Spectrum Learning and Access for Cognitive Satellite Communications under Jamming

Yi Shi and Yalin E. Sagduyu
Intelligent Automation, Inc., Rockville, MD, USA
Email: \{yshi, ysagduyu\}@i-a-i.com

Abstract—This paper presents a robust dynamic spectrum access (DSA) decision making framework for a satellite communication (SATCOM) system with primary users (PUs) and secondary users (SUs) operating in the presence of cognitive jammers. Different levels of uncertainty are considered for the channel availability under SATCOM delays and jamming effects. Spectrum uncertainty is quantified regarding the parameter estimation for Markov model of spectrum occupancy and the prediction of future channel states under SATCOM delays. Based on these uncertainty models, the optimal spectrum sensing approach is used to identify available channels. Both offline and online algorithms are used for DSA. Offline algorithm first learns spectrum dynamics and then applies DSA, whereas online learning repeats this process and adapts to fast spectrum dynamics. Results show performance gains of DSA over the case without considering spectrum uncertainty.

Index Terms—Dynamic spectrum access; spectrum sensing; cognitive radio security; satellite communication; LEO satellites; jamming; spectrum uncertainty.

I. INTRODUCTION

We consider modeling the uncertainty and analyzing its impact in satellite communication (SATCOM) from the following perspectives: 1) uncertainty modeling for a low Earth orbit (LEO) system [1], [2] with primary users (PUs), secondary users (SUs), and jammers, 2) algorithms for spectrum sensing and dynamic spectrum access (DSA). Under DSA, each user selects which channel to transmit or interfere at. This system model is built on the notion of cognitive radio networking [3]–[6].

The target application is that ground PUs and SUs share the channels to access LEO satellites, while ground jammers attempt to interfere PU and SU transmissions. Each channel has random PU traffic, which can be modeled by a two-state Markov model. Each SU can sense and access idle channels not used by PUs while a jammer can sense and jam busy channels with PU/SU transmissions. There is no synchronization among PUs, SUs, and jammers. The first goal is to build uncertainty models for SATCOM regarding parameter estimation for spectrum occupancy (based on Markov models) and prediction of future channel states (idle or busy) under SATCOM delays. Based on these uncertainty models, we then consider the optimal spectrum sensing to identify the channel idle/busy probability under jamming.

With sensing results and acknowledgments, the next step is to design and implement a DSA decision making framework with optimization modules for each type of users with both offline and online algorithms. To improve security in such a cognitive radio network, the design of DSA scheme for SUs should consider jamming efforts by cognitive jammers. Under an offline algorithm, a user first senses channel status and then makes DSA decisions for the remaining time. Offline algorithm works well when the PU traffic does not change frequently and thus the channel status usually does not change within a time frame. When the PU traffic changes frequently, a SU/jammer needs an online algorithm to adjust transmission/jamming promptly. In particular, a SU can maintain a channel status Markov model and then make DSA decisions based on the predicted channel status under jamming.

Our contribution can be summarized as follows:

- Building uncertainty models for SATCOM under jamming,
- Designing spectrum sensing algorithms,
- Designing a DSA decision making framework with both offline and online algorithms under jamming.

The rest of the paper is organized as follows. Section II presents the system model. Section III introduces the uncertainty models. Section IV describes spectrum sensing. Sections V and VI present offline and online algorithms, respectively. Section VII concludes the paper.

II. SYSTEM MODEL

We consider DSA for satellite-ground transmissions in a LEO system. LEO satellites can use various bands for communications, e.g., Ka, K, L, S, or C bands. LEO satellites are not stationary with respect to a fixed ground user on the Earth’s surface. The satellite ground track speed is much greater than Earth’s rotation speed and the ground user speed. In the LEO system, handovers occur frequently due to the movement of satellites [1]. The orbit shell can be defined by a matrix of spacecraft latitude and longitude positions. At each position on the orbit shell, the database of cities is searched for those visible from the satellite using their approximated geographic size, which may not be accurate or complete. Therefore, we consider uncertainty regarding the mobility and databases.

A. LEO System

A LEO satellite has an altitude between 200 km and 2000 km with one-way delay 2 ms to 5 ms [2]. Given that the Earth’s equatorial radius is \( R_E = 6378 \) km, the radius of a LEO satellite is between 6578 km and 8378 km. Since we have \( V = \frac{g \times R_E}{2} \)
Transmit power at a ground user can be adjusted from 100 mW to 10 kW. The noise power is $P_n = kT_{\text{noise}}$, where Boltzmann’s constant $k = 1.39 \cdot 10^{-23}$ J/K = 228.6 dBW/KHz, $\tau$ is the physical temperature of source in kelvin degrees, $W$ is bandwidth in Hz. The achieved data rate is determined by a particular system (e.g., channel gain, noise, and modulation and coding) and ranges from 10 kb/s to 64 mb/s.

### B. Primary User (PU) Model

We assume that each PU is assigned one channel and can access this channel without consideration on SUs. The PU traffic follows a continuous Markov on-off process with two states $S_t(t) = 0$ (idle) or 1 (busy), where channel $m = 1, \cdots, M$ and time $t \geq 0$. These Markov processes on $M$ channels are jointly independent. In particular, for channel $m$, the sojourn times in the busy and idle states are exponentially distributed with rates $\lambda_{1,m}$ and $\lambda_{0,m}$, respectively. There is certain equivalence between DSA in unslotted primary systems and in slotted primary systems [9]. With this equivalence, we assume that the PU traffic may change from busy to idle (or from idle to busy) for every $T$ time slots, where one time slot corresponds 1 ms. Note that the PU traffic are not synchronized over $M$ channels. The transition probabilities are $p_{00}^{m}, p_{01}^{m}, p_{10}^{m}, p_{11}^{m}$, where $p_{ij}^{m}$ is the transition probability from state $i$ to state $j$ on channel $m$. This model was used to represent traffic dynamics of primary users in cognitive radio network systems [7]. These parameters are unknown to SUs and jammers. Parameter $T$ reflects whether the PU traffic is fast changing or slow changing, which in turn calls online or offline DSA decision making, respectively, at SUs and jammers.

### C. Secondary User (SU) Model

We assume that a SU is always backlogged and looks for opportunistic access in PU channels. A SU needs to find idle channels (not used by PUs) and then use them to transmit its data. Once data is transmitted, it will receive ACK or negative acknowledgement (NAK) as a feedback. The DSA decision making can be either offline or online. Under an offline decision making scheme, a SU first senses channels and then decides on which available channels to transmit. Under an online decision making scheme, a SU maintains a channel busy/idle model and makes DSA decisions based on this model. This model will be continuously updated to characterize the most recent channel states. Under an offline scheme, a SU can sense a PU or another SU transmission by ACK/NAK. When a control packet is sensed on one channel, this channel is busy.

A PU/SU ground transmitter uses a directional antenna to communicate with the satellite. Although majority power is transmitted in the main lobe towards the satellite, there is some power in sidelobes. A SU in a sidelobe can detect channel busy/idle status for uplink traffic. However, such information may not always be available and add to uncertainty, e.g., a SU is not in a sidelobe or power in sidelobe is much smaller than interference. Under an online scheme, a SU can start an initial channel busy/idle model and keep updating the transition probabilities based on received ACK/NAK for its transmissions. That is, if an ACK is received, the corresponding channel would be idle if this SU did not transmit. As a result, $p_{00}^{m}$ and $p_{11}^{m}$ can be increased. On the other hand, if an NAK is received, then the corresponding channel is busy even if this SU did not transmit. Thus, $p_{01}^{m}$ and $p_{10}^{m}$ can be increased.

### D. Cognitive Jammer Model

A jammer needs to identify busy channels by sensing ACKs and jam PU and SU transmissions on these channels. Once a jammer transmits on a channel, it will check whether there is an NAK as feedback. If there is an NAK, then one transmission is jammed. Otherwise, the jammed channel is idle and no transmission is jammed. The jamming decision making can be either offline or online. Under an offline decision making scheme, a jammer maintains a channel busy/idle model and makes jamming decisions based on this model. This model will be continuously updated based on NAKs to characterize the most recent channel states.

### III. Uncertainty Quantification

In SATCOM systems, uncertainty can be caused by the environment (e.g., channel estimation error), regulation (databases, interference between PUs and SUs), uncontrollable user effects (persistent/adaptive jamming, traffic, mobility, synchronization), and prediction (false positive/negative, delay, prediction on future events).

#### A. Markov Model Parameter Estimation

We now consider particular uncertainties in SATCOM. First, we consider uncertainty regarding Markov model parameter estimation. In Section II-B, we model the channel states (idle or busy) to follow a Markov chain with transition probabilities $p_{ij}$ between channel states $i$ and $j$ (note that we omit superscript $m$ to simplify notation), where state 0 means idle and state 1 means busy. The typical assumption is that the transition probabilities of this Markov chain are accurately
known a priori and these probabilities are used to track channel states often more reliably compared to the case of independent and identically distributed (i.i.d.) assumption of channel state evolution (where only energy detector is used without tracking channel state transitions). In reality, these probabilities must be learned. We design the algorithm to learn channel transition state probabilities by applying energy detector (combined with power estimation). In this learning mechanism, we observe energy detector outcomes and build the Markov model by tracking the number of detected transitions among “idle” and “busy” states. Next, we develop the algorithm to reliably learn these probabilities over time. We first determine Markov chain parameters based on the detection results and then predict the future status. The detection results are either idle (0) or busy (1). We count the number of changes from idle to idle, idle to busy, busy to idle, and busy to busy. Denote these number as \( n_{00}, n_{01}, n_{10}, n_{11} \), respectively. Then we estimate the transition probabilities by

\[
\hat{p}_{ij} = \frac{n_{ij}}{n_{i0} + n_{i1}} \quad (0 \leq i, j \leq 1).
\]  

Figure 1 shows the estimated Markov chain parameters (state transition probabilities) over time slots for a channel with Markov model parameters \( p_{00} = 0.2, p_{01} = 0.8, p_{10} = 0.3, p_{11} = 0.7 \) (shown as the dashed lines). We find that these estimated parameters quickly approach the actual values.

### B. Prediction of Future Channel States

We also use estimated channel state transition probabilities to predict future channel states. This capability is needed because the channel states change over time and due to propagation delays in SATCOM systems, the system can only detect states in previous time. In addition, the system will make transmission decisions based on spectrum sensing results and the effect will be observed later at the corresponding receiver. Therefore, we expect significant delay between “what the system should sense” and “what the system actually senses”. This effect is also present (even without propagation delay) if there are multiple frequencies and a user cannot observe all channels at a given time such that it is necessary to predict spectrum sensing results on some channels. Thus, we design and implement the algorithm to project spectrum sensing results into future by evolving them according to the estimated Markov chain model. Suppose we need to predict channel status at time \( t + D \) given status at time \( t \), where \( D \) is the number of possible status changes that depends on sensing and transmission delays in SATCOM system. We first build an initial distribution, i.e., if the current channel status is idle, we have \( p_0^t = 1, p_1^t = 0 \), otherwise we have \( p_0^t = 0, p_1^t = 1 \). We then update the channel status by

\[
p_i^{t+1} = p_0^{t} p_j^t + p_1^{t} p_{ij} \quad (t + 1 \leq i \leq t + D, j = 0, 1).
\]  

Figure 2 shows the probability of prediction errors versus prediction delay (measured by time slots). If the arbitrary selected parameters are very different from the actual parameters, prediction with those arbitrary parameters is worse than no prediction. As the estimated parameters are close to the actual values, prediction with those estimated parameters is close to the actual values.

### Table I: Prediction Parameter Setting

<table>
<thead>
<tr>
<th>Actual value</th>
<th>Estimated value</th>
<th>Arbitrary Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_{00} )</td>
<td>0.3</td>
<td>0.2912</td>
</tr>
<tr>
<td>( P_{01} )</td>
<td>0.7</td>
<td>0.7088</td>
</tr>
<tr>
<td>( P_{10} )</td>
<td>0.4</td>
<td>0.3925</td>
</tr>
<tr>
<td>( P_{11} )</td>
<td>0.6</td>
<td>0.6075</td>
</tr>
<tr>
<td>( P_0 )</td>
<td>0.3636</td>
<td>0.3564</td>
</tr>
<tr>
<td>( P_1 )</td>
<td>0.6364</td>
<td>0.6436</td>
</tr>
</tbody>
</table>
actual parameters, the prediction with estimated parameters achieves the same performance (because the final decision is to compare the probability of being in one state is greater than 0.5). With estimated parameters, the prediction performance can be improved compared to the approach that simply uses the current channel status as the future status. Moreover, the prediction error probability remains the same as the delay increases (the probability of hitting the correct state remains above 0.5).

In summary, our results showed that this prediction approach performs significantly better (in terms of average detection error) compared to the naive approach of using current spectrum sensing results for future events. Especially, we showed that the performance gap potentially increases with the increasing delay.

IV. SPECTRUM SENSING

Once the channel state transition probabilities are learned, the next step is to determine how they can be used to improve channel state detection (we will later show how we can use them also for prediction of future channel states). Energy detector [7] is optimal for Gaussian signals when the objective is to satisfy a target detection (or misdetection) probability. A better approach is to use the optimal channel state detector using Maximum A posteriori Probability (MAP) detection. The resulting detector is still an energy detector; however, the threshold is optimally selected to minimize the average detection error probability. We analytically derive this threshold in terms of values of prior state probabilities, signal and noise power terms.

Our objective is to minimize average probability of error

\[ p_E = P(H_0)p_F + P(H_1)p_M, \]  

(3)

where \( p_F \) is the false alarm probability and \( p_M \) is the misdetection probability. We consider the knowledge on prior probabilities of each hypothesis (i.e., channel state). For instance, we can use the stationary probabilities of states in Markov chain and use it during the decision process. The received signal \( Y_i \) (ith sample) is complex Gaussian \( CN(0,\sigma_0^2) \) under hypothesis \( H_0 \) and is \( CN(0,\sigma_0^2 + \sigma_1^2) \) under hypothesis \( H_1 \). The optimal detector follows from hypothesis test, where we perform MAP detection:

\[ P(H_0) \prod_i P(Y_i|H_0) \leq P(H_1) \prod_i P(Y_i|H_1) \]  

(4)

For Gaussian signal and noise, MAP rule can be rewritten as:

\[ P(H_0) \prod_i \frac{e^{-\frac{1}{2} \sigma_0^2 |Y_i|^2}}{\sigma_0 \sqrt{2\pi}} \leq P(H_1) \prod_i \frac{e^{-\frac{1}{2} \sigma_1^2 |Y_i|^2}}{\sqrt{\sigma_0^2 + \sigma_1^2} \sqrt{2\pi}}. \]  

(5)

This optimal rule leads to energy detection

\[ \sum_{i=1}^{N} |Y_i|^2 \leq \gamma, \]  

(6)

where the threshold is

\[ \gamma = 2\frac{\sigma_0^2(\sigma_0^2 + \sigma_1^2)}{\sigma_1^2} \ln \frac{P(H_0)\sqrt{\sigma_0^2 + \sigma_1^2}}{P(H_1)\sigma_0^2}. \]  

(7)

This detector explicitly uses the knowledge on prior probabilities on the hypothesis.

This method uses only stationary state probabilities without tracking the Markov states over time. Alternatively, we can track these states and set the threshold depending on the detected state in the previous time slot. If we detect the state to be in \( H_0 \) at time \( t \), then at time \( t+1 \), the probability of being in state \( H_0 \) is \( 1-p_{01} \) and the probability of being in state \( H_1 \) is \( p_{01} \). So we use the energy detector threshold

\[ P(H_0)=1-p_{01}, P(H_1)=p_{01} \]  

and the energy detector threshold is given by

\[ \gamma = 2\frac{\sigma_0^2(\sigma_0^2 + \sigma_1^2)}{\sigma_1^2} \ln \frac{(1-p_{01})\sqrt{\sigma_0^2 + \sigma_1^2}}{p_{01}\sigma_0^2}. \]  

(8)

Similarly, if we detect the state to be in \( H_1 \) at time \( t \), the energy detector threshold is given by

\[ \gamma = 2\frac{\sigma_0^2(\sigma_0^2 + \sigma_1^2)}{\sigma_1^2} \ln \frac{p_{10}\sqrt{\sigma_0^2 + \sigma_1^2}}{(1-p_{10})\sigma_0^2}. \]  

(9)

We alternate between two thresholds as we track and detect the states over time. The state transition probabilities are not known \( a \ priori \) and therefore we use the estimated values (as outlined in previous section). We compared the performance (in terms of \( p_E \)) of the energy detector (with power estimation) with the optimal energy detector. The optimal energy detector provides a lower bound on \( p_E \) than the energy-based detector. As shown in Fig. 3, the energy detector with power estimation provides a suboptimal performance with a gap less than 0.02
by learning the signal and noise power, and adapting the threshold for the energy detection.

We consider the performance of energy detector with the optimal threshold using the learned parameters $\sigma_0^2, \sigma_1^2, P(H_0)$, and $P(H_1)$. The estimated parameters are learned based on the average and the variance values of the historic samples. In particular, the system learns parameters in the first 1500 slots by applying energy detector. For the rest of the time, we apply the two different schemes: 1) energy detector or 2) optimal detector. Figure 4 shows that the performance of the optimal detector is better than that of the energy detector with power estimation. The performance gain is rather small and this points at the effective use of energy detector in joint power estimation and energy detection.

V. OFFLINE LEARNING AND DECISION MAKING FOR DSA

When the system status is slowly-varying, we can use the offline mechanism for learning and decision making, where we model the uncertainty due to other users’ behavior with Markov model-based approach. There is a trade-off between time allocated for spectrum sensing and access [8]. We consider learning schemes to carefully balance this trade-off.

A. Offline Mechanism

Both SUs and jammers first perform sensing, and then perform transmission or jamming. The entire time of one period is $L$ time slots. The sensing time on one channel is one. The number of sensed channels can be up to $U_{sense}$. There is no synchronization among SUs and jammers. SU transmissions may be collided by PU, other SUs or jammed by jammers. In this case, there is no throughput. Otherwise, the achieved throughput is a function of transmit power. Each SU uses the average throughput on a channel over $L$ time slots as its reward by using this channel. Each jammer may jam PU or SU in a time slot. It uses the number of jammed slots on a channel over $L$ time slots as its reward by jamming this channel. Offline algorithms work well when PU traffics do not change much during one sensing and transmission period, i.e., we should have $T > L$. We choose $T = 2L$.

SUs. Denote $N_{SU}$ as the number of SUs. Each SU decides on which channels to sense and how to allocate transmit power.

Initially, each channel is assigned a default reward value $r_0$ and a SU randomly selects $U_{sense}$ channels to sense. Once transmitted, SU $i$ updates the reward $r_{if}$ for a channel $f$ by the achieved throughput $c_{if}$ as follows.

$$r_{if}(t) = (1 - \beta)r_{if}(t - L) + \beta c_{if}(t), \quad (10)$$

where weight $\beta$ is within $[0, 1]$ and $c_{if}(t)$ is the achieved throughput during $[t - L + 1, t]$ (to be discussed later). Note that if a channel is not used for transmission, its throughput may be set as $0.4 \times E[c_{if}(t)]$, where the expectation is performed over all transmitted channels, to update its reward. Denote the number of sensed channels as $M_{sense}$. Then the remaining time for transmission is $L - M_{sense}$. In the next round, a SU chooses channels based on $r_{if}(t)$ values. Note that a SU may not always choose $U_{sense}$ channels, if there are no enough channels with good rewards, e.g., we may not transmit on a channel with a reward less than a threshold $0.1 \times E[c_{if}(t)]$.

The sensing process determines the idle/busy status, channel gain, and noise with uncertainty. Then SU $i$ needs to allocate its total transmit power $P$ on available channels, e.g., power can be equally split into selected channels. The throughput on a channel $f$ is

$$c_{if}(t) = \frac{L_{suc}}{L} W \log_2 [1 + \frac{g_f p_{if}}{N_f}], \quad (11)$$

where $L_{suc} \leq L - M_{sense}$ is the number of successful transmission slots that are not interfered by a PU/SU or a jammer, $W$ is bandwidth, $g_f$ is the channel gain, $p_{if}$ is the power on channel $f$, and $N_f$ is the noise on channel $f$.

Jammers. Denote $N_{jam}$ as the number of jammers. Each jammer decides which channels to sense and then jams busy channels. A jammer can jam up to $U_{jam}$ channels. Initially, each channel is assigned a default reward value 0.5 and a jammer randomly selects $U_{jam}$ channels to sense. Once jammed, jammer $j$ updates the reward $r_{jf}$ for a channel $f$:

$$r_{jf}(t) = (1 - \beta)r_{jf}(t - L) + \beta J_{jf}(t), \quad (12)$$

where $J_{jf}(t)$ is the number of jammed time slots during $[t - L + 1, t]$ divided by $L$. Note that if a channel is not selected for jamming, this value may be estimated as $0.4 \times E[J_{jf}(t)]$, where the expectation is performed over all jammed channels, to update its reward. Denote the number of sensed channels as $M_{sense}$. Then the remaining time for jamming is $L - M_{sense}$. In the next round, a jammer chooses channels based on $J_{jf}(t)$ values. The above reward for jammers only counts the number of channels jammed, not the throughput of transmissions jammed. This is because that although a jammer can sense NAK to identify the number of channels jammed, it cannot find the throughput when no jamming. The sensing process determines the idle/busy status with some uncertainty. Once channels are sensed, jammer $j$ selects up to
There are various uncertainties in the system. In particular, sensing results may not be perfect and we have the following scenarios: (1) A SU fails to detect a busy channel. (2) A SU claims an idle channel as busy. (3) A SU obtains a value for $g_i/N_f$ with some error. (4) A jammer fails to detect a busy channel. (5) A jammer claims an idle channel as busy.

We designed our sensing and decision algorithms above without consideration on uncertainty. We now handle uncertainty by considering reward in DSA decision making. That is, in addition to channel idle/busy probability, a SU/jammer also considers a reward to make a better decision. A SU can first select idle channels based on idle probability. It then identifies the worst channel (with the smallest reward) among all selected channels and the best channel (i.e., the channel with the largest reward) among all unselected channels. If the best unselected channel has a larger reward than that of the worst channel selected channel, then this SU may use the best unselected channel instead (with certainty switching probability). A jammer can perform a similar switching process.

**B. System Setting for Numerical Results**

We consider a ground-satellite system with two SUs, two jammers, and a set of LEO satellites such that every ground user can always communicate with one LEO satellite. If we only consider propagation gain between a ground user and a LEO satellite, this gain first become stronger over time and then become weaker. Typically, this propagation gain can be changing between $-20$ dB to $-30$ dB and the total communication time during an orbit cycle can be approximately 10 minutes. Due to the different location of SUs/jammers, propagation gain for each user may be different. We consider $M = 10$ channels, each with a two-state Markov model for busy/idle status. The number of time slots for a sensing and transmission period is $L = 10$, while the PU traffic may change for every $T = 20$ time slots. The number of sensed channels is upper bounded by $U_{sense} = 5$. The number of jammed channels is upper bounded by $U_{jam} = 3$. We assume the bandwidth of one channel is $W = 1$ MHz. The maximum transmit power on a channel is $p_{max} = 10$ W and the total transmit power on all channel is $P = 20$ W.

**C. Results for Offline Algorithms**

We implemented two offline algorithms. The first one does not consider uncertainty when making DSA decisions while the second one considers uncertainty when making DSA decisions. Figure 5 shows the throughput that can be achieved without consideration on uncertainty while Fig. 6 shows the throughput that can be achieved with consideration on uncertainty. We can see that with the consideration on uncertainty, throughput can be increased.

**D. The Impact of Uncertainty**

The show the impact of uncertainty, we increase the uncertainty level on sensing results. The sensed results in Section IV have error probability up to 10%. That is, when a channel is busy the mis-detection probability is up to 10% and when a channel is idle the false alarm probability is up to 10%. We now tune this bound as 20% and keep all other parameters unchanged. The results are shown in Fig. 7. With increased uncertainty, both SUs’ throughput are decreased.

**VI. Online Learning and Decision Making for DSA**

When stochastic system uncertainty is time-varying, offline approaches can result in large overhead for updates. In this case, online algorithms can be adapted to such fast dynamics. Online algorithm keeps learning the system status and change decisions accordingly.
A. Online Mechanism

We can again consider a slotted online problem for fast changing PU traffics. An extreme case is that a PU may change its status for every time slot by a Markov model. In this case, it is necessary for SUs/jammers to learn the Markov model parameters. We assume that transition probabilities $p_{01}^m$ and $p_{10}^m$ are close to 0 or 1. Otherwise, PU traffics are unpredictable and there is no way to protect PU transmissions.

**SU.** Each SU initially assumes a Markov model with the number of state transitions $n_{00}^m = n_{01}^m = n_{10}^m = n_{11}^m = 1$ for channel $m$. Then it calculates transition probabilities by (1). By assuming equal busy/idle probability 0.5 at the beginning, a SU predicts future channel status for each channel and update this status whenever there is new feedback. Once channels are selected, SU $i$ needs to allocate its total transmit power $P$ on these channels, e.g., via equal power allocation. If transmission on a channel $f$ is successful, the throughput $c_{if}(t)$ can be determined at receiver with no uncertainty. Initially, each channel is assigned a default reward value $r_0$. Once transmitted, SU $i$ updates the reward $r_{if}$ for a channel $f$ by the achieved throughput $c_{if}$ as follows.

$$r_{if}(t) = (1 - \beta)r_{if}(t-1) + \beta c_{if}(t).$$

Note that if a channel is not used for transmission, its throughput may be set as $0.4 \times E[c_{if}(t)]$ to update its reward.

**Jammer.** A jammer also assumes an initial Markov model with the number of state transitions $n_{00}^m = n_{01}^m = n_{10}^m = n_{11}^m = 1$ and equal initial busy/idle channel status. It will maintain a predicted channel status for each channel and update this status whenever there is a new sensing/feedback result. Then a jammer selects $n_{jam}$ channels with high busy probability to jam, updates both channel status and Markov model based on feedback. Initially, each channel is assigned a default reward value 0.5. Once jammed, jammer $j$ updates the reward $r_{jf}$ for a channel $f$ as follows.

$$r_{jf}(t) = (1 - \beta)r_{jf}(t-1) + \beta n_{jam},$$

where $J_{jf}(t)$ is the number of jammed slots. If a channel is not selected for jamming, this value may be estimated as $0.4 \times E[J_{jf}(t)]$ to update its reward.

**Uncertainty.** We need to consider various uncertainties for each SU/jammer, as we discussed in Section V-A. We compare two online algorithms, the one with and without consideration on uncertainty. Results by online algorithms without and with consideration on uncertainty are shown in Fig. 8 and Fig. 9, respectively. Throughput improves with consideration on uncertainty.
VII. CONCLUSION

We presented a robust DSA decision making framework for a SATCOM system for ground PUs and SUs communicating to LEO satellites in the presence of cognitive jammers. We considered different levels of uncertainty on the channel availability under SATCOM delays and jamming. We quantified spectrum uncertainty regarding parameter estimation for spectrum occupancy model and prediction of future channel states. We applied the optimal spectrum sensing to detect and track available channels. We used both offline and online algorithms: offline algorithm first learns spectrum dynamics and then applies DSA, whereas online learning repeats this process under fast spectrum dynamics. Results demonstrated the performance gains of DSA with consideration of spectrum uncertainty over the case without considering uncertainty in SATCOM.

ACKNOWLEDGMENTS

This research was partly supported by the United States Air Force under contract number FA9453-14-M-0004. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the United States Air Force.

REFERENCES