optimal policy, the system tolerates collisions with low SINR penalties more than the other models. For the single tone case described in Section 5.2.1, the MDP simply avoided transmitting where the

![Figure 6.2](image)

**Figure 6.2:** SLA collision and missed opportunity rates for the synthetic RFI scenarios. Cases defined in Table 6.1.
Figure 6.3: Number of waveform adaptations for the synthetic two-tone cases described in Table 6.1. Waveform adaptations refers to the number of times the transmit center frequency or bandwidth changes for the LFM chirp waveform. Cases refined in Table 6.1.

switching sinusoid was located while the stochastic approaches would coexist with the RFI. This behavior for intermittent interference is likely caused by the MDP reward function accounting for a large SINR degradation associated with colliding with the tone. Implementing multiple memory states could improve the MDP’s performance in higher variance cases, but requires offline training due to the larger state space.

For testing real RFI scenarios, the stochastic prediction threshold $\theta$ was retuned for each individual case to trade-off collisions and missed opportunities. Once again this manual tuning mimics the behavior of the threshold optimization discussed in Section 4.3. The RFI statistics (mean on/off durations and variances) greatly affect the optimal threshold for the stochastic model. Since the MDP is non-parameteric, the model inherently optimizes its decisions during training. The real RFI scenarios (shown in Figures 6.4 and 6.5), show overall MDP performance improvement compared to the randomized two-tone scenarios. In the 2450 MHz case, the MDP outperformed both the log-normal CDF and hazard function approaches in terms of collision and missed opportunities. The cumulative hazard function also shows improved performance compared to the randomized synthetic cases.
Since the hazard function is non-parametric the busy and idle time distributions may be more accurately estimated if they deviate from a log-normal distribution. In three of four cases, the MDP occupies more bandwidth (corresponding to less missed opportunities) than the stochastic approach while the stochastic models typically favor less bandwidth to avoid collisions. In the stochastic model, the interference and bandwidth trade-off can be adjusted with respect to the threshold $\theta$. A higher threshold utilizes less bandwidth (more missed opportunities) with less collisions while a low threshold favors wide bandwidths and collides more often. Similarly, the MDP reward structure can be modified to adjust the SINR and bandwidth trade-off. In some applications, it may become more desirable to heavily favor avoiding mutual interference at the cost of coarser range resolution. Additionally, the MDP reward structure could be modified to weigh additional parameters, such as waveform adaptation.

Figure 6.4: SLA collision and missed opportunity rates for the real RFI scenarios.
6.3 Notched FM Noise Waveforms

In prior sections, SPA, SLA, and SRA present alternatives to the spectrum perception component of the PAC. Here, we propose notching as an alternative to avoidance as the waveform action component of the PAC. An avoid action requires the radar to occupy the widest contiguous available bandwidth. This results in significant reduction of the radar’s effective bandwidth for operation. In the case of an RF emitter in the middle of a radar’s operating band, avoid leaves nearly half of the bandwidth unused. Waveforms with efficient notch generation are necessary to provide an alternative to avoidance and achieve SRN, SPN, and sense-learn-notch (SLN) CR strategies. The rest of this chapter introduces FM noise waveforms and an algorithm for efficient notch generation using the SDR.

Noise waveforms for radar are characterized by signals with a random amplitude and/or random phase according to some distribution. Several notable applications have utilized noise waveforms improve various radar characteristics. These waveforms allow provide a low probability of intercept aided by low probability of detection and performance in low-power scenarios [71,72]. Some have demonstrated improved range resolution in synthetic aperture radar imagery [73] and the ability
to suppress range sidelobes [74]. Both of these factors greatly improve the ability to resolve targets. In particular, FM noise waveforms are well aided toward high power radar applications due their ability to maintain constant amplitude. Similarly, FM noise signals have shown notable applications for efficient spectral shaping [72] and more specifically notching [48]. In the interest of efficiently generating notched waveforms, this section focuses on the pseudo-random optimized (PRO) FM technique. PRO-FM was originally introduced for continuous wave radar applications [72] but has since been applied to notched coexistence with pulsed radar [48]. This section discusses a real-time SDRadar implementation of notched PRO-FM waveforms to achieve SRN (from [67]). Then, this SRN implementation is extended to SPN and analyzed.

Achieving sufficient notch depth on COTS SDR hardware posed issues for real-time implementation. The USRP-2944R uses a low fidelity digital-to-analog converter (DAC) with a modest sampling rate. Before transmitting, the DAC reconstructs an analog signal through a zero-order hold model. This model involves the DAC holding an input sample constant for $T_s$ seconds during reconstruction. These synthesis imperfections reduce potential notch depth while transmitting. Work in [75] presents a zero-order reconstruction of waveforms (ZOROW) algorithm to preserve the notch quality. The real-time SRN implementation presented in [67] pairs PRO-FM with ZOROW to transmit waveforms with precise notches using the SDRadar.

PRO-FM uses the alternating projections method to generate an FM noise waveform adhering to constant amplitude and spectral shaping requirements [67]. Let $\bar{s}_0$ be a time-domain vector containing a digital representation of the waveform at the $0^{th}$ iteration of alternate projections. PRO-FM initializes $\bar{s}_0$ as a complex signal with constant amplitude and uniformly distributed random phase. Alternating projections is performed for $k \in \{1, \ldots, K\}$ iterations between the time and frequency domain of the signal

$$\bar{r}_{k+1} = \mathcal{F}^{-1}\{\bar{g} \odot \exp (j \angle \mathcal{F}\{\bar{s}_k\})\},$$

$$\bar{s}_{k+1} = \bar{u} \odot \exp (j \angle \bar{r}_k).$$

$\bar{r}_{k+1}$ is the frequency domain projection of $\bar{s}_k$ which imposes a spectral shape
defined by \( \tilde{g} \). The \( \odot \) symbol denotes the Hadamard product and the \( \angle(\cdot) \) operator extracts phase from a complex value. \( \mathcal{F} \) and \( \mathcal{F}^{-1} \) describe the Fourier transform and inverse Fourier transform respectively. In Equation (6.5), the time domain projection step imposes a discrete rectangular window \( \tilde{u} \) defined by the desired pulse width of the waveform. The spectral shape \( \tilde{g} \) is a discretization of \( |G(f)| \) which, in this case, is defined by \( |G(f)|^2 \) being a Gaussian shape over continuous frequency \( f \). The choice of a Gaussian for \( |G(f)|^2 \) allows for the radar waveform autocorrelation to be approximately Gaussian during processing. To place spectral notches in the waveform, a cognitive perception method informs PRO-FM of desired notch locations via \( \Omega \).

\[
|G(f)| = \begin{cases} 
  h_L(f), & f \in \Omega_L \\
  0, & f \in \Omega \\
  h_U(f), & f \in \Omega_H.
\end{cases}
\]  

(6.6)

The notching algorithm places lower \( h_L(f) \) and upper \( h_U(f) \) notch tapers at the edges of each notch location given by \( \Omega_L < \Omega < \Omega_U \).

After \( K \) iterations of notched PRO-FM, the resultant waveform can be described as \( \tilde{s}_K = \exp(j\phi) \) where \( \phi = [\phi_1, \ldots, \phi_N]^T \). Here, \( \phi \) is a vector of phases for an \( N \) sample notched PRO-FM waveform. After obtaining the digital representation of the notched PRO-FM waveform, ZOROW is applied to maintain efficient notch depth by accounting for the SDR DAC process. The DAC’s zero-order hold model holds each obtained sample of \( \tilde{s}_K \) constant for \( T_s \) seconds before passing the signal through a reconstruction filter [67]. The reconstructed analog signal can be represented in the frequency domain as

\[
S(f_m, \phi) = \frac{\sin(\pi f_m T_s)}{\pi f_m} \sum_{n=1}^{N} \exp \left( -j \left( 2\pi f_m(n - 1/2)T_s + \phi_n \right) \right).
\]  

(6.7)

\( f_m \) describes the discrete frequency index of this signal representation. After obtaining the zero-order hold reconstruction model from Equation (6.7), ZOROW optimizes the phase vector of \( \tilde{s}_K \) to minimize the following cost function

\[
\tilde{\phi} = \arg \min_{\phi} \sum_{f_m \in \Omega} \left| S(f_m, \phi) \right|^2.
\]  

(6.8)
The cost function’s summation is only performed at frequencies with notches given by \( f_m \in \Omega \). After computing \( \phi \), the ZOROW-aided notched waveform given by \( \tilde{s} = \exp(j\phi) \) is ready for radar transmission. The SDRadar implementation of notching (from [67]) estimates \( \phi \) via adaptive gradient descent with momentum and a backtracking technique to compute step-size.

### 6.4 Analysis of Sense-Predict-Notch

In consideration of the SPN implementation, the stochastic prediction model presented in Chapter 4 informs the notching algorithm of future RFI locations. These RFI locations determine the frequencies to generate notches (shown in Figure 6.6). SPN presents a simple modification of the SRN implementation on the SDRadar from [67]. The SDRadar architecture presented in Section 4.4 describes waveform generation being performed on the SDR’s FPGA. Maintaining this architecture, host-based spectral prediction is combined with FPGA-based notching to form a SPN CR strategy. The FPGA implementation was simplified to the application of fast Fourier Transforms (FFTs), inverse FFTs, multiplies, and additions in a burst streaming format. To meet the minimum timing constraints, 2 PRO-FM iterations and 6 ZOROW iterations were deemed sufficient to impose a desired spectral shape with a \( \sim25 \, \text{dB} \) notch depth relative to peak power after

![Figure 6.6: Example spectrum of a sense-react-and-notch (SRN) radar waveform generated with notches to coincide with sensed RFI.](image)
accounting for worst-case spectral variation. The SDR with this added waveform diversity supports pulse repetition frequencies up to 2.2 kHz, a minimum adaptation interval of 942 µs, and may incorporate multiple spectral notches per waveform.

Here, SPN is analyzed on a synthetic RFI set based on the experiments in Chapter 5 (and work in [76]). The SD Radar maintains the same 100-MHz operating bandwidth for spectrum sharing. The host performs energy detection and predictive processing by partitioning the spectrum into \( M = 20 \) separate 5 MHz wide frequency sub-bands. After using the stochastic model to predict spectral notch locations, the respective parameters are sent to the FPGA to perform waveform optimization and generation. This FPGA implementation of notched waveform generation minimizes the computational latency of a complex process. Given a single spectrum sensing FFT duration of \( T_0 = 40.96 \) µs, the time to predict ahead \( t_0 \) is discretized by \( N_0 \) timesteps where \( t_0 = N_0 T_0 = 491.5 \) µs. Due to waveform adaptation delay, the system operates with a minimum \( N_0 = 12 \) timesteps and a radar PRI determined by \( (N_0 - 1)T_0 = 450.6 \) µs. Consequently, the radar predicts and adapts the transmit waveform every PRI.

In these experiments, the VST generates RFI in real-time and feeds into a combiner with the SD Radar's transmitter (as per Chapter 5). The cognitive radar estimates the stochastic model for 410 ms of RF data and performs radar operation in the subsequent 410 ms. Here, only the parameteric log-normal model is considered (Section 4.2.1). During stochastic model estimation, this SPN implementation performs threshold optimization (described in Chapter 4) with \( \alpha = 0.95 \) to favor collision avoidance. The system is evaluated using three RFI scenarios: 1) swept tone, 2) random single tone, and 3) random two-tone (as shown in Chapter 5). For swept tone signals, the VST continuously sweeps a sinusoid over all 20 channels with a fixed dwell time. This case demonstrates the radar's ability to predict and coexist with deterministic RFI.

The random single tone case involves a single sinusoid at -17.5 MHz baseband with a fixed on-time and randomly varying off-times, or idle intervals. Baseband refers to the waveform frequencies after the SDR performs digital down-conversion with some arbitrary carrier frequency. The idle intervals are randomly generated according to a selected mean \( \mu_I \) and standard deviation \( \sigma_I \). This scenario evaluates performance at increasing levels of variation characterized by the idle coefficient of variation (CoV) \( c_v = \sigma_I / \mu_I \). For each fixed on-time \( \mu_B \), the mean idle interval
is set to be equal as $\mu_I = \mu_B$. Finally, for the two-tone case, the VST transmits two random switching sinusoids at -27.5 MHz and 22.5 MHz baseband. Similar to the single tone case, the on-time $\mu_B$ is fixed, with the randomly generated idle intervals defined by $\mu_I = \mu_B$ and some selected CoV $c_v = \sigma_I/\mu_I$. Each tone is given independent $\mu_I$ and $\sigma_I$ parameters with several tested combinations.

We consider performance metrics of collision rate, missed opportunity rate, and bandwidth improvement. A collision involves a radar waveform colliding or interfering with coexisting emitters while a missed opportunity refers to an unused open frequency channel during radar operation [15]. For this system, collisions are equivalent to a predicted missed detection or type II error and missed opportunities refer to predicted type I errors that result in falsely placed notches. The collision and missed opportunity rates validate the performance of prediction for spectrum processing. Bandwidth improvement specifies the additional bandwidth gained by transmitting a notched waveform as opposed to notchless LFM chirp-based avoidance that selects the widest available contiguous bandwidth.

For the swept tone RFI pattern, we evaluate dwell times of 819.2 $\mu$s, 2.05 ms, and 4.1 ms. The dwell time refers to the time spent by the RFI in each 5

**Figure 6.7:** Spectrogram of cognitive radar operation using a SPN strategy on swept tone interference (819.2 $\mu$s dwell time).
Table 6.2: Swept tone bandwidth improvement over notchless avoidance.

<table>
<thead>
<tr>
<th>Dwell Time</th>
<th>819.2 µs</th>
<th>2.05 ms</th>
<th>4.1 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bandwidth Improvement</td>
<td>20.3 MHz</td>
<td>19.2 MHz</td>
<td>17.2 MHz</td>
</tr>
</tbody>
</table>

MHz sub-band. Figure 6.7 demonstrates the system accurately avoiding a swept tone pattern. During RFI channel hops, the prediction approach may widen the notch to cover adjacent channels and minimize collisions. This redundant notching results in a higher missed opportunity rate compared to collisions shown in Figure 6.8. The missed opportunity rate increases for longer dwell times since the system gradually widens notches in anticipation of transitions. Table 6.2 shows a significant bandwidth improvement as a result of notching compared to traditional avoidance.

The random single tone RFI scenario evaluates performance with respect to the CoV $c_v$ of the idle time interval (summarized in Figure 6.9). These tests evaluated 3 sets with on-times $\mu_B$ of 819.2 µs, 2.05 ms, and 4.1 ms for a tone at -17.5 MHz. For each $\mu_B$, the idle $c_v$ value ranges from 0 to 1. Maintaining a fixed on-time ($\sigma_B=0$), emulates a communication system with a fixed size data burst and random

![Swept Tone Performance](image)

**Figure 6.8:** Collision and missed opportunity rates for a swept tone RFI pattern with varying dwell times.
Figure 6.9: Performance for the random single tone RFI in terms of (a) collision rate, (b) missed opportunity rate, (c) bandwidth gained, and (d) number of waveform adaptations between pulses.

time between requests for data transmission. Similar to results in [46], the missed opportunity rate increases with idle interval variation (Figure 6.9b). The use of prediction threshold optimization results in the number of observed waveform adaptations decreasing to almost 0 (Figure 6.9d) with a high \( c_v \). The threshold optimization results in a reduced adaptation rate to preserve performance in highly variable scenarios. As a result, threshold selection causes the collisions (Figure 6.9a) to decrease as variability increases. When no waveform adaptation occurs,
Table 6.3: Two-tone case definition and bandwidth improvement over notchless avoidance.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>-27.5 MHz CoV $c_v$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.225</td>
<td>0.45</td>
</tr>
<tr>
<td>22.5 MHz CoV $c_v$</td>
<td>0</td>
<td>0.32</td>
<td>0.45</td>
<td>0.45</td>
<td>0.225</td>
</tr>
<tr>
<td>Bandwidth Improved (MHz)</td>
<td>22.8</td>
<td>23.2</td>
<td>23.4</td>
<td>23.8</td>
<td>24.0</td>
</tr>
</tbody>
</table>

the system places a constant notch at the RFI location. Before the system stops adapting the transmit waveform, the collision rate shows a slight increase with $c_v$ (Figure 6.9a). Figure 6.9c shows a significant bandwidth improvement over the notchless avoidance implementation.

Finally, the two-tone RFI scenarios evaluate performance for two sinusoids with 5 different combinations of idle time statistics (Table 6.3). Similar to the single tone tests, the on-time is deterministic and equal to the mean idle interval such

Figure 6.10: Spectrogram of cognitive radar operation using a SPAN strategy on two-tone RFI.
that $\mu_B = \mu_I$ and $\sigma_B = 0$. Figure 6.10 shows an example of this switching two-tone RFI scenario. Similar to the single tone results, performance degrades for cases with higher $c_v$ values (Figure 6.11). Per Table 6.3, Case 1 is deterministic in that $c_v = 0$ for both tones. This case shows significantly lower missed opportunity rates than the other random cases where $c_v > 0$. Case 4 has the highest error rates where $c_v$ is largest for the shorter duration signal ($\mu_B = 2.05$ ms). Variability for shorter duration RFI has a larger impact on cognitive radar performance than slower changing RFI. The collision rate shows slight degradation for cases with a larger $c_v$. Notching demonstrates consistent bandwidth improvement for the two-tone scenarios (Table 6.3).

Compared to avoidance, results across all scenarios show that notching consistently occupies more bandwidth. This allows for improved range resolution during radar spectrum sharing. Despite additional processing latency for notching compared to LFM chirp avoidance, the system maintains prediction performance for different levels of time variation. Prediction error rates for predictive notching are comparable to the notchless avoidance results despite operating with a longer PRI and double the adaptation latency. This SPN framework allows for more efficient bandwidth utilization and mitigation of errors caused by RF state transitions.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{random_twotone_performance.png}
\caption{Random Two-tone Performance}
\end{figure}

\textbf{Figure 6.11:} Collision and missed opportunity rates for two-tone RFI with varying statistics in each case. Case statistics are shown in table 6.3.
Chapter 7

Metacognitive Radar

In prior chapters, this dissertation discusses a variety of CR strategies implemented on the SDRadar system for spectrum sharing. Particular emphasis was placed on the stochastic model-based SPA strategy. This spectral prediction model was evaluated against a comprehensive set of RFI, merged with notching to achieve SPN, and compared against SLA. The analysis of SPA and SPN with respect to variation in the EME provides the intuition that reactive CR strategies such as SRA and SRN may perform comparatively better with increasing variation. Furthermore, this comparison of CR strategies suggest that these cognitive PAC variants are well suited to particular RFI environments. These realizations led to an effort to develop a best-of-breed approach for cognitive spectrum sharing radar. The best-of-breed approach requires intelligently adapting the PAC and CR strategy based on observations and feedback from the EME. Here, we extend the state-of-art for cognitive radar by proposing a radar metacognition architecture for optimizing a spectrum sharing PAC with respect to the EME. Collaborative efforts between the U.S. Army Research Laboratory and several universities have resulted in the development of SPA, SRA, SLA, SRN, and SPN implementations on SDRadar platforms as separate systems. This chapter pairs the proposed metacognitive radar architecture with a merged implementation of several CR strategies on the SDRadar platform for real-time spectrum sharing.
7.1 Introduction to Metacognition

In the context of RF spectrum sharing, Fuster [19] characterizes cognition as a combination of the PAC, memory, attention, and intelligence. As described in Chapter 2, the PAC is the main controller of cognitive processing which regulates memory and attention via intelligence to actualize autonomous behavior in some agent. The agent under consideration here is a spectrum sharing cognitive radar. The PAC drives the processing for the aforementioned CR strategies implemented in real-time on the SDRadar. Previous sections discuss the notion that various CR strategies or PAC variants are better suited for particular EME conditions and scenarios. In dynamic rapidly changing EMEs, using a single CR strategy may present challenges and suboptimal performance. For spectrum sharing, the proliferation of wireless technologies such as 5G will further increase spectral congestion and volatility of EMEs. This emphasizes the need for a solution to optimize the cognitive PAC with respect to changing EMEs. In the interest of further optimizing a radar’s PAC, we extend cognitive RF to the bio-inspired concept of metacognition. This concept is used to describe a system which regulates its PAC by self-evaluating performance with respect to the observed environment. For cognitive radar, metacognition is applied to a mixed strategy of PAC variants to address time-varying scenarios where a single variant may not perform optimally.

Metacognition can broadly be described as learning about learning. It is a widely studied topic in the field of human cognition, which is used as a measure to self-assess problem solving capability, critical thinking, and to evaluate the overall efficacy of active learning [77]. This topic evaluates an agent’s accuracy in quantifying one’s own confidence in whether an action is successful or unsuccessful. Ridley [78, 79] defines metacognition as "taking conscious control of learning, planning and selecting strategies, monitoring the progress of learning, correcting errors, analyzing the effectiveness of learning strategies, and changing learning behaviors and strategies when necessary." For example, metacognition indicates learning potential for patients with neurological impairments by measuring their ability to self-evaluate [80]. This concept has been widely applied to the field of education to study the ability of undergraduate students to monitor metacognitive knowledge and self-regulate learning to optimize academic performance [79, 81]. Metacognitive capability has been evaluated in monkeys and other animal studies to perform perceptual and
memory tasks [79, 82]. Work in [83] describes a signal detection theoretic metric for metacognition to assess the quality self-evaluated confidence from performing cognitive tasks.

In this vein, metacognition in computation has become another wide area of research. The ability to assess the quality of one’s own perceptions can be exploited to adapt perception strategies and improve learning over time. Work in [84] proposes an intelligent dialogue system for students engaged in educational classes. This dialogue system acts as a mentor which autonomously detects weak points in a student’s learning and generates language-based suggestions to assist the student in self-regulating their learning. More canonical examples in computation include the algorithm selection problem and the self-tuning of parameters [85]. In the algorithm selection problem, a computer program selects the “best” algorithm (out of a set of algorithms) for a given task. The selection process considers characteristics of the given problem in addition to knowledge of each algorithm’s performance. During this process, the program performs self-evaluation to develop an understanding of each algorithm’s performance in the given task. This selection process has also been used for radar, where filter and CFAR algorithm pairs are selected for particular environmental terrain conditions [86]. Similarly, [79] demonstrates a metacognition loop to select strategies for air traffic control. This application monitors the environment for anomalies and adapts the strategy to minimize air traffic violations.

For spectrum sharing, metacognition has been applied to cognitive radio to select or intelligently tune parameters of the appropriate congnitive engine (CE) to address dynamic scenarios where a single CE may not suffice [87, 88]. The key components of metacognition for cognitive radio include metacognitive knowledge to define the performance and learning capabilities of each CE for given scenarios, metacognitive monitoring to determine if/when the operating scenario changes, and metacognitive control process to select different CE’s depending on radio performance. Recently, metacognition for radar has resurfaced as a means to improve the situational awareness of an intelligent active sensor and optimize its PAC in real-time [60, 89, 90].
7.2 Metacognitive Radar Architecture

Shifting the focus to radar spectrum sharing, a conceptual metacognition model is described to regulate a cognitive radar’s PAC. In the context of spectrum sharing, the PAC is defined by iterative sensing of the EME, some perception of the sensed EME, and an action in the form of some selected waveform. This cycle repeats as the radar observes the EMEs response to the selected actions. Here, we have modeled the cognitive SDRadar’s perception-action cycle as a combination of some spectrum perception method and a waveform action method. With sensing via energy detection as the first step of the PAC, this model provides the characterization of each CR strategy such as sense-predict-avoid or SPA. Previous chapters have presented a variety of CR strategies with different combinations of spectrum perception and waveform action methods. This metacognition architecture merges the aforementioned CR strategies onto one SDRadar platform and self-regulates to select the optimal CR strategy. In the merged implementation, the PAC considers spectrum perception methods: react, predict, and learn with waveform action methods: avoid and notch. All combinations of perception and action types result

![Diagram of the metacognitive engine interacting with the SDRadar PAC. The metacognitive engine observes spectrum features and radar/spectrum sharing performance as feedback. Example shows the PAC set to SLA.]

Figure 7.1: Diagram of the metacognitive engine interacting with the SDRadar PAC. The metacognitive engine observes spectrum features and radar/spectrum sharing performance as feedback. Example shows the PAC set to SLA.
in six possible CR strategies which include SRA, SRN, SPA, SPN, SLA, and SLN (see Figure 7.1). Predict for spectrum perception and avoid for waveform action is described in detail in Chapter 4. React for perception is introduced in Chapter 2 and discussed again alongside learn and the notch action in Chapter 6. Metacognition adapts the radar PAC through environmental assessment and self-evaluation of cognitive performance. The rest of this section presents a high level model for metacognitive radar (MCR) before discussing specific aspects of the SDRadar implementation.

The MCR engine operates in a cyclical fashion with respect to the PAC. This engine receives feedback from the cognitive radar in the form of EME information and self-evaluated performance. The cyclical operation of the MCR engine acts like a meta-PAC on top of a standard cognitive PAC (Figure 7.2). The MCR engine first monitors and categorizes the environmental scenario to develop situational awareness that aides in the selection of advantageous CR techniques. For example, a spectrum-sharing scenario would require spectrum monitoring to develop knowledge of emitter types and radiation patterns to help select the best CR techniques for radar performance [60,88]. The MCR engine is formed using MCR knowledge,
MCR monitoring, and MCR control components. The MCR knowledge defines the learning rate and performance capabilities for each CR technique for scenarios of interest.

The MCR knowledge is an expansion of the meta-strategic knowledge (MSK) model used for human learning which defines task and strategic knowledge [88, 91]. Here, the MSK task and strategic knowledge encompass the CR strategies and procedures to determine when a particular strategy is advantageous. These procedures include the algorithm for selecting the CR strategy as well as the EME metrics and performance self-evaluation to inform the algorithm. Given this, MCR knowledge provides the MCR engine with information about the CR strategies and utility [87]. MCR monitoring categorizes EME scenarios to determine the viability of CR strategies. The communications community has employed case-based reasoners to select an appropriate CR strategy based on the state of the EME [87,88]. This dissertation considers a similar case-based reasoner approach to aid the metacognitive engine’s strategy selection process. MCR control regulates the learning process and defines the algorithm to select a CR strategy by adapting the PAC.

Here, we present a MCR control framework based on a non-stationary form of the multi-armed bandit (MAB) problem known as the restless bandit problem [92]. Together, MCR knowledge, monitoring, and control cyclically operate in conjunction to regulate the cognitive radar PAC (shown in Figure 7.3). Restless bandits present a simple yet widely effective model for making decisions under uncertainty in non-

Figure 7.3: Decomposition of MCR engine showing the knowledge, monitoring, and control components interacting with a cognitive radar PAC.
stationary environments [93]. Through repeated interaction with the environment, the radar learns the expected performance of each CR strategy directly from feedback information. Each strategy must be selected enough times such that a reliable estimate of its performance is learned. However, the MCR eventually wishes to select the optimal strategy as often as possible. Thus, the MCR control process uses a MAB model to balance the fundamental trade-off between exploration and exploitation. These bandit algorithms are described in more detail in Section 7.4.

Given the SDRadar platform, all the aforementioned CR strategies or PAC variants (from Figure 7.1) must be merged onto a single system along with the MCR engine. Figure 7.4 shows where perception processing is implemented on the host CPU. Similarly, the PAC regulation process of the MCR engine is also performed using the host CPU on a larger timescale. As described in Chapter 4, the cognitive radar is initialized with a passive spectrum sensing and learning period where no radar transmission occurs. During this period, the radar learns stochastic models for the predict method and a MDP model for the learn method. For metacognition, the radar extracts spectral features from the stochastic model to perform case-based reasoning and categorize the EME. This case-based spectrum evaluation is performed repeatedly for some operator defined spectrum evaluation interval (SEI). After the initial SEI, the radar begins to transmit and cognitively adapt the waveform every pulse. After completing a CPI, the radar performs Doppler processing, the MCR engine measures performance feedback, and selects a CR strategy according to the restless bandit model.

**Figure 7.4:** Hardware configuration shows the data flow for the metacognitive SDRadar system. The MCR engine and PAC regulation is performed on the host CPU alongside the RFI perception and waveform decision (not shown).
Figure 7.5: Example MCR engine timeline with periodic explore-first. SEIs shown in green where the first one is denoted as passive (no radar transmission). CPIs (blue) denote performance evaluation occurring. The bottom line shows the periodic explore-first alternating between exploration (pink) and exploitation (red).

Every CPI the MAB model performs PAC regulation and every SEI the case-based reasoner identifies viable CR strategies for consideration by the MAB model. An example of a metacognitive processing timeline is shown in Figure 7.5 where a periodic explore-first algorithm is used. This algorithm explores each CR strategy at the beginning of each SEI and exploits the best performing strategy for the rest of the SEI (described in more detail in Section 7.4).

7.3 Case-based Reasoner

In Chapters 5 and 6, the performance of the disparate CR strategies were evaluated against a variety of RFI scenarios and conditions. The analysis suggested that these different CR strategies are well suited to specific conditions. To optimize MCR performance, the case-based reasoner extracts features from the EME and categorizes the scenario to identify viable CR strategies. Feature extraction exploits information formed by the stochastic model-based prediction method. While the spectral categories are arbitrary, a few potential classifiers are described based on observed behavior from prior experiments and expert knowledge of the CR strategies.

In this work, the case-based reasoner is be modeled as a generalized classification function

$$\omega_c = \Omega(Y_{SS}).$$

Above, $Y_{SS}$ is a spectrogram in the form of a $P \times N$ matrix where the $N$-dimension
describes discrete frequency domain samples and \( P \) is the number of FFTs obtained by the SDRadar in a SEI. This classifier performs generalized processing on the obtained spectrogram to provide a spectral category \( \omega_c \). Let there be a set of \( C \) spectral categories where \( c \in \{1, 2, \ldots, C\} \). Each category \( \omega_c \) can represent some combination of CR strategies that deemed viable for the EME conditions or a normalized ranking of CR strategies. Once again, a CR strategy is characterized by a pair of spectrum perception and waveform action methods (e.g. SPA, SRN, etc.). While the classifier is arbitrary, we consider a few different variations in this dissertation. This section describes a more thorough classifier concept presented in [60]. The experiments in the upcoming Section 8.3 consider a classifier which only categorizes the waveform action while including all perception methods. Conversely, experiments in Section 8.2 consider a classifier which ranks spectrum perception methods only (react, predict, and learn) while ignoring waveform actions (notch and avoid).

In [60], we presented a tiered case-based reasoner based on expert knowledge for Equation (7.1) shown in Figure 7.6. The case filtering provided by the top two tiers of the classifier captures corner-cases and rules out poor CR strategy choice. The various information used for classification criteria is obtained from

**Figure 7.6:** Spectrum classification based on a tiered knowledge-based classifier to identify the categories \( \omega_1, \omega_2, \ldots, \omega_C \).
the SDRadar’s memory after performing spectrum sensing and stochastic model learning. The rest of this section borrows notation from Chapter 4 and assumes the spectrum sensing and stochastic model estimation processes remain the same. This case-based reasoner relies on extracting average congestion $C_G$ and average complexity $C_X$ metrics. The cognitive radar operates with $M$ frequency sub-bands and obtains a $P \times M$ matrix of RFI states $S$ at the end of each SEI. $S$ is obtained from $Y_{SS}$ according to Equation (4.2). Here, we define average congestion as follows

$$C_G = \begin{cases} \frac{\sum_{p=1}^{P} \sum_{m=1}^{M} S_{p,m}}{M(P-P_0)}, & P_0 \neq P \\ 0, & \text{else}. \end{cases}$$  \quad (7.2)$$

Above, the elements of $S$ are given by $p \in \{1, 2, \ldots P\}$ and $m \in \{1, 2, \ldots M\}$. Let $P_0$ be the number of rows where no RFI was detected. This allows $C_G$ to describe the average rate of busy sub-bands only when RF activity is present. If there is no RFI the congestion is zero and if $C_G = 1$ then the spectrum is always full (top tier of Figure 7.6). In the top tier, the SDRadar enters a special case with fixed solutions consisting of no radar transmission (all RFI) and constant full bandwidth transmission (no RFI). The second tier investigates special conditions to determine the suitability of notching and reaction [60]. The average complexity exploits the statistics from the learned stochastic model

$$C_X = \frac{1}{2} \sum_{m=1}^{M} \left( \frac{\sigma_{B_m}}{\mu_{B_m}} + \frac{\sigma_{I_m}}{\mu_{I_m}} \right) \bigg/ M.$$  \quad (7.3)$$

Complexity is defined by the average coefficient of variation of both busy and idle times for all sub-bands. The $m^{th}$ sub-band on-time mean $\mu_{B_m}$ and standard deviation $\sigma_{B_m}$ are given by Equation (4.4). Similarly, the off-time statistics $\mu_{I_m}$ and $\sigma_{I_m}$ are given by Equation (4.5). As described in Chapter 5, the coefficient of variation is defined by the ratio of a standard deviation to the respective mean.

The bottom level of Figure 7.6 described an "advanced" classifier which scores or categorizes CR strategies according to thresholds on $C_G$ and $C_X$. Based on results in earlier sections, predict performs better with lower variation which is characterized by a low complexity score $C_X$. With increasing variation or complexity $C_X$, react becomes a more suitable strategy. Since learn is an MDP-based prediction strategy, we conjecture that learn maintains performance with comparable degradation to
predict as $C_X$ increases. Given this, $C_X$ acts as a threshold parameter to categorize the spectrum perception methods (react, predict, and learn). The notched PRO-FM waveform experiences poor autocorrelation peak-to-sidelobe ratios when the total width of all notches expands beyond 40% of the available bandwidth [76]. To mitigate this negative impact on radar processing, a congestion threshold of $C_G = 0.4$ can be used to categorize waveform action. If $C_G < 0.4$ then a notch action could be included in the spectral category and excluded otherwise. Figure 7.7 shows an example of a realization for an advanced classifier from Figure 7.6. The thresholds shown for the example in Figure 7.7 are loosely based on heuristic results from [42, 46, 76] and a priori knowledge of the algorithms. The $C_X$ thresholds given for perception methods require further refinement despite providing an appropriate baseline.

![Figure 7.7: Example of a bottom-tier advanced spectral classifier for metacognition. Each cluster contains the viable CR strategies to be included in the restless bandit model for MCR control.](image)
7.4 The Restless Bandit Problem

Here, we present the restless bandit problem for CR strategy selection as the MCR control framework on the SDRadar. This model is an extension of the multi-armed bandit (MAB) problem. The MAB presents a simple but powerful framework for algorithms that make decisions under uncertainty over time [93]. The term MAB comes from the gambling scenario where someone seeks to maximize payout from several identical looking slot machines (known as one-armed bandits). These identical slot machines yield different average payouts. To maximize winnings, the gambler faces a classic trade-off between exploration and exploitation. They must sufficiently explore the slot machines to determine which pays the best but spending too many rounds exploring results in suboptimal winnings. The MAB problem presents a wide array of algorithms to optimize this trade-off while maximizing some defined reward.

While MABs apply to a broad variety of domains, one of the earliest documented applications pertained to the design of ethical medical trials [93]. In 1933 [94], Thompson investigated approaches to attain useful medical data while maximizing the health of the patients. This gave rise to the seminal Thompson Sampling algorithm (discussed in Section 7.4.3). Other applications include recommendation systems such as movie or music recommendations on a website. These systems seek to maximize the number of recommendations followed by a user. MABs have been used to model stock investment and product pricing to maximize profits and sales respectively [93].

A basic stochastic MAB model is characterized by $K$ possible actions with unknown random independent and identically distributed (IID) rewards. This stochastic MAB is played for $T$ rounds where an algorithm selects an action and observes some reward for each round. In the case of the MCR, the actions refer to $K$ possible CR strategies and the reward is the normalized performance self-assessment done by the MCR. This reward metric is characterized by spectrum sharing or radar performance self-assessment (described in Section 7.5). A round is characterized by the MCR CPI where the MCR’s total runtime is $T$. Typically, observed rewards (in this case performance metrics) are bounded by the interval $[0, 1]$. Most stochastic MAB solutions assume that the reward distributions from each action remain constant over time [95]. In the context of spectrum sharing,
a CR strategy’s average performance depends on the EME conditions. If the EME scenario remains constant, then the CR strategy will maintain the same average performance. However, some RF environments can be volatile and change rapidly over time. With today’s increasing spectral congestion, RF activity becomes increasingly complex and non-stationary.

The restless bandit problem extends the stochastic MAB by assuming the action’s reward distributions are non-stationary over time. Restless bandits were originally proposed in [96]. Optimizing action selection with this model results in an NP-hard problem [95]. The restless bandit model presents solutions well aided to the non-stationary problem of MCR spectrum sharing. In the rest of this section, we present a set of simple algorithms to address non-stationary MABs for MCR. After the case-based reasoner identifies $K$ available CR strategies, a restless bandit-based algorithm selects a CR strategy every CPI. At each CPI, the MCR observes a performance self-assessment and factors this information in optimizing action selection for the next CPI.

### 7.4.1 Periodic Explore-first

As a baseline, this section presents a variation on the explore-first algorithm [93] that we call periodic explore-first. Explore-first is characterized by an initial explore phase which selects each action an equal number of times. Then, the action with the highest average reward is exploited until round $T$. Periodic explore-first simply resets the algorithm periodically or after each passing of some fixed time interval.

To further parameterize the algorithm, let the set of all $K$ viable CR strategies be $A_{CR} = \{a_1, a_2, \ldots, a_K\}$. The current CPI number is $t \in \{1, 2, \ldots, T\}$. Let $z_k$ be the latest reward feedback or self-assessed performance for the $k$th CR strategy and the set of average performance for each strategy be $\bar{Z} = \{\bar{z}_1, \bar{z}_2, \ldots, \bar{z}_K\}$. $T_{SEI}$ describes the number of CPIs in an SEI. $T_{exp}$ describes the fixed exploration time in number of CPIs such that $T_{exp}$ modulo $K = 0$. We maintain a history of the actions selected in $H(t)$, $\forall t$ such that $a_{H(t)}$ defines the selected strategy at time $t$. Similarly, a history of obtained performance is defined by $G(t) = \{z_{H(1)}, z_{H(2)}, \ldots, z_{H(T)}\}$. During the explore phase, if $T_{exp}$ is a multiple of $K$, each strategy $a_k$ is selected
evenly according to

\[ k = (t \mod T_{\text{SEI}}) - \left\lfloor \frac{(t \mod T_{\text{SEI}}) - 1}{K} \right\rfloor K. \] (7.4)

For this algorithm, \( i \) is defined for the previous explore phase from current CPI \( t \) to estimate the average performance \( \bar{z}_k \) given by

\[ i \in \left\{ \left\lfloor \frac{t - 1}{T_{\text{SEI}}} \right\rfloor + 1, \left\lfloor \frac{t - 1}{T_{\text{SEI}}} \right\rfloor + 2, \ldots, \left\lfloor \frac{t - 1}{T_{\text{SEI}}} \right\rfloor + T_{\text{exp}} \right\}, \] (7.5)

\[ \bar{z}_k = \frac{\sum_i G(i) 1\{H(i)=k\}}{\sum_i 1\{H(i)=k\}}. \] (7.6)

Above, \( \bar{z}_k \) is the average performance from the explore phase only. Indicator function \( 1\{H(i)=k\} \) is equal to 1 when the argument in the subscript is true and 0 otherwise. This constrains the average estimate \( \bar{z}_k \) to contain only the \( k \)th strategy.

Algorithm 4 describes the process for periodic explore-first. This presents a straightforward modification of the baseline explore-first for stochastic MABs to offset non-stationarity.

**Algorithm 4: Periodic Explore-first**

**input:** \( t \) - current round or CPI number  
\( z_k \) - performance feedback from the previously selected \( k \)th strategy  
\( T_{\text{exp}}, T_{\text{SEI}} \) - exploration parameters  

**output:** \( a_k \) - CR strategy to use for the next CPI

if \((t \mod T_{\text{SEI}} < T_{\text{exp}}) \land (t \mod T_{\text{SEI}} > 0)\) then

i) \( G(t-1) = \bar{z}_k \)

ii) Select \( a_k \) and \( H(t) = k \) according to Eqn. (7.4)

else

iii) Get average reward \( \bar{z}_k \), \( \forall k \) from exploration [Eqn. (7.5) and (7.6)]

iv) Select \( a_k \) and \( H(t) = k \) such that \( k = \arg \max_j \bar{z}_j \)

end

In Algorithm 4, steps (ii) and the preceding condition ensure that each strategy is explored evenly for \( T_{\text{exp}} \) CPIs repeating every SEI. Step (iv) exploits the strategy with the best average performance during the explore phase (given by Equation (7.6)).

Typically MAB models characterize performance bounds for algorithms over a
long time period using cumulative regret. Cumulative regret measures difference between the total maximum attainable reward and the total reward from an algorithm’s selected sequence of actions. Let $G^*(t)$ represent the maximum attainable performance if the best CR strategy sequence $H^*(t)$ was selected for all $t$. After operating for $T$ CPIs, the cumulative regret is described by

$$R(T) = \sum_{t=1}^{T} G^*(t) - \sum_{t=1}^{T} G(t).$$  
(7.7)

Most MAB and restless bandit frameworks have formally derived expected regret bounds $E[R(T)]$ with respect to a large $T$ to characterize performance over time. Lower cumulative regret corresponds to better performance. These regret bounds describe rate of convergence as well as potential performance limits. While the regret bounds of periodic explore-first have not been formalized, for traditional explore-first the expected regret is $E[R(T)] \leq T^{2/3} \times \mathcal{O}(K \ln T)^{1/3}$. Here, the $\mathcal{O}(\cdot)$ operator refers to big O notation which evaluates the argument as $T \to \infty$. As opposed to Algorithm 4, the explore-first algorithms explores uniformly from $i \in \{1, 2, \ldots, T_{exp}\}$ and exploits the best action the rest of the time. During uniform exploration for explore-first, the regret bounds are poor but significantly improve during exploitation [93]. For the stationary stochastic MAB case, most other algorithms achieve lower regret bounds than explore-first. While formal regret bounds have not been derived for Algorithm 4, we conjecture that more advanced restless bandit algorithms have better performance bounds.

### 7.4.2 Sliding Window UCB1-tuned

Here, we present a restless bandit variant of the upper confidence bound 1 (UCB1) tuned algorithm or UCB1-tuned with a sliding observation window. This is a variant of the UCB1 algorithm which presents a solution for adaptive exploration based on the premise of optimism under uncertainty [93,97]. UCB1 presents a simple solution to the stochastic MAB by exploiting the action with the largest average reward plus an upper confidence bound interval. This upper confidence interval (derived from the Chernoff-Hoeffding bound) describes the one-sided confidence interval in which the expected reward resides with overwhelming probability [97]. This confidence interval is $C \sqrt{\ln \frac{T}{t_k}}$ where $t_k$ is the number of times the $k^{th}$ strategy has been selected.
and $C$ is a constant controlling the likelihood of exploration. In the classical UCB1 formulation, $C = \sqrt{2}$. UCB1-tuned is an empirically derived variant of UCB1 which multiplies the confidence interval with a term dependent on the reward variance [97]. In the stationary case, UCB1-tuned estimates the mean reward and confidence intervals over the entire history of observed rewards. [92] presents a non-stationary variant of the UCB1 algorithm for abruptly changing environments called sliding window UCB1. This variant simply constrains the mean reward and confidence interval estimates to a local observation window. The upper regret bound for sliding window UCB1 depends on the number of breakpoints or abrupt changes in the environment. In this dissertation, we combine UCB1-tuned [97] with the sliding observation window [92] to present the sliding window UCB1-tuned algorithm. For both UCB1-tuned and our proposed sliding window variant, there are no formally derived regret bounds.

For sliding window UCB1-tuned (SW UCB1-tuned), the average performance $\bar{z}_k$ for strategy $k$ is estimated according to Equation (7.6) with the sliding window indexing for $i$ defined as

$$i \in \{t - T_w + 1, t - T_w + 2, \ldots, t\}.$$  (7.8)

The UCB1-tuned decision metric or average performance plus confidence interval over sliding window indices $i$ is given by

$$u_k = \bar{z}_k + C \frac{\ln \left( \min(T_w, t) \right)}{\sum_i 1_{\{H(i)=k\}}} \min \left( \frac{1}{4}, \frac{\sigma_k^2}{\sum_i 1_{\{H(i)=k\}}} \right) + \sqrt{2 \ln \left( \min(T_w, t) \right)} \left( \frac{2 \ln \left( \min(T_w, t) \right)}{\sum_i 1_{\{H(i)=k\}}} \right).$$  (7.9)

Above, the $\bar{z}_k$ term describes the sliding window average given by Equation (7.6) from CPI $t - T_w + 1$ to $t$ and the second term is the confidence interval for UCB1-tuned. $C$ is a tunable constant describing the likelihood of exploration. $\sigma_k^2$ is the variance of the performance for the $k^{th}$ strategy over the sliding window

$$\sigma_k^2 = \frac{\sum_i (G(i) - \bar{z}_k)^2 1_{\{H(i)=k\}}}{\sum_i 1_{\{H(i)=k\}}}.$$  (7.10)

SW UCB1-tuned explores each strategy once before exploiting the strategy that maximizes Equation (7.9). Algorithm 5 describes the process for SW UCB1-tuned.
Algorithm 5: Sliding Window UCB1-tuned

**Input:**
- \( t \) - current round or CPI number
- \( z_k \) - performance feedback from the previously selected \( k^{th} \) strategy
- \( C, T_W \) - exploration and sliding window parameters

**Output:** \( a_k \) - CR strategy to use for the next CPI

if \( t \leq K \) then
  i) \( G(t - 1) = z_k \)
  ii) Select \( a_k \) and \( H(t) = k \) according to \( k = t \)
else
  iii) \( G(t - 1) = z_k \)
  iv) Select \( a_k \) and \( H(t) = k \) such that \( k = \arg \max_j u_j \) from Eqn. (7.9)
end

Above, step (ii) defines the initial uniform exploration period while step (iv) selects strategy \( a_k \) with the best average performance plus confidence interval according to Equation (7.9).

### 7.4.3 Discounted Thompson Sampling

Discounted Thompson Sampling (DTS) is a non-stationary variant of the Thompson Sampling (TS) algorithm. TS was one of the earliest documented MAB algorithms in 1933 [94]. Despite this, formally proven performance guarantees have only been recently shown [93]. TS employs Bayesian estimation to iteratively estimate the reward distributions for each action. Given some assumed prior distribution for each action, these distributions are randomly sampled to obtain pseudo-reward values for each action. The action with the highest sampled pseudo-reward is selected. Once the true reward for the selected action is observed, the prior distributions for each arm are updated via Bayesian inference. Then, the resultant posterior distributions are sampled to select the next action and the process repeats. This sequential Bayesian update process allows TS to accurately estimate action reward distributions in stationary MAB problems. While the classical TS algorithm assumes a Beta distributed prior, others have used Gaussian priors and priors based on application specific assumptions [93]. Here, we assume a Beta prior for simplicity and its position as a common model for uncertainty in an event’s probability of success.
In consideration of non-stationarity, DTS (presented in [98]) adds a discounting or forgetting factor $\gamma \in (0, 1]$ to the Bayesian update for placing less weight on updates from older rewards. Here, each strategy’s reward distribution is initialized with a Beta prior $\theta_k \sim Beta(\alpha_0, \beta_0), \forall k$. In this work, we initialize the distribution parameters to $\alpha_0 = \beta_0 = 1$ where $\theta_k$ is a random sampling of the $k^{th}$ strategy’s distribution. We define $s = \{s_1, s_2, \ldots, s_K\}$ and $f = \{f_1, f_2, \ldots, f_K\}$ as a count of successes and failures for each strategy respectively. These values are initialized to $s_k = f_k = 0, \forall k$. Based on these initializations, Algorithm 6 defines DTS.

**Algorithm 6: Discounted Thompson Sampling**

**input**: $t$ - current round or CPI number  
$z_k$ - performance feedback from the previously selected $k^{th}$ strategy  
$\alpha_0, \beta_0, \gamma$ - initial Beta prior and discounting parameters

**output**: $a_k$ - CR strategy to use for the next CPI

**if** $t = 1$ **then**

i) Sample rewards $\theta_j \sim Beta(\alpha_0, \beta_0), \forall j \in \{1, 2, \ldots, K\}$

ii) Select $a_k$ and $H(t) = k$ such that $k = \arg \max_j \theta_j$

**else**

iii) $G(t - 1) = z_k$

iv) Perform Bernoulli trial $\hat{z}_k \sim Ber(z_k)$ to obtain 'success'*

v) Bayesian updates $s_j \leftarrow \gamma s_j + \hat{z}_k 1\{j = k\}$ and $f_j \leftarrow \gamma f_j + (1 - \hat{z}_k) 1\{j = k\}$

vi) Sample rewards $\theta_j \sim Beta(s_j + \alpha_0, f_j + \beta_0), \forall j \in \{1, 2, \ldots, K\}$

vii) Select $a_k$ and $H(t) = k$ such that $k = \arg \max_j \theta_j$

**end**

In the first CPI, step (i) samples pseudo-rewards for each initialized prior and step (ii) selects the strategy $a_k$ with the largest pseudo-reward. For every subsequent CPI, step (iv) determines the 'success' of a strategy by playing a Bernoulli trial with success rate $z_k$. In step (v), the outcome of this trial $\hat{z}_k$ is used to update the set of successes $s$ and failures $f$ with the discounting multiplier applied to the current values. Here, the $\leftarrow$ symbol is an assignment operator for the variable update. This Bayesian update only adds the outcome $\hat{z}_k$ to the previous chosen strategy $a_k$.

Finally, pseudo-rewards are sampled from the new posterior distribution in step (vi) and the strategy $a_k$ with the largest pseudo-reward is selected in step (vii). If $\gamma = 1$ and $\alpha_0 = \beta_0 = 1$ is chosen, then Algorithm 6 becomes standard TS for stationary environments. Compared to TS, DTS does not have formally defined
regret bounds.

7.4.4 REXP4

Here, we describe the REXP4 algorithm (from [99]) based on a non-stationary variant of the EXP4 algorithm. Presented in [100], EXP4 stands for exponential-weight algorithm for exploration and exploitation using expert advice (given by four words starting with "exp"). EXP4 demonstrates a solution for the special case of the MAB problem known as adversarial bandits with experts. In adversarial MABs, some adversary or set of adversaries controls the reward (or in this case cost) of each action. A randomized oblivious adversary is a special case which behaves similar to the stochastic MAB. Here, the adversary defines the rewards over some fixed random distribution. Adaptive adversaries alter the environment that an agent operates in [93]. In the case of a randomized adaptive adversary, the reward distributions may change over time. Some adaptive adversaries may competitively alter the environment in response to the agent’s actions. The restless bandit problem can be modeled similarly to a randomized adaptive adversary that adapts reward distributions obliviously or randomly compared to the agent’s actions. Given the definition of adversarial MABs, EXP4 assumes the environment is modeled as adversarial MABs with experts. This model assumes the existence of experts which provide the agent with recommendations or predictions an action’s likelihood of success. In our metacognition application, we define the expert as the output of the generic case-based reasoner or classifier described in Section 7.3 (from Equation (7.1)). In this section, we will present an alternative definition for the classifier which ranks each CR strategy based on complexity $C_X$.

EXP4 assigns each strategy $a_k$ a distribution $p_k$ where all the distributions $p_1, p_2, \ldots, p_K$ are defined as a mixture of the uniform distribution on the interval $[0, 1]$. These distributions are iteratively reassigned each CPI $t$ according to a probability mass exponential based on the cumulative reward for the action [100]. This distribution assignment is based on weighted expert advice and rewards from each strategy. Here, $\xi_m \in [0, 1]^K$ is a $K$-dimensional expert vector for the $m$th expert with $m \in \{1, 2, \ldots, M\}$. Let $\xi_{m,k}$ denote the $m$th expert’s score for the $k$th strategy. Each expert is assigned a weight $w_m$ which is updated each CPI. Here, we define a special case of the classifier from Section 7.3 in Equation (7.1)
to generate a set of $M$ experts $\xi_m$ that rank the $K$ CR strategies each SEI. The classifier extracts complexity features from each of the individual sub-bands to produce expert recommendations. For simplicity, this expert only provides heuristic recommendations for the spectrum perception methods (react, predict, and learn) with the number of CR strategies $K = 3$. Using the notation in Section 7.3, the cognitive radar operates with $M$ sub-bands and learns stochastic model parameters each SEI. The complexity for the $m^{th}$ sub-band is given by

$$c_{X,m} = \frac{1}{2} \left( \frac{\sigma_{I,m}}{\mu_{I,m}} + \frac{\sigma_{B,m}}{\sigma_{B,m}} \right). \quad (7.11)$$

The $m^{th}$ on-time mean $\mu_{B_m}$ and standard deviation $\sigma_{B_m}$ are given by Equation (4.4). Similarly, the off-time statistics $\mu_{I,m}$ and $\sigma_{I,m}$ are given by Equation (4.5). Let CR strategies $a_1, a_2,$ and $a_3$ correspond to SRA, SPA, and SLA respectively. The expert recommendation is obtained from this complexity estimate each CPI as follows

$$\xi_m = [\xi_{m,1}, \xi_{m,2}, \xi_{m,3}]^T = \begin{cases} [1, 0.6, 0.8]^T, & c_{X,m} > 0.5 \\ [0.6, 1, 0.8]^T, & \text{else} \end{cases} \quad (7.12)$$

This expert vector acts as a realization for the classifier in Equation (7.1) in the special case where only SRA, SPA, and SLA are considered. These expert vectors were determined heuristically based on observations from empirical data (including the results in Chapter 6). We pose this classifier realization along with the one presented in Section 7.3 as examples while this EXP4 construction will be demonstrated for proof-of-concept in Chapter 8.

Modeling the experts based on each sub-band’s features directly considers the impact of RF activity on the potential CR strategies. At the start of EXP4, the weights are initialized as $w_m = 1$, $\forall m$. Sub-bands that provide less accurate recommendations with respect to performance are iteratively weighed less over $t$. These adaptive weights $w_m$ provide more precise consideration of each sub-band’s contribution to the performance feedback. This algorithm includes a regret bound related tunable constant $r \in (0, 1]$. The aforementioned probability distribution assignments for each $k^{th}$ strategy is estimated as follows

$$p_k = \frac{(1-r)}{\sum_{m=1}^{M} w_m} \sum_{m=1}^{M} w_m \xi_{m,k} + \frac{r}{K}. \quad (7.13)$$
The REXP4 algorithm (from [99]) is a simple variation of EXP4 where the weights are reinitialized to \( w_m = 1 \), \( \forall m \) every \( T_R \) CPIs. Based on this description, Algorithm 7 defines the process for REXP4.

Algorithm 7: REXP4

\[
\begin{aligned}
\text{input} & : t - \text{current round or CPI number} \\
& z_k - \text{performance feedback from the previously selected } k^{th} \text{ strategy} \\
& r, T_R - \text{EXP4 constant and reset interval parameters} \\
\text{output} & : a_k - \text{CR strategy to use for the next CPI} \\
\text{if } (t - 1) \mod T_R = 0 \text{ then} & \\
& \text{i) Reset REXP4 by reinitializing weights } w_m = 1 \text{, } \forall m \\
\text{end} \\
\text{if } t > 1 \text{ then} & \\
& \text{ii) } G(t - 1) = z_k \\
& \text{iii) Apply IPS to performance } \hat{z}_j = (z_k/p_k) 1_{\{j = k\}}, \forall j \in \{1, 2, \ldots, K\} \\
& \text{iv) Weigh expert with pseudo-reward } \hat{y}_m = \xi_m \cdot \hat{z}, \forall m \\
& \text{v) Update weights } w_m \leftarrow w_m \exp \left( r \hat{y}_m / K \right), \forall m \\
\text{end} \\
\text{if } (t - 1) \mod T_{SEI} = 0 \text{ then} & \\
& \text{vi) Obtain updated expert vectors } \xi_m, \forall m \\
\text{end} \\
& \text{vii) Reassign probability distributions } p_k, \forall k \text{ according to Eqn. (7.13)} \\
& \text{viii) Sample the uniform distribution } \theta \sim U(0, 1) \text{ and let } p_0 = 0 \\
& \text{ix) Select } a_k \text{ and } H(t) = k \text{ such that } \sum_{j=0}^{k-1} p_j \leq \theta \leq \sum_{j=0}^{k} p_j \\
\end{aligned}
\]

Above, the condition before step (i) resets the EXP4 algorithm by reinitializing the weights. After each reward is obtained, an inverse propensity score (IPS) weighing is applied in step (iii) to improve the estimate of the reward distribution. This gives a pseudo-reward vector \( \hat{z} \) which is weighed against each reward vector via a dot product in step (iv). Next, this expert weighed pseudo-reward is used to update the expert weights \( w_m \) in step (v). For this MCR, the expert vectors \( \xi_m \) are updated once every SEI or \( T_{SEI} \) CPIs. Then, the probability distributions are updated according to Equation (7.13) in step (vii) and a random strategy is selected according to them (steps (viii) and (ix)). Formal regret bounds for REXP4 have been derived in [99].
7.5 Metacognitive Performance Self-evaluation

Each CPI, this MCR performs a performance self-evaluation as feedback to the metacognitive engine. This process is described in Section 7.2 and depicted in Figure 7.5. Here, we present two possible metric types for different desired spectrum sharing radar operating modes. One feedback metric is based on spectral feedback (described in Section 7.5.1) which is designed to optimize spectrum sharing efficiency. This takes into account the SINR in the spectrum and the radar’s transmit bandwidth. Similarly, we propose a metric to optimize radar performance based on the Jensen-Shannon divergence (described in Section 7.5.2). This metric optimizes a radar’s target detection ROC at all costs. The Jensen-Shannon divergence captures ROC impacts due to intra-CPI target distortions observed in [53]. This presents a 'greedy' mode of operation which favors the radar’s own performance at the potential cost of harming other RF emitters in the EME. The rest of this dissertation only considers the log-normal parametric model for prediction (Section 4.2.1).

7.5.1 Spectrum Sharing Performance

For spectrum sharing performance feedback, we defined a normalized value for spectral SINR by identifying potential occurrences of collisions. This spectral SINR considers the RF power before any radar processing is applied. Based on the notation in Chapters 4 and 5, $A_{TX(i,m \ast t_{PRI})} \in \{0, 1\}$ describes the waveform action selected by the cognitive radar. This describes whether the transmit waveform is occupying the $i$th sub-band at the $(mt_{PRI})_{th}$ FFT acquisition. $t_{PRI}$ describes the PRI in number of FFT acquisitions while $m$ describes the pulse number. $S_{TX(i,m \ast t_{PRI} - 1)}$ describes the detected RFI state (Equation (4.2)) for each $i$th sub-band and the $(mt_{PRI} - 1)_{th}$ FFT acquisition. In the remaining experiments, we define $M = 20$ total sub-bands for spectrum sensing. For MCR spectral performance feedback, collisions are estimated as

\[
c_{est(i,m)} = A_{TX(i,m \ast t_{PRI})} \wedge S_{TX(i,m \ast t_{PRI} - 1)}. \tag{7.14}
\]

Based on notation in Chapter 5, the numbers of pulses in a CPI are defined as $m \in \{1, 2, \ldots, Q\}$. Above, $c_{est(i,m)}$ is defined by a binary logical operator resulting in a 1 for true and 0 for false. Collisions are estimated using the FFT acquisition.
before each radar pulse to ensure reliable spectrum sensing. The estimation holds
given that the RFI states remain constant over average timescales longer than an
FFT acquisition (40.96\(\mu\)s). This remains valid for the RFI scenarios presented
in this dissertation. After \(c_{est(i,m)}\) is estimated for each each sub-band in a CPI
containing \(Q\) pulses, the average spectral interference power due to collisions is
estimated
\[
I_{est} = \frac{\sum_{m=1}^{Q} \sum_{i=1}^{M} c_{est(i,m)} \sum_{n=N_{s_i}}^{N_{e_i}} |Y_{SS,(m*t_{PRI}-1)}[n]|^2}{\sum_{m=1}^{Q} \sum_{i=1}^{M} c_{est(i,m)}}.
\] (7.15)

Based on notation in Chapter 4, \(Y_{SS,(m*t_{PRI}-1)}[n]\) is the \(n^{th}\) sample of the FFT
acquisition at the \((m*t_{PRI} - 1)^{th}\) timestep. This corresponds with the states in
\(S_{TX,(i,m*t_{PRI}-1)}\) for the \(i^{th}\) sub-band. The start and end indices of each \(i^{th}\) sub-band
is given by \(N_{s_i}\) and \(N_{e_i}\) respectively. Given some operator set transmit power \(S_{est}\)
and a calibrated RF receiver noise power \(N_{est}\), the spectral SINR is estimated as
follows
\[
\eta_1 = \frac{S_{est}}{I_{est} + N_{est}}.
\] (7.16)

We denote \(\hat{\eta}_1\) where \(\eta_1\) is converted to a logarithmic scale and heuristically nor-
malized from -20 dB to 50 dB to bound the metric to \([0,1]\). Similarly, we define a
normalized bandwidth utilization metric for each CPI
\[
\hat{\eta}_2 = \frac{\sum_{m=1}^{Q} \sum_{i=1}^{M} A_{TX(i,m*t_{PRI})}}{MQ}.
\] (7.17)

The resulting spectrum sensing performance feedback is defined as
\[
z_k = \hat{\eta}_1 (1 - \hat{\alpha}) + \hat{\eta}_2 \hat{\alpha}.
\] (7.18)

Referencing the notation from Section 7.4, \(z_k\) refers to the performance feedback
for the \(k^{th}\) CR strategy. Equation (7.18) allows the operator to define the spectral
SINR and bandwidth utilization trade-off as a weighted sum of the two metrics
given \(\hat{\alpha} \in [0,1]\). In this dissertation, we use a performance weighting of \(\hat{\alpha} = 0.25\)
to place emphasis on minimizing collisions. A smaller \(\hat{\alpha}\) favors avoiding collisions
with coexisting RF emitters while a larger \(\hat{\alpha}\) favors improving the radar’s range
resolution.
7.5.2 Jensen-Shannon Divergence

For performance feedback that describes radar target detector quality, we present a novel metric influenced by detection theoretic sensitivity measures. Missed detections and false alarms may result from reduced SINR due to collisions with RFI. Another major contributor to false alarms occurs from intra-CPI adaptation during radar spectrum sharing [53]. This section poses a metric to capture the intra-CPI distortion effects as well as impacts due to RFI collisions. According to Chapter 3, the MCR employs CA-CFAR detector on the processed range-Doppler map. From Equation (3.11), this detector compares the estimated energy in each cell to a threshold $|d[t_r, m_r]|^2 \geq \Lambda$. Here, $d[t_r, m_r]$ describes the range-Doppler cell under test at range $t_r$ and Doppler $m_r$ while $\Lambda$ is the adaptive threshold calculated by the CA-CFAR criteria. In the field of psychology and neuroscience, the distance between noise and signal distributions $d'$ is a used as a heuristic measure of detection quality or cognitive sensitivity in performing a task [101]. An example of this heuristic metric is shown in Figure 7.8. In psychology, signal detection theory is sometimes applied to model the successful completion of cognitive tasks such

![Figure 7.8](image)

**Figure 7.8:** Example of probability distributions for a generic signal detection problem with the statistic under test denoted by $|D|^2$. The $d'$ metric is shown as the distance between the means of the distributions.
as answering a question correctly. For these applications, the \(d'\) metric measures the quality of a detector or sensitivity of performing a task with respect to some parameter. Others have developed variants of this \(d'\) metric for metacognition [83] to assess the sensitivity of a self-reported confidence value in psychology.

Here, we draw influence from this signal detection theoretic notion describing the distance \(d'\) between two distributions. The Jensen-Shannon (JS) divergence presents a method of measuring the distance or similarity between two probability distributions. This metric is bounded on the interval \([0,1]\) in addition to acting as a symmetrized and smoothed version of the Kullback-Leibler (KL) divergence [102]. JS divergence allows for a more clearly defined metric for metacognition compared to \(d'\) which is often estimated heuristically or iteratively. After each CPI, the MCR performs CA-CFAR to detect targets and estimates the noise and target probability distributions using \(\Lambda\). Let the noise distribution be \(P_0(d) = \Pr[d \in D : |d|^2 < \Lambda]\) and target distribution be \(P_1(d) = \Pr[d \in D : |d|^2 > \Lambda]\). The distributions \(P_0\) and \(P_1\) are generated using histogram based estimates on the MCR. The KL divergence between \(P_0\) and \(P_1\) is given by

\[
D_{\text{KL}}(P_0 || P_1) = \sum_{d \in D} P_0(d) \ln \left( \frac{P_0(d)}{P_1(d)} \right).
\]

From this, the JS divergence or performance feedback for the \(k\)th CR strategy is described

\[
z_k = D_{\text{JS}}(P_0 || P_1) = \frac{1}{2} D_{\text{KL}}(P_0 || M_P) + \frac{1}{2} D_{\text{KL}}(P_1 || M_P).
\]

Above, \(M_P = \frac{1}{2}(P_0 + P_1)\) describes a mixture of the two distributions averaged together. By construction, the JS divergence should capture any false alarm and missed detection errors that occur in \(P_0\) and \(P_1\).

The efficacy of this JS divergence metric was confirmed using a MATLAB-based radar simulation with parameters and a target scene identical to that from Section 3.3. To reiterate (description and notation from Section 3.3), the simulation models a pulsed radar system with a 100 MSamples/s sampling rate that generates range-Doppler maps with 400 pulses per CPI and a 9.8 kHz PRF. The radar transmits LFM chirp pulses with some start and end frequency. Each target has a different loss factor with the leftmost target having \(L = 1\), the target at range 325 and
Doppler 0.25 has a loss $L = 0.7$, range 225 and Doppler 0.25 has loss $L = 0.4$, and the rightmost target has $L = 0.2$. This radar processing is simulated in MATLAB with a -90 dB noise floor and the radar transmit power at 0 dB. CA-CFAR target detection is run on each target scene with a $P_{FA} = 10^{-7}$ and a guard cell radius of $g = 1$. Figure 3.5 in Section 3.3 shows an example of the target scene with no RFI and fixed full-bandwidth radar pulses during the CPI. Similarly, in that previous section four different RFI scenarios were described: 1) No RFI, 2) a constant -30 dB tone at -10 MHz baseband, 3) a -30 dB tone linearly sweeping from -10 MHz to 10 MHz baseband and 4) a -30 dB tone randomly hopping between -10 MHz and 10 MHz baseband. The simulation was ran against each of these RFI scenarios and measured the JS divergence according to Equation (7.20), the number of actual false alarms, and the peak-to-average SINR in the range-Doppler map. For each RFI scenario, the radar transmit power tested at $P_t = \{-50, -40, -30, -20, -10, 0\}$ where $P_t$ is measured in dB. Additionally, a set of 12 unique radar pulse adaptation patterns were tested for each scenario and transmit power pair. The start and end frequencies of an LFM chirp waveform are reset pulse-to-pulse according to each pulse pattern sequence. This set pulse pattern sequence repeats until the end of the CPI. For specific details on the selected pulse patterns, see the Appendix. When the transmit waveforms collide with the RFI, this may result in reduced SINR. The reduced SINR, in turn, reduces the probability of detection with respect to some CFAR [52]. Depending on the pulse pattern, intra-CPI adaptation results in varying levels of false targets (as seen in Section 3.3). The purpose of these pulse patterns serve to produce varying levels of target distortion due to intra-CPI adaptation.

This entire set of scenarios and parameters were evaluated and the respective JS divergences, number of false alarms, and peak-to-average SINRs were collected for each range-Doppler map. From this, the Pearson correlation coefficient was calculated to compare the JS divergence to number of false alarms as well as the JS divergence to the SINR. For a pair of random variables $X$ and $Y$ the correlation coefficient is defined as $\rho = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$. Here, $\text{cov}(X,Y)$ is the covariance of the two arguments. The correlation coefficient for JS divergence and the actual number of false alarms is $\rho_{FA} = -0.943$. Comparing JS divergence the peak-to-average SINR gives a correlation of $\rho_{SINR} = 0.671$. Both the strong negative correlation suggests that as the distance between the noise and target distributions grow, the number
of false alarms decreases. This reinforces the idea that as the distance between the distributions grow, the detector has greater sensitivity or detection quality. The moderate positive correlation suggests that as the distance between distributions grows, the SINR also shows some growth. This further supports the idea that JS divergence may provide an informative metric to characterize radar detection performance as MCR feedback. JS divergence is beneficial for radar metacognition since no a priori knowledge or assumptions about the ground truth targets are needed. Since intra-CPI adaptation has been shown to be a large source of false alarms [53], the JS divergence metric has potential to characterize the level of distortion caused. The strong negative correlation with false alarms paired with the presence of intra-CPI distortions in the simulation further supports this idea.
Experimental Analysis of Metacognition

This chapter analyzes a few variations of the metacognition architecture implemented on the SDRadar. This metacognitive SDRadar runs in real-time on a set of RFI scenarios with hardware emulated point targets in the scene. The next section describes the hardware configuration and RFI scenarios. Section 8.2 compares restless bandit models for selecting the spectrum perception method with avoidance. In this case, the MCR chooses between SRA, SPA, and SLA. Section 8.3 presents a case-based reasoner for selecting the waveform action (avoid or notch) while the restless bandit algorithm optimally decides the perception method (react, predict, or avoid).

8.1 RFI Test Scenarios

Here, the MCR’s real-time spectrum sharing performance is evaluated against a sequence of RFI scenarios that occur in succession. The SDRadar parameters remain the same as previous experimental sections. The receiver performs IQ sampling at 100 MSamples/s giving the radar a 100 MHz maximum bandwidth. The radar operates with a PRI of 410 µs and 100 pulses per CPI or 41 ms. Spectrum sensing and model estimation is performed with 20 spectral sub-bands at 5 MHz each while the MDP model runs with five 20 MHz sub-bands. An SEI contains 15 CPIs or 615 ms. Each SEI the MCR trains the stochastic model for predict, the MDP for learn, and classifies the spectrum if applicable. For these experiments, only the parameteric log-normal model is considered for predict and Section 8.2 only performs
spectrum classification according to the REXP4 algorithm for metacognition. The radar performs passive sensing with no transmission during the first SEI. Each CPI the radar conducts performance self-evaluation as input to the restless bandit model and then this model selects a CR strategy for the subsequent CPI in real-time. In Section 8.2, the MCR always uses avoid and uses the restless bandit model to select the perception strategies (react, predict, and learn). This makes the set of CR strategies SRA, SPA, SLA. Constraining the choice of waveform action allows for proof-of-concept and validation of the restless bandit model. MCR with notching is considered in later sections.

In these experiments, the hardware configuration is similar to that of Chapter 4 except an additional VST performs real-time target emulation. One VST generates RFI in real-time and feeds into a combiner along with the SDRadar transmitter. The SDRadar transmit signal passes through a second VST before reaching the combiner. This second VST applies a time delay and quadratically increasing phase shift to create test targets with a desired range and Doppler shift. This configuration is shown in Figure 8.1. In these experiments, the VST emulates two constant velocity point targets for obtaining radar detector performance (shown in Figure 8.2). Not visible in the figure, the metacognitive processing is performed on

**Figure 8.1:** Diagram of the hardware configuration for the metacognition experiments. This diagram shows an added target emulator compared to prior analyses.
Figure 8.2: Capture of range-Doppler map with two constant velocity point targets generated by the VST. This was captured on the LabVIEW front panel on the SDRadar host.

the host CPU in real-time. In the next section, the VST generates a sequence of four different RFI scenarios lasting for 4 SEIs or 2.46 s each. These four scenarios lasting for 2.46 s each result in a total test time of 9.84 s, all processed in real-time. The first RFI scenario is a swept tone interferer that steps at 5 MHz intervals with a dwell time of 1 ms (shown in Figure 8.3a). Next, the second RFI scenario is a random frequency hopping tone with a dwell time of 4.1 ms (shown in Figure 8.3b). The next two RFI scenarios are emulated LTE waveforms designed to randomly generate realistic 4G LTE spectra for testing spectrum sharing systems. Presented in [103], the emulator outputs LTE frames from a bank of 10 ms waveform snippets with random idle periods between each frame. The third RFI scenario consists of an emulated single carrier 20 MHz LTE uplink emitter (Figure 8.4a).
Figure 8.3: Spectrograms of the first two RFI scenarios for the metacognitive SDRadar tests in Section 8.2. (a) shows a case I) swept tone RFI and (b) shows case II) a random frequency hopper.

Figure 8.4: Spectrograms of the second two RFI scenarios for the metacognitive SDRadar tests in Section 8.2. (a) shows case III) a single 20 MHz LTE uplink carrier and (b) shows case IV) two 20 MHz LTE uplink carriers.
Figure 8.5: Timeline of RFI scenarios for these experiments. The scenarios are as follows: I) swept tone, II) random frequency hopper, III) single 20 MHz LTE uplink carrier, and IV) 40 MHz LTE (two 20 MHz uplink carriers).

The last RFI scenario consists of two evenly spaced 20 MHz LTE uplink carriers (Figure 8.3b). This last scenario results in 40% spectral congestion. The RFI sequence can be summarized as: I) Swept tone, II) Random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE (shown in Figure 8.5).

These initial experiments use restless bandits to select from SRA, SPA, and SLA. To accurately characterize the performance of these restless bandit models, the performance upper bound is empirically measured for each RFI scenario and CR strategy. Performance upper bounds require measuring the spectrum sharing performance metric from Equation (7.18) and the JS divergence from Equation (7.20) over multiple CPIs of each RFI scenario. The maximum measured performance for each metric defines the upper bound. For the experiments using spectrum sharing performance feedback, the weighing parameter is set to $\hat{\alpha} = 0.25$. This weighing favors avoiding collisions at the cost of bandwidth utilization. From the measured upper bound, we determine the best CR strategy in each scenario and the $G^*(t)$ parameter from Equation (7.7) which describes the maximum attainable performance from the best strategy. From this, the cumulative regret $R(T)$ over the experiment can be measured to characterize each restless bandit model’s
Figure 8.6: Upper bounds for spectrum sharing performance in each RFI scenario over time. Spectrum sharing performance weighing set to $\hat{\alpha} = 0.25$ to favor avoiding collisions. The scenarios are: I) swept tone, II) random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE.

For the spectrum sharing performance feedback, the upper bounds on performance are shown in Figure 8.6. For the first swept tone RFI case, SPA strongly outperforms the other CR strategies. Since this pattern is deterministic, the prediction model learns and reliably anticipates changes in the spectrum. SRA performs best in the next random frequency hopper RFI case. In random hopping EMEs, SPA has shown a tendency to reduce bandwidth to maintain collision avoidance. The random pattern is difficult to predict for SPA or SLA. Both SRA and SPA demonstrate similar performance for the 20 MHz LTE case. In the last 40 MHz LTE case, SPA performance degrades due to a likely increase in spectral variation.

The JS divergence upper bounds show that different strategies perform better when considering radar detection (Figure 8.7). For case 1 with a swept tone RFI, SPA still performs the best. SLA and SPA both perform better in the random hopping case as well as the 40 MHz LTE case. These CR strategies tend to reduce
the rate of intra-CPI adaptation in highly variable scenarios. This may be reducing the intra-CPI target distortion and, in turn, false alarm rate. The adaptation pattern presented by SRA in the third 20 MHz LTE case outperforms the other strategies. Since all of these CR strategies have demonstrated a fair ability to avoid collisions, it’s likely that the JS divergence metric describes the CR strategy with minimal intra-CPI distortions or false alarm rate.

8.2 Comparing Restless Bandits

Here, the restless bandit models presented in Section 7.4 are compared from real-time processing using the RFI and target scenarios described last section. The bandit models under consideration include Periodic Explore-first, Sliding Window UCB1-tuned, Discounted Thompson Sampling, and REXP4. In this case, REXP4 employs a special case of spectrum classification (Section 7.3) to generate "expert" recommendations for CR strategies based on the EME conditions. The other bandit
models do not utilize any case-based reasoning. These experiments only consider the avoid waveform action leaving the CR strategies to include SRA, SPA, and SLA. Periodic Explore-first is set to reexplore every SEI with $T_{SEI} = 15$ CPIs and an exploration period of $T_{exp} = 6$ CPIs. The sliding window size for SW UCB1-tuned was set to $T_W = 20$ CPIs and the exploration constant set as $C = 0.25$. In DTS, the discounting parameter was set to $\gamma = 0.95$. Lastly, REXP4 was resets every $T_R = 60$ CPIs with the adaptation constant set to $r = 0.05$. These parameters were manually tuned to heuristically optimize performance.

### 8.2.1 Spectrum Sharing Feedback

This realization of the metacognitive SDRadar uses spectrum sharing performance self-evaluations as feedback. The feedback is a linear combination of spectral SINR estimates and bandwidth utilization each CPI. The weighing parameter is set to $\hat{\alpha} = 0.25$ to favor improving SINR or avoid collisions. From the upper performance bounds, Figure 8.8 shows the cumulative regret of each bandit model over time.

![Cumulative Regret for Each Model](image)

**Figure 8.8:** Cumulative regret for each bandit model based on the spectrum sharing performance bounds. SW UCB1-tuned shows the lowest regret and, in turn, best performance. The scenarios are: I) swept tone, II) random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE.
Figure 8.9: Number of times the best CR strategy is selected for each RFI scenario. RFI scenario is defined on the x-axis. Best CR strategy is defined in order of RFI scenario: I) SPA, II) SRA, III) SRA or SPA, IV) SRA. where lower regret corresponds to better performance. Here, SW UCB1-tuned performs the best followed by Periodic Explore-first, REXP4, and DTS. This experiment’s RFI sequence presents a case of abruptly changing RF environments. The SW UCB1-tuned algorithm was originally modeled to account for abruptly changing environments in [92]. For the REXP4 algorithm, the first SEI after a RFI scenario change can contain outdated classifier information. The spectrum features result from average statistics over an entire SEI. Similarly, REXP4 may result in better performance with more carefully designed classifiers or experts.

Figure 8.9 shows the number of times the best CR strategy is selected in each RFI scenario. Aside from the 20 MHz RFI case, SW UCB1-tuned selects the best strategy more than the other bandit models. Aside from the 40 MHz LTE case, REXP4 demonstrates close performance to Periodic Explore-1st. Figure 8.10 shows the spectrum sharing metrics as individual components averaged over each RFI scenario. In all metrics (spectral SINR, collision rate, and bandwidth utilization), the individual performance corresponds with SW UCB1-tuned outperforming other models (except for the 20 MHz LTE case). In the case of bandwidth utilization (Figure 8.10c), SW UCB1-tuned significantly outperformed the other models in the
random hopper case. This is likely due to the tendency of SPA and SLA to reduce bandwidth in the case of high variation.

Figure 8.10: Average spectrum sharing metrics for each individual RFI scenario. (a) shows the spectral SINR in dB, (b) shows average collision rate, (c) shows the average bandwidth utilization. Each metric is presented as an average over each RFI scenario. The scenarios are: I) swept tone, II) random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE.
Figure 8.11: Cumulative regret for each bandit model based on the JS divergence performance bounds. SW UCB1-tuned shows the lowest regret and, in turn, best performance. The scenarios are: I) swept tone, II) random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE.

8.2.2 Jensen-Shannon Divergence Feedback

This section considers using the JS divergence metric as metacognitive performance feedback to describe radar detection quality. The JS divergence is applied according to Equation (7.20) each CPI to estimate the distance between the detected noise and target distributions. With the JS divergence upper bounds in consideration, Figure 8.11 shows the cumulative regret for each restless bandit model. The analysis in Section 8.1 suggests that the JS divergence metric should minimize intra-CPI distortions and, in turn, false alarm rate. Relative to the spectrum sharing performance feedback, each bandit model performed similarly. The SW UCB1-tuned algorithm outperformed the other models. This reinforces its design for tolerating abruptly changing environments. REXP4 showed degraded performance for the JS divergence metric. This may be a result of the expert recommendations still being based on the spectrum classifier. This spectral complexity statistic does not account for intra-CPI distortions or radar detection performance.

Figure 8.12 shows the number of times the best CR strategy is selected in each RFI scenario based on the JS divergence. Despite SW UCB1-tuned achieving
Figure 8.12: Number of times the best CR strategy is selected for each RFI scenario for the JS divergence metric. RFI scenario is defined on the x-axis. Best CR strategy is defined in order of RFI scenario: I) SPA, II) SLA or SPA, III) SRA, IV) SPA or SLA.

the lowest cumulative regret, it was not consistent in selecting the best strategy according to the upper bounds on performance. The empirically estimated upper bounds presented in Figure 8.7 were mostly close in value. Due to this, it may be possible that the JS divergence experiences large variation between CPIs. Explore-first (second best performing model) appears to select best strategy with greater frequency than SW UCB1-tuned.

Figure 8.13 describes the ground truth range-Doppler metrics and bandwidth utilization (Figure 8.13c) as an average over each RFI scenario. Aside from the swept tone RFI case, SW UCB1-tuned results in the lowest average false alarm rate (Figure 8.13a). Similarly, periodic explore-first achieved the second lowest false alarm rate. This reinforces the idea that JS divergence describes the false alarms that occur with respect to intra-CPI adaptation. In this vein, SW UCB1-tuned achieved the lowest bandwidth utilization followed by Periodic Explore-first. This lowered bandwidth utilization may be a result of the selected CR strategies avoiding occupying sub-bands with large variations. As a result, this may result in adaptation patterns which produce less intra-CPI distortion.
Figure 8.13: Average range-Doppler metrics for each individual RFI scenario, (a) shows the average false alarm rate, (b) shows average range-Doppler SINR. (c) shows the average bandwidth utilization. Each metric is presented as an average over each RFI scenario. False alarm rate is defined with respect to the total number of cells in the scene. Range-Doppler SINR is the peak target signal to average interference-plus-noise power. The scenarios are: I) swept tone, II) random hopper, III) 20 MHz LTE, and IV) 40 MHz LTE.

The performance bound results (Figure 8.7) show that the best strategies for these RFI scenarios are SPA or SLA in most cases. Both of these strategies tend to adapt more conservatively which reduces the number of waveform adaptations. For
range-Doppler SINR in Figure 8.13b, the bandit models showed small difference in performance (<2 dB). SW UCB1-tuned showed the overall poorest SINR while Periodic Explore-first showed mostly high SINR values. REXP4 achieved comparatively higher SINRs which may be due to the use of spectrum classification. Since all the CR strategies have been demonstrated to achieve fair RFI avoidance, this JS divergence metric may prioritize minimizing the intra-CPI distortions. These distortions have an overall larger impact on target detection.

8.3 Classifier-aided Periodic Explore-first

Here, another realization of the metacognitive SDRadar architecture is presented. This implementation includes spectrum classification to switch between an avoid or notch waveform action, while the restless bandit optimally selects react, predict, or avoid. Compared to the previous section, this expands the action space to include SRA, SPA, SLA, SRN, SPN, and SLN. Here, this section shows proof-of-concept of metacognition with notching as well as examines the performance of individual CR strategies. The MCR implementation utilizes spectrum sharing performance metrics (Equation (7.18)) with a weighing parameter of $\hat{\alpha} = 0.25$ to favor avoiding

Figure 8.14: Spectrograms of the new RFI scenarios for the metacognitive SDRadar tests in Section 8.3. (a) shows case I) two intermittently switching tones and (b) shows IV) the 60 MHz case with three 20 MHz LTE uplink carriers.
collisions. This implementation is evaluated against a modified sequence of RFI scenarios. The RFI scenarios under consideration include: I) two periodically switching tones spaced 40 MHz apart (switches on-to-off/off-to-on 2.05 ms/4.1 ms respectively) in Figure 8.14a, II) one 20 MHz LTE uplink carrier (Figure 8.4a), III) 40 MHz LTE from two 20 MHz uplink carriers (Figure 8.4b), and IV) 60 MHz LTE from three 20 MHz uplink carriers (Figure 8.14b). The 60 MHz LTE case uses the same LTE emulation process discussed in the previous section except with three LTE carriers. For these tests, the MCR uses similar settings as Section 8.1 with 100 MSamples/s sampling rate, 20 spectral sub-bands, 410 µs PRI, and a 41 ms CPI. As opposed to previously, the MCR uses a SEI of 20 CPIs or 820 ms. Each RFI scenario lasts for 4 SEIs or 3.28 s. With 4 RFI scenarios, the total test time is 13.1 s. Here, Periodic Explore-first (Algorithm 4) is used for metacognition. The algorithm is set to reexplore every $T_{SEI} = 20$ CPIs and uniformly explore for $T_{exp} = 6$ CPIs.

During the first SEI, the MCR performs passive spectrum sensing with no radar transmission to train the stochastic model and MDP as well as extract spectral features. After this, the radar begins radar processing and reextracts spectral features each SEI. In this realization of metacognition, the average spectral congestion is used to select an avoid or notch waveform action. This presents a specific case of the classifier from Section 7.3 that utilizes the bottom tier classifier (Figure 7.6) to threshold the average spectral congestion $C_G$. In this implementation, we only consider the $C_G$ of the classifier presented in Figure 7.7. From Equation (7.2) to obtain $C_G$, this case-based reasoner is defined as

$$A_{CR} = \{a_1, a_2, a_3\} = \begin{cases} \{\text{SRN}, \text{SPN}, \text{SLN}\}, & C_G < 0.4 \\ \{\text{SRA}, \text{SPA}, \text{SLA}\}, & \text{else.} \end{cases}$$

Based on the reasoner’s selection of avoid or notch each SEI, the Periodic Explore-first algorithm chooses between the spectrum perception methods (react, predict, and learn) each CPI.

Figure 8.15 shows the CR strategies selected by the restless bandit model over time while Figures 8.16a through 8.16c show individual performance metrics for each spectrum perception method (React, Predict, or Learn) and all four RFI scenarios. Each RFI scenario lasts for 80 CPIs. RFI scenario I (switching two-tone)
Figure 8.15: Selected CR strategy and MAB operating mode with respect to time. RFI scenarios labeled as I) two-tone, II) 20 MHz LTE, III) 40 MHz LTE, and IV) 60 MHz LTE.

is deterministic over-time and well suited for Predict. Since the two-tones are narrowband, the MCR consistently converges to SPN and is shown to have the best SINR and collision rate compared to other strategies. In scenario II (20 MHz LTE), the bandit model evenly switches between SPN and SRN. While SRN has slightly better average SINR, the MAB likely selects both due to their relatively small difference in spectral SINR. In scenario III (two 20 MHz LTE UL carriers), all CR strategies are selected with SRN being chosen the most. This is reflected in the spectral SINR where SRN achieves 20 dB better performance than SPN and SLN. In the last scenario IV (three 20 MHz LTE carriers), the spectral congestion grows past the 40% threshold in the case-based reasoner which results in the SDRadar switching from Notch to Avoid. After switching to avoidance, the restless bandit model consistently converges to SRA in this scenario. There is a brief transition period from CPIs 220 to 240 where the MCR must adjust to the abrupt change in RFI and obtain an accurate spectral congestion measurement. After this, the spectrum classifier includes Avoid instead of Notch in the CR strategy.
Figure 8.16: Average spectrum sharing metrics for each individual RFI scenario. (a) shows the spectral SINR in dB. (b) shows average collision rate. (c) shows the average bandwidth utilization. Each metric is presented as an average over each RFI scenario.

Lower collision rates shown in Figure 8.16b correspond to a lower impact on coexisting LTE emitters due to the radar. Similarly, high SINRs in Figure 8.16a correspond to low collision rates. Figure 8.16c shows that the MCR can maximize bandwidth utilization while maintaining a high SINR in lower congestion scenarios.
and II. As congestion increases in III and IV, the MCR begins to sacrifice bandwidth to maintain a positive SINR. These results demonstrate the efficacy of metacognition with notching aided by case-based reasoning and the restless bandit model.
Conclusions and Future Work

9.1 Conclusions and Summary

Real-time spectral prediction via a stochastic model has been demonstrated to allow cognitive SDRadars to efficiently share the spectrum. This real-time implementation was extended to multiple CR strategies controlled by metacognition. The metacognitive radar allows for robust spectrum sharing in rapidly changing EMEs by selecting the optimal CR strategy. This was similarly demonstrated in real-time using COTS SDRadar hardware. The main contributions of this work include the following:

1) Present a stochastic model-based prediction scheme for radar spectrum sharing with prediction threshold optimization.

The problem of spectrum crowding and the increasing demand for wireless throughput was introduced in Chapter 1. This was followed by a discussion of cognitive RF and spectrum sharing as a solution to this problem. Chapter 2 describes the cognition from a neuroscience perspective followed by the application of cognition to machines and RF technology. Here, we showcase that while there is much literature on cognitive spectrum sharing for communication networks, spectrum sharing for cognitive radar is still an emerging concept. Next, Chapter 3 introduces a brief history of radar, basic pulse-Doppler radar processing and target detection, followed by an analysis of the impacts of RFI and spectrum sharing on radar processing. This portion demonstrates that RFI can result in false alarms and reduced target detectability.
As a solution to these negative effects on radar, Chapter 4 outlines an architecture to perform sense-predict-avoid for spectrum sharing on a cognitive radar. Basic energy detection for spectrum sensing is introduced. Next, a general stochastic model for RF activity is described using both parametric and empirical models. Predictive processing that leverages these models is discussed which estimates the likelihood of future spectrum availability. The next section describes the real-time implementation of spectral prediction with SDRadar hardware.

2) Characterize the performance of a real-time SDR implementation of spectral prediction with spectrum sharing cognitive radar.

Chapter 5 describes metrics to characterize the performance of a spectrum sharing radar as well as a comprehensive set of RFI scenarios to demonstrate its capability. An analysis of real-time closed-loop experiments with the cognitive SDRadar demonstrates the system’s ability to accurately predict RFI ranging from deterministic to moderately random over time. As the EME becomes increasingly random, the cognitive radar is able to sacrifice effective bandwidth to mitigate collisions with RFI. Predictive spectrum sharing was shown to improve the radar’s detectable range up to \(\sim 45\%\) in the test cases.

3) Compare the performance of spectral prediction to alternative cognitive radar strategies for spectrum sharing.

The SPA implementation was compared against and combined with alternative cognitive radar implementations in Chapter 6. Compared to an MDP model for prediction (SLA), SPA demonstrates better performance in intermittent RFI while SLA is able to better maintain bandwidth in realistic RFI scenarios. The efficacy of combining spectral prediction with notching (SPN) was similarly demonstrated in real-time with various RFI scenarios. SPN improves bandwidth utilization while maintaining prediction accuracy despite additional processing latency.

4) Describe a metacognitive control architecture based on case-based reasoning and the restless bandit problem to optimize CR strategy selection.
Given a variety of cognitive radar implementations, each has been demonstrated to be well suited to particular EME conditions. Chapter 7 presents a merged SDRadar architecture which utilizes the bio-inspired concept of metacognition to optimally select the best CR strategy for the EME conditions. This architecture employs a restless bandit model to trade off the exploration and exploitation of each CR strategy given information describing the EME. A real-time implementation of the metacognitive SDRadar is demonstrated on set sequences of rapidly changing RFI scenarios in Chapter 8.

5) Pose the Jensen-Shannon divergence as a metacognition metric to describe target detection quality and distortions due to intra-CPI adaptation.

In Chapter 7, this work introduces a Jensen-Shannon divergence metric to characterize target detection performance with no \textit{a priori} knowledge of the environment. Radar simulations verified that this metric provides a strong indicator of false alarm rate and moderate indication of range-Doppler SINR. Similarly, experiments in Chapter 8 optimize the metacognitive radar performance to minimize false alarm rate in changing RFI scenarios. The results suggest that the JS divergence is a strong indicator of distortions due to intra-CPI waveform adaptation.

6) Assess the effectiveness of real-time metacognitive radar implementations to optimize performance in dynamic EMEs.

Chapter 8 presents a few implementations of metacognitive radar architectures and assesses their ability to optimize performance in dynamic EMEs. Four restless bandit models are compared with spectrum sharing and target detection reward feedback. These models choose between SRA, SPA, and SLA over a sequence of RFI scenario in real-time. The performance bounds for these RFI scenarios are characterized for each CR strategy. From this comparison, sliding window UCB1-tuned was shown to best maximize performance for both reward types. Later sections analyze, a classifier-aided explore-first implementation that includes notching in the possible CR strategies. This demonstrates the efficacy of the classifier to switch between notching and avoidance as well as optimize spectrum sharing performance.
9.2 Future Work

Given a proof-of-concept for metacognitive radar and spectral prediction, further developments could improve system robustness and better define use cases. The spectral prediction process could potentially be improved by leveraging the stochastic model as features to a machine learning algorithm. One particular case could employ a partially observable MDP (POMDP) where the stochastic model characterizes the likelihood of being in some combination of RF sub-band states over time. Then, the MDP model (introduced in Section 6.2) factors this likelihood into the prediction.

Additional metrics could better characterize the impacts of spectrum sharing with both SPA and metacognition on communication networks. The performance of communication networks are often evaluated in terms of BER. This can be characterized by over-the-air tests with the cognitive SDRadar and a 5G network. These over-the-air tests may vary the distance and power between the SDRadar and 5G systems to obtain a relationship between SINR and BER.

In Chapter 4, the spectrum sensing and prediction models only account for DSA in the time and frequency domains. The cognitive radar would benefit from the ability to perceive RF activity in the spatial domain. By employing an array of antennas with beam steering, the radar could sense RF activity in the angular domain by periodically scanning the EME. This would present a simple extension of SPA by increasing the dimensionality of the model. The angular domain could be discretized into discrete sub-spaces, much like the spectrum. Each angular sub-space would contain a unique set of spectral sub-bands and stochastic model parameters. Such an implementation would require significantly more processing resources to maintain real-time requirements and perform algorithms to control beam steering. Similarly, the possible antenna array would require an RF system with additional receive channels and digital RF front-end components.

Here, the presented metacognitive radar architecture acts as a simple proof-of-concept with significant areas for development. Other aspects of metacognition, such as metalearning, could be implemented. One example of metalearning includes tuning machine learning algorithms hyperparameters or models autonomously to optimize learning from a biased or limited dataset. For spectrum sharing cognitive radar, this may include optimally adapting the parameters of perception...
algorithms for prediction or the MDP. Another realization of this could consist of excluding certain spectral sub-bands from each CR strategy’s action space. Given some classifier that identifies harmful RFI or unsuitable spectral conditions, the metacognitive engine could adapt the action space of each CR strategy.

As discussed in earlier sections, adapting radar waveforms from pulse-to-pulse results in distortions during Doppler processing. This work measures this distortion effect using the JS divergence from target detection distributions. Others have proposed using Richardson-Lucy deconvolution to correct for these distortion effects during spectrum sharing [53,54]. Future work may consider pairing real-time metacognitive SDRadar with such Doppler distortion correction methods. Richardson-Lucy deconvolution is an iterative process which requires some model of the radar’s point spread function. This would demonstrate a complete system which maximizes spectrum sharing performance, given additional processing resources.

Beyond this proof of concept for spectrum sharing MCR, the potential of this technology could be further demonstrated through integration with additional hardware configurations. A distributed network of RF sensors could provide for more detailed awareness of the EME. This network of sensors could perform metacognitive and spectral perception processing to provide detailed information to a radar system over a large spatial area. Such technology would improve a radar’s ability to anticipate sources of harmful RFI in operating environments.
Appendix

Performance Metric Simulation
Parameters

Here, the pulse pattern sequences for the MATLAB simulation described in Section 7.5.2 are defined. Each of these 12 sequences are tested alongside the RFI scenarios and each transmit power level described in Section 7.5.2. For the simulation, a radar transmits 400 LFM chirp pulses to generate a CPI for Doppler processing in the presence of RFI. Each sequence below describes the order in which the start $f_{seq,S}[i]$ and end $f_{seq,E}[i]$ frequencies adapt from pulse-to-pulse for each $i^{th}$ step in the sequence. If the end of the sequence is reached, then the pattern repeats unless the simulation reaches the end of a CPI. In Pulse Pattern 4, $U(0,1)$ refers to a sample from the uniform distribution on the interval [0,1]. These patterns are defined via set-builder notation where each step in the sequence is conditioned on the index $i$. 

128
Table A1: Simulation pulse pattern parameters

<table>
<thead>
<tr>
<th>Pulse</th>
<th>$f_{\text{seq.S}}[i]$</th>
<th>$f_{\text{seq.E}}[i]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern 1</td>
<td>${i = 1 : -45}$</td>
<td>${i = 1 : 45}$</td>
</tr>
<tr>
<td>Pattern 2</td>
<td>${i = 1 : -22.5}$</td>
<td>${i = 1 : 22.5}$</td>
</tr>
<tr>
<td>Pattern 3</td>
<td>${i \in {1, 2, \ldots, 10} : -30 \text{ if } i \leq 9 \text{ and } -20 \text{ if } i = 10}$</td>
<td>${i \in {1, 2, \ldots, 10} : 10 \text{ if } i \leq 9 \text{ and } 20 \text{ if } i = 10}$</td>
</tr>
<tr>
<td>Pattern 4</td>
<td>${i \in {1, 2, \ldots, 400} : (-10)U(0, 1)\forall i}$</td>
<td>${i \in {1, 2, \ldots, 400} : (10)U(0, 1)\forall i}$</td>
</tr>
<tr>
<td>Pattern 5</td>
<td>${i \in {1, 2, \ldots, 400} : -40(1 - i/400)\forall i}$</td>
<td>${i \in {1, 2, \ldots, 400} : 40(i/400)\forall i}$</td>
</tr>
<tr>
<td>Pattern 6</td>
<td>${i \in {1, 2, \ldots, 4} : -35 \text{ if } i \leq 2 \text{ and } -15 \text{ if } i &gt; 2}$</td>
<td>${i \in {1, 2, \ldots, 4} : -5 \text{ if } i \leq 2 \text{ and } 25 \text{ if } i &gt; 2}$</td>
</tr>
<tr>
<td>Pattern 7</td>
<td>${i \in {1, 2, \ldots, 10} : -40 \text{ if } i \leq 7 \text{ and } -10 \text{ if } i &gt; 7}$</td>
<td>${i \in {1, 2, \ldots, 10} : 0 \text{ if } i \leq 7 \text{ and } 30 \text{ if } i &gt; 7}$</td>
</tr>
<tr>
<td>Pattern 8</td>
<td>${i \in {1, 2, \ldots, 10} : -30 \text{ if } i \leq 5 \text{ and } -25 \text{ if } i &gt; 5}$</td>
<td>${i \in {1, 2, \ldots, 10} : 10 \text{ if } i \leq 5 \text{ and } 15 \text{ if } i &gt; 5}$</td>
</tr>
<tr>
<td>Pattern 9</td>
<td>${i \in {1, 2, \ldots, 20} : -30 \text{ if } i \leq 10 \text{ and } -35 \text{ if } i &gt; 10}$</td>
<td>${i \in {1, 2, \ldots, 20} : 10 \text{ if } i \leq 10 \text{ and } 15 \text{ if } i &gt; 10}$</td>
</tr>
<tr>
<td>Pattern 10</td>
<td>${i \in {1, 2, \ldots, 200} : -20 \text{ if } i \leq 170 \text{ and } -11 \text{ if } i &gt; 170}$</td>
<td>${i \in {1, 2, \ldots, 200} : 0 \text{ if } i \leq 170 \text{ and } 9 \text{ if } i &gt; 170}$</td>
</tr>
<tr>
<td>Pattern 11</td>
<td>${i \in {1, 2, \ldots, 200} : -20 \text{ if } i \leq 100 \text{ and } 0 \text{ if } i &gt; 100}$</td>
<td>${i \in {1, 2, \ldots, 200} : 0 \text{ if } i \leq 100 \text{ and } 20 \text{ if } i &gt; 100}$</td>
</tr>
<tr>
<td>Pattern 12</td>
<td>${i \in {1, 2, \ldots, 100} : -40 \text{ if } i \leq 70 \text{ and } -10 \text{ if } i &gt; 70}$</td>
<td>${i \in {1, 2, \ldots, 100} : 0 \text{ if } i \leq 70 \text{ and } 30 \text{ if } i &gt; 70}$</td>
</tr>
</tbody>
</table>
Bibliography


Vita
Jacob A. Kovarskiy

EDUCATION

The Pennsylvania State University, University Park, PA, USA
Ph.D. Electrical Engineering, May 2021
B.S. Electrical Engineering, December 2015

WORK EXPERIENCE

Army Research Laboratory, Adelphi, MD, USA
Research Fellow, Summer: 2016, 2017, 2018 and 01/2019 - 05/2021

• Led the integration of algorithms from several universities onto a single real-time prototype
• Applied learning-based RF spectral prediction methods to spectrum sharing cognitive radar
• Developed a metacognition framework for spectrum sharing via a multi-armed bandit model

ACADEMIC RESEARCH EXPERIENCE

Penn State College of Engineering, University Park, PA, USA
Graduate Research Assistant, August 2016 - December 2018

• Investigated interference detection, prediction and avoidance for radar spectrum sharing
• Measured and surveyed the local RF spectrum to analyze realistic interference scenarios

Graduate Teaching Assistant, January 2016 - May 2016

• Instructed and graded lab sessions for students in EE 210 – Circuits and Devices

Undergraduate Research Assistant, August 2014 - May 2015

• Researched adaptive filter designs and wavelet-based denoising for fetal electrocardiogram extraction