U. S. NAVAL RESEARCH LABORATORY
Washington, DC 20375-5320

NRL/FR/5520--20-10,403

S-Gen: Tool for Generating Distributed Spectrum Occupancy Data

CADEN J. PICI, SASTRY KOMPELLA, MATTHEW T. DILLON, AND JOSEPH A. MOLNAR

Networks and Communication Systems Branch
Information Technology Division

May 24, 2021

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From – To)
23-09-2020	NRL Formal Report	06/02/2020 - 09/30/2020

4. TITLE AND SUBTITLE

S-Gen: Tool for Generating Distributed Spectrum Occupancy Data

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Pici, Caden, J.
Kompella, Sastry
Dillon, Matthew, T.
Molnar, Joseph, A.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

1X10

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5520- -20-10,403

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Naval Research Laboratory
4555 Overlook Ave., SW
Washington, DC 20375

10. SPONSOR / MONITOR'S ACRONYM(S)

NRL NISE Category 1

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION A: Approved for public release, distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The purpose of this document is to outline the current development of a general tool, called S-Gen, for generating correlated spectrum occupancy data over a distributed sensor system. The tool is meant to model much of the physical layer effects on spatio-temporal spectrum occupancy and the associated losses. An object-oriented approach is taken to create sensors and emitters in a distributed environment. This allows for the generation of synthetic spectrum occupancy data that can be used for testing spectrum occupancy fusion models and algorithms.

15. SUBJECT TERMS

spectrum occupancy, sensor fusion, Dynamic Spectrum Access, distributed sensor measurements.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSION
a. REPORT	b. ABSTRACT	c. THIS PAGE			Caden J. Pici
U	U	U	SAR	18	19b. TELEPHONE NUMBER (Include area code) 202-767-5098

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
LaTeX

This page
intentionally
left blank

CONTENTS

1. INTRODUCTION 1
   1.1 Current and Future-Use Cases 1

2. MODEL OVERVIEW 2

3. DATA GENERATION 2

4. SENSOR MEASUREMENTS 5
   4.1 Additive Noise 6
   4.2 Threshold 7
   4.3 Sensor Definition Example 8

5. MATLAB SIMULATION 8

6. NEXT PHASE 11

7. CONCLUSION 11

REFERENCES 12

FIGURES

1 Energy detector 3

2 Global parameter example 4

3 Emitter definition example 4

4 Occupancy example 5

5 Sensor definition example 8

6 Network topology example 9

7 Two emitter occupancy data 10

8 Sensor to emitter path loss 10

S-GEN: TOOL FOR GENERATING DISTRIBUTED SPECTRUM OCCUPANCY DATA

1. INTRODUCTION

In cognitive radio (CR) and dynamic spectrum access (DSA) systems, accurate and robust perception of the radio environment is paramount in the development of a communication network that can effectively take advantage of spectrum vacancies. Much of the work around cognitive radio and monitoring occupancy for DSA has been focused around the many spectrum sensing signal processing approaches, as outlined in the review [1], with more focus on the network related issues a clear next step [2]. Combining local radio environment measurements to construct a more robust radio environment map becomes a new challenge towards the goal of a cognitive radio network. This will involve developing novel sensor fusion algorithms to take these local measurements from the distributed nodes, and combine them to account for both the spatial and temporal aspects of the radio environment. There is a considerable volume of literature and approaches to general sensor fusion, with reviews of the field such as in [3], [4], and [5].

The purpose of the work in this paper is towards the creation of a general tool for generating synthetic data for testing spectrum occupancy fusion models and algorithms over a distributed sensor system. The tool is designed to model much of the physical layer effects on spatio-temporal spectrum occupancy and the associated losses. In a wireless sensor network scenario, distributed nodes will perceive the radio environment differently, as their measurements are dependent on their own radio properties and tolerances, distances to emitters, and the propagation path to those emitters. Relying solely on local spectrum estimates to determine the presence of other users could provide consistent errors in the estimates, as some frequencies could be consistently attenuated more than others, or obstructions between a transmitter and radio could cause consistent missed detections of the signal. Additionally, taking advantage of the availability of external sensor data will enhance our chances of identifying signal presence and will aid in proper decision-making in these intelligent systems.

1.1 Current and Future-Use Cases

This tool in its current state is being used to generate spectrum occupancy data for a sensor fusion approach that uses the most recently received updates from the distributed sensors to come up with a fused occupancy decision for the network. Using the data from the transmitters combined with sensor information, a confidence weight is assigned to characterize the quality of information from that sensor in the fusion algorithm. Additionally, the algorithm works to locate the source of the emissions. Please refer to [6] for more information on this work.

Future work on this problem involves incorporating current and past measurements in a Bayesian filtering framework for generating the radio environment map. This involves implementing a Kalman filtering approach to sensor fusion. If the network can provide sufficiently timely updates of their local spectrum data, then the use of *a priori* data should be able to increase the estimation accuracy.

Manuscript approved May 24, 2021.

2. MODEL OVERVIEW

To allow for a wireless network that can consist of heterogeneous sensors and emitters, we take an object-oriented approach to developing the tool. The general idea is to be able to abstract out the important properties of a radio system, and to input those as parameters into the simulation objects. At this stage of the development, this will consist of creating emitter and sensor classes to allow for the generation of transmitted data, and the spatial dependence of the measurements, respectively.

The classes being developed are used to generate a network with an arbitrary number of sensors and emitters, with each being able to be defined with different operating parameters. This heterogeneous network of pseudo-radios, needs to come from class descriptions that take into account the important radio properties, such as transmission power, receiver sensitivity, operating band, etc., as well as associated methods, such as data generation, spectrum measurements, loss calculations, etc.

In summary, the current classes defined qualitatively are as follows:

- **Emitter class:** Important methods include those for data generation. An emitter instantiation will be responsible for generating occupancy data for a user-defined band/channels and location.
- **Sensor class:** An object created from this class should include fields for relevant sensor information. Methods associated with this class would include ways of generating local occupancy data from environmental readings, and to calculate losses from the emitter data.

Next, how some of these features will be implemented will be shown.

3. DATA GENERATION

As stated in the previous section, the purpose of an emitter object is to generate occupancy data. The emitter class will have properties associated with its location, transmission power, transmission channels, time steps and anything else important to defining a transmitter. The other important feature of the emitter class is its associated methods. At this point in the development, this just includes the tools and procedure for generating data. There are a number of ways data generation can be interpreted, so to further specify what is being transmitted, it is beneficial to consider what the data needs to look like from the perspective of a sensor. Specifically, at this point, we are interested in what the sensors see in order to test and evaluate data-fusion algorithms. The primary purpose of the measurements is energy detection, i.e., determining whether or not a signal is present. To do this, it is unnecessary to do the front-end signal processing and to generate I/Q signals, at least at this point. Looking at Fig. 1, this is a typical block diagram of an energy detector at a receiver. The data that is needed corresponds to the red, dotted line, which is, in effect, just the energy in a bin of frequencies determined by the band pass filter and window length. Thus, at the transmitter, it is only necessary to generate these bins of occupancy at some transmission power across the desired channels. Potential future enhancements could include data generation for classification and feature detection. In that case, the model would need to include true signal generation and processing.

To create the transmitted data, it is first needed to instantiate an emitter object. This, in effect, creates a transmission radio utilizing the tools and parameters provided in the emitter class. The user would need to define, at minimum, the following:

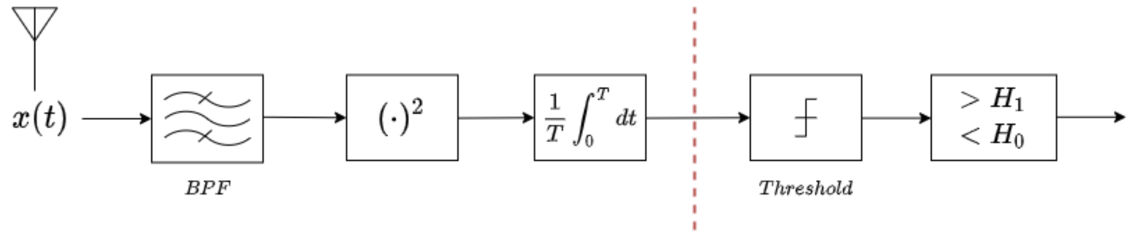


Fig. 1— Energy detector block diagram. Dotted, red line signifies where data is generated.

Input:

- Location: The node location coordinates (e.g. latitude and longitude, or location on coordinate axis)
- Frequency band: The frequency band, or bands, over which the emitter is transmitting data. This is later used in path loss calculations.
- Transmitter power: The power at the output of the antenna in units of dBm. Received power will be found at a sensor based on this, and the calculated losses. This will be shown in next section.
- Channels: The possible frequency band used by all emitters and sensors is divided into known channels. This parameter specifies the specific channels that are being used.
- Time steps of transmission: As all data is generated at once, the user must specify the specific times of transmission in the form of discretized units of time.

An example of how this is implemented in the current MATLAB version of this tool can be seen in Fig. 2 and 3. First, in Fig. 2, we show the global parameter definitions. Since, at this point, all of the data is generated at one time, the user specifies the total number of channels and time steps considered for the whole network. The user also specifies the overall frequency range over which this network will operate. Next, in Fig. 3, we see an example of instantiating an emitter. This involves defining its location, transmission power, and channels and the time steps over which it will transmit, and finally the method for generating the data is called to generate its spectrum occupancy. The results from the use of these parameters, along with another emitter and three sensors, is shown in a later section.

The output data will appear qualitatively as follows:

Output:

- Matrix of binary occupancy data of size $time\ steps \times channel\ number$. Instead of a binary 1, an occupied value will correspond to the transmission power.

With the structure of the data in place, the method for generating this data in a given channel is needed. There are a number of methods discussed in literature that describe various statistical approaches to this problem [7]. Currently, this process is simplified by generating channel occupancy according to some

```
%% Define Global Network Properties

channelNum = 10;
timeStepNum = 100;
occupancy = zeros(timeStepNum,channelNum);

fHi = 2e9;     % GHz
fLo = 1e9;     % GHz

scale = 1000;
```

Fig. 2— Simulation example global parameters.

```
%% Define Emitters

e1 = Emitter;
e1.location = scale*[-4,4];
e1.label = 1;
e1.transmitPower = 20;
e1.channels = [2:5];
e1.timeStepNum = timeStepNum;
e1.timeSteps = [5:55];
e1.data = e1.occupancyGeneration(length(e1.channels),length(e1.timeSteps));
```

Fig. 3— Emitter Object

random variable and distribution, such as the Poisson or exponential. First, to generate variability across channels, the parameter of one of these distributions is generated randomly. For example, in the Poisson distribution shown in (1), the parameter λ would be generated randomly for each channel. This is similar for the exponential distribution.

$$f(k;\lambda) = Pr(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{1}$$

Next, using this parameter, a value can be generated from the desired distribution. This value gets compared to a threshold to determine whether it corresponds to being occupied or not. If an iteration does come out occupied, then data is generated according to the rounded value of the random variable. An example of occupancy generated according to this procedure for 20 channels and 100 time steps is shown in Fig. 4. The red values correspond to the channels being occupied in that bin, where again, the value is the transmitted power. Blue is unoccupied.

The procedure for generating data in a channel can, and will, be modified as the simulation framework develops. Currently, planned modes for generating data in a desired channel includes

1. continuous wave

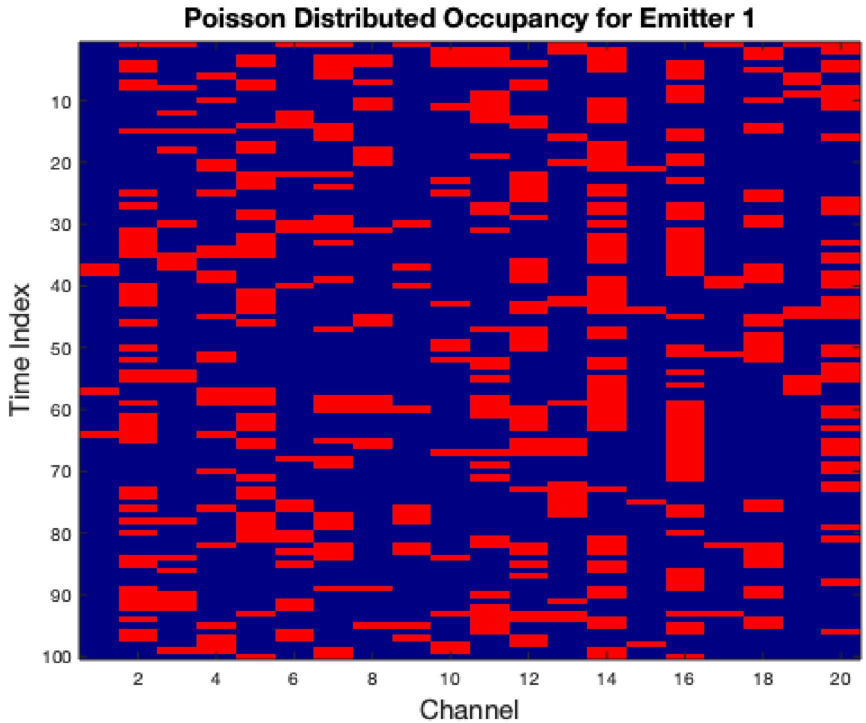


Fig. 4— Example of occupancy generation using the Poisson distribution across 20 channels and 100 time steps. Red is occupied and blue is unoccupied.

2. desired duty cycle
3. distribution type

Additional improvements to emitter generation include designating antenna characteristics, such as gain and directionality, and adding mobility to a node.

4. SENSOR MEASUREMENTS

Once the transmitted data is generated, it is sent to the sensors to make their local measurements. Recall that the main two components of the simulation framework involved the construction of the emitter and sensor classes. An instantiation of a sensor will be similar to an emitter in that the user specifies its location, channels being observed, optional antenna characteristics, etc. A sensor will differ in having additional properties associated with a radio, such as receiver sensitivity and decision threshold. The methods in the sensor class will provide the ability to add noise to the measurements, to calculate all of the losses, do a link budget calculation to each emitter to determine the received power in each channel, and to make the final occupancy decision.

As mentioned in the previous section, a property of the emitter class is the transmitted power. A sensor object will use this, and the distance to each emitter to calculate the received power. To compute this received power, a sensor can do a link budget calculation according to the general formula

$$P_{RX} = P_{TX} + G_{TX} - L_{TX} - L_{FS} - L_M + G_{RX} - L_{RX}, \tag{2}$$

where,

$$\begin{aligned}
P_{RX} &= \text{ received power (dBm)} \\
P_{TX} &= \text{ transmitter output power (dBm)} \\
G_{TX} &= \text{ transmitter antenna gain (dBi)} \\
L_{TX} &= \text{ transmitter losses (dB)} \\
L_{FS} &= \text{ free space path loss (dB)} \\
L_M &= \text{ miscellaneous losses (dB)} \\
G_{RX} &= \text{ receiver antenna gain (dBi)} \\
L_{RX} &= \text{ receiver losses (dB)}
\end{aligned}$$

As of now, the received-power calculation is simplified by using only the free space path losses, L_{FS}, where antenna gains and hardware losses can be incorporated later. The link budget calculation then simplifies to

$$P_{RX} = P_{TX} - L_{FS} \tag{3}$$

The received power then can be approximated using a known equation based on the Friis transmission formula below. A sensor will perform this calculation for each emitter and channel. Since a channel corresponds to some bandwidth, it will take the appropriate center frequency and will use that with the average loss for the whole channel.

$$L_{FS}(dB) = 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) = 20\log_{10}\left(\frac{4\pi d f}{c}\right) \tag{4}$$

4.1 Additive Noise

In addition to the received power after performing the link budget calculation, additive noise also is added to the measured data. Recall this tool was created to test and evaluate sensor fusion algorithms. Also, the algorithm only deals with power, or energy, values in generation and processing. To this end, adding the common white noise would correspond to adding a signal with an energy spectral density that is constant over all frequencies. Instead, non-white noise is added to the measurements. Because the model skips the front-end signal processing, non-stationary energy will need to be added in each bin. This will be modeled with a noise energy added as a random variable according some mean and variance. This gives a noise energy spectrum that will vary across frequency at each instant of time. To justify this, consider a received signal of an occupied bin in (5) consisting of the transmitted signal $s(t)$ and additive noise $n(t)$.

$$x(t) = s(t) + n(t) \tag{5}$$

The energy of a signal, denoted by ξ, corresponds to the area under the squared magnitude of that signal, defined as

$$\xi = \int_{\infty}^{\infty} |x(t)|^2 dt. \tag{6}$$

Since we are only interested in the energy in a given bin, we consider a bin window length T. Using this, (6) simplifies to

$$\xi = \int_{t}^{t+T} |x(t)|^2 dt. \tag{7}$$

Assuming the signal and the noise are uncorrelated, the expected energy in this bin then will be

$$\begin{aligned} E\{\xi\} &= E\left\{\int_{t}^{t+T} x(t)x^*(t)dt\right\} \\ &= E\left\{\int_{t}^{t+T} (s(t)+n(t))(s(t)+n(t))^* dt\right\} \\ &= \int_{t}^{t+T} |s(t)|^2 + E\{|n(t)|^2\} dt. \end{aligned}$$

If the additive noise is white, then the expected power in a bin would be a constant value since the expected value in the integral would be constant. To make the noise non-white, we then give it a time dependent value by generating this energy according to a random variable with a user defined noise and variance. Future additions could include additional factors that would affect the reception of the true signal, such as adding the statistical behavior of multipath propagation and, in the case of true signal generation and reception, Doppler processing.

4.2 Threshold

After the received power is calculated, the measurements correspond to the red, dotted line in Fig. 1. To make the final occupancy decisions, the received power in each bin needs to be evaluated against some threshold. Currently, a universal threshold is set across all sensors, though the class provides the ability to define a sensor-specific threshold. In practice, due to the possibility that any given sensor could see a different noise floor than the others in the network, this threshold value likely would be unique to each sensor. A more rigorous procedure to determine a node's threshold is future work.

4.3 Sensor Definition Example

In Fig. 5, we see a MATLAB example of instantiating a sensor. Similar to an emitter, this involves defining its location, the channels being observed, the receiver bandwidth, the local threshold value, and the receiver sensitivity. This is the minimum number of parameters that must be defined for the program to run. Additional properties could include the gain related to the antenna and hardware related losses. Many of the methods associated with a sensor run behind the scenes, calculating losses, center frequencies, etc. The main method that would need to be called takes as its argument the emitter objects, and uses that to calculate bin power. An example of what this output looks like is shown in the next section.

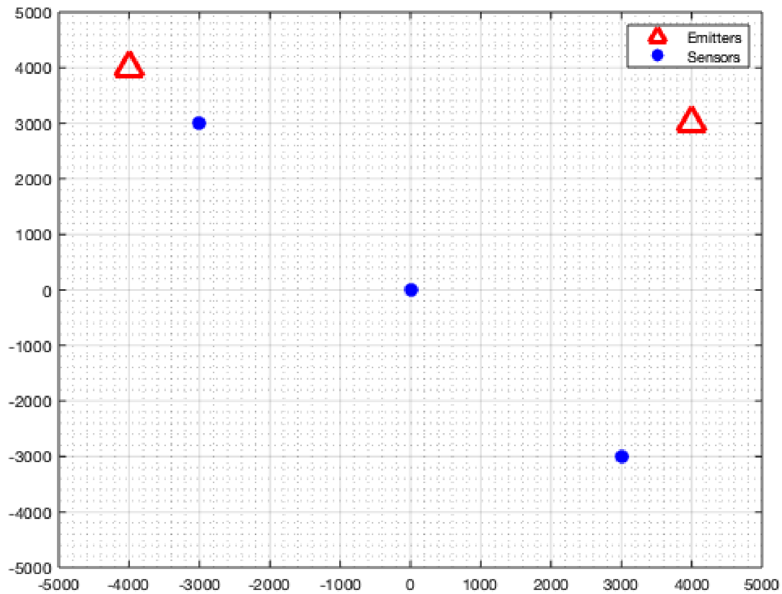
```
%% Define Sensors

s1 = Sensor;
s1.label = 1;
s1.location = scale*[-3,3];
s1.channelNum = channelNum;
s1.channels = [1:channelNum];
s1.fHi = fHi;
s1.fLo = fLo;
s1.threshold = threshold;
s1.receiverSensitivity = -90;
```

Fig. 5— Sensor Object

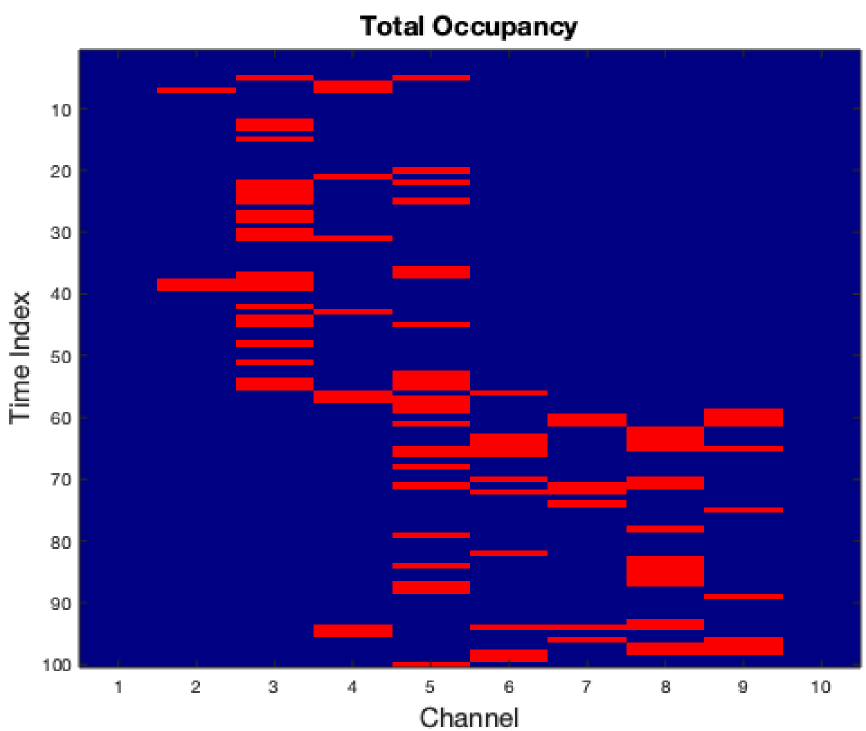
5. MATLAB SIMULATION

Using the properties and methods associated with the emitter and sensor classes, the current flow of generating a network begins by defining the global parameters. Currently, this involves defining the high and low frequencies of interest, and the total number of channels and time steps for the whole network. The next step is to define an emitter object and associated properties. This includes the field mentioned previously. Current work on the simulation framework allows for sensor and emitter that may be using a fraction of the aforementioned global channels and time steps. An example of this is shown in Fig. 6, where the network consists of two transmitters and three sensors. The emitters were defined such that they generate data over different channels and time steps.

The user first defines the emitters and their necessary properties as shown in a previous section. Then the three sensors are defined as previously shown, with each having the ability to have different operating parameters. The user then calls the sensor class method to calculate the received power in each bin. Many of the necessary methods for this run behind the scenes, such as calculating distances, center frequencies, and losses. An example of the last four lines of output from the network defined in Fig. 6 is shown in Fig. 7, with the associated losses to each node shown in Fig. 8. What this data shows is the output data from the total occupancy matrix, in which an emitter is transmitting at 20 dB in the occupied bins at the time steps shown. That data is taken by the sensors, the losses calculated and used in a link budget calculation, and random noise energy is added according to a user-defined mean and variance. The resulting received power values in each bin are shown.



(a)



(b)

Fig. 6— (a) Two-dimensional network example consisting of two emitters and three sensors. (b) The occupancy data generated by the two emitters is defined for different channels and time steps. Red bins denote occupied, and blue unoccupied.

```
Ground truth occupancy data w/ transmit power:
     0     0     0     0     0     0     0    20    20     0
     0     0     0     0     0    20     0    20    20     0
     0     0     0     0     0    20     0     0     0     0
     0     0     0     0    20     0     0     0     0     0

Sensor 1 located at (-3000,3000)
  -73.8167  -70.3183  -82.4281  -90.0000  -90.0000  -83.3316  -82.9572  -55.8988  -56.6270  -75.1470
  -90.0000  -86.1442  -83.7895  -83.9501  -77.5820  -56.9434  -75.9516  -50.3767  -72.0495  -90.0000
  -82.9188  -89.6246  -85.3266  -81.8382  -86.3507  -75.3048  -90.0000  -90.0000  -82.5899  -90.0000
  -76.2547  -87.8308  -87.6716  -90.0000  -77.0100  -75.0842  -88.1889  -90.0000  -78.3382  -79.9206

Sensor 2 located at (0,0)
  -60.2823  -65.6931  -79.8460  -71.5800  -78.2584  -59.2963  -88.6801  -69.8180  -61.6813  -79.3148
  -68.9679  -80.9920  -90.0000  -81.6893  -67.6493  -51.1906  -68.5389  -67.2944  -55.4874  -82.4177
  -70.7714  -83.3075  -87.2088  -90.0000  -79.3189  -70.8309  -80.1683  -79.8369  -90.0000  -73.9450
  -80.7756  -78.0395  -76.3865  -82.9817  -69.1002  -90.0000  -83.5423  -90.0000  -86.5881  -87.7622

Sensor 3 located at (3000,-3000)
  -84.7033  -88.4955  -79.4010  -73.8582  -84.0423  -68.2690  -73.7489  -71.7029  -68.1961  -90.0000
  -81.6793  -90.0000  -90.0000  -90.0000  -69.1203  -65.4439  -80.3401  -60.5239  -58.7338  -90.0000
  -73.3100  -90.0000  -66.6382  -71.2868  -89.5508  -66.1061  -90.0000  -67.5412  -90.0000  -90.0000
  -82.5469  -83.4348  -72.3463  -64.9738  -66.5769  -88.6275  -84.5533  -85.6536  -90.0000  -62.7668
```

Fig. 7— Example data for last four time bins. Received data includes free space losses and added noise.

```
Channel loss for sensor 1:
Emitter 1:   95.8759   96.6660   97.3903   98.0587   98.6794   99.2587   99.8018  100.3128  100.7955  101.2528

Emitter 2:  109.7675  110.5577  111.2819  111.9504  112.5711  113.1504  113.6934  114.2045  114.6872  115.1444

Channel loss for sensor 2:
Emitter 1:  107.9171  108.7072  109.4315  110.0999  110.7206  111.2999  111.8430  112.3540  112.8367  113.2940

Emitter 2:  106.8450  107.6351  108.3594  109.0278  109.6485  110.2278  110.7709  111.2819  111.7646  112.2219

Channel loss for sensor 3:
Emitter 1:  112.7778  113.5680  114.2922  114.9607  115.5814  116.1607  116.7037  117.2148  117.6975  118.1547

Emitter 2:  108.5476  109.3377  110.0620  110.7305  111.3511  111.9304  112.4735  112.9846  113.4672  113.9245
```

Fig. 8— Channel loss in dB for each sensor to emitter. Emitter 1 located at (-4000, 4000) and Emitter 2 located at (4000, 3000).

6. NEXT PHASE

The immediate upcoming improvements to the model involve improvements to how the simulation is run, and modifying the physical layer losses and environmental modeling. First, the model should generate and process data in an on-line manner. This is in contrast to the current iteration, in which all of the data is generated at the same time. Handling data in this way, allows the model to introduce additional aspects and errors that would be in a real wireless network. An example of this is to introduce delays in data generation and processing, to see how that affects the algorithm. Another addition to the model includes adding node mobility. This includes both moving nodes and transmitters. Next, we would like to add more realistic propagation models and to allow for the handling of inter-channel interference. An improved propagation model could allow for the statistical behavior of multi-path, and a more rigorous model for generating noise. Along with these improvements, work will begin towards moving the simulation framework over to Python.

After adding improvements to the physical layer modeling, the next goal is to add the functionality of other layers in the network. This would be beneficial towards the algorithm design because it would allow for adding other network errors in the design. Such errors could include packet losses, network congestion, Age of Information (AoI), etc. Modifications could include generating packets to transmit the data, and adding protocols to the network operation. There is also potential for this to be tied in with other networking tools as an extension to modeling the physical layer. This is subject to further study and consideration.

7. CONCLUSION

This report outlines a new, distributed-spectrum-generation tool that is under development for the purpose of testing and evaluating spectrum-fusion algorithms. The model is currently being developed in MATLAB, utilizing an object-oriented approach to generating emitters and sensors. This is an important aspect of the tool, as it allows for generating spectrum occupancy data with heterogeneous set of transmitters and sensors that can transmit and receive over different frequency bands. As of now, data is generated completely at an emitter, and then is processed at a sensor to account for any of the losses. The framework can handle an arbitrary number of sensors and emitters, and can generate spectrum occupancy data accordingly. In the next iterations of the tool, there are a number of changes and improvements we want to make, as outlined in the previous section, that add and improve current functionality to account for more realistic issues pertaining to data transmission (delivery of spectrum data from the sensors to the fusion center) and performance.

REFERENCES

1. Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors* **19**(1), 126 (2019).

2. D. M. Alias and Ragesh G. K, "Cognitive Radio networks: A survey," Proceedings of the 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 1981–1986.

3. F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal* **2013** (2013).

4. W. Li, Z. Wang, G. Wei, L. Ma, J. Hu, and D. Ding, "A Survey on Multisensor Fusion and Consensus Filtering for Sensor Networks," *Discrete Dynamics in Nature and Society* **2015**, 12 (10 2015), doi:10.1155/2015/683701.

5. M. L. Fung, M. Z. Q. Chen, and Y. H. Chen, "Sensor fusion: A review of methods and applications," Proceedings of the 2017 29th Chinese Control And Decision Conference (CCDC), 2017, pp. 3853–3860.

6. M. Dillon, C. Pici, S. Kompella, J. Molnar, and L. Tran, "Multi-Sensor Correlation and Fusion for Spectrum Occupancy," NRL Formal Report NRL/FR/5524–20-10,406, Naval Research Laboratory, Washington, DC, 2020.

7. M. López-Benítez and F. Casadevall, "An Overview of Spectrum Occupancy Models for Cognitive Radio Networks," in V. Casares-Giner, P. Manzoni, and A. Pont, eds., *NETWORKING 2011 Workshops*, Berlin, Heidelberg (Springer Berlin Heidelberg), 2011, pp. 32–41.